Personalised Modelling Framework and Systems for Gene Data Analysis and Biomedical Applications

Yingjie Hu

A thesis submitted to Auckland University of Technology
in fulfillment of the requirements
for the degree of Doctor of Philosophy - PhD

March, 2010

School of Computing and Mathematical Sciences

Primary Supervisor: Prof. Nikola Kasabov
Secondary Supervisor: Dr. Qun Song

Contents

List of Figures

List of Tables

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the qualification of any other degree or diploma of a university or other institution of higher learning, except where due acknowledgment is made in the acknowledgments.

Yingjie Hu

List of Abbreviations

CD	-	Crohn's disease	9
cDNA	-	Complementary DNA	13
cEAP	-	Co-evolutionary algorithm based method for gene selection and parameter optimisation in personalised modelling	xxv
cGA	-	Compact genetic algorithm	33
cGAPM	-	Compact GA search based personalised modelling system	120
CNS	-	Central nervous system embryonal tumours	xxx
DLBCL	-	Diffuse large B-cell lymphoma	xxix
DNA	-	deoxyribonucleic acid	11
ECF	-	Evolving classification function	39
ECOS	-	Evolving Connectionist System	6
EFuNN	-	Evolving Fuzzy Neural Networks	37
EHE	-	epithelioid hemangioendothelioma	19
FN	-	False negative	105
FP	-	False positive	105
FL	-	Follicular lymphoma	xxix
FPR	-	False positive rate	91
GA	-	Genetic algorithm	26
GWA	-	genome-wide association	167
iPM	-	Increment search based approach for personalised modelling	xxv

KEDRI	-	the Knowledge Engineering and Discovery Research Institute	xxii
KNN	-	K nearest neighbour	3
LOOCV	-	Leave-one-out cross validation	58
MF	-	Membership functions	37
MLP	-	Multi layer perceptron	44
MLR	-	Multiple linear regression	48
mRNA	-	Messenger ribonucleic acid	4
PCA	-	Principal component analysis	16
PMF	-	Personalised modeling framework	xxiv
PMS	-	Personalised modelling system	xxiv
RBF	-	Radial basis function	46
RMSE	-	Root mean square error	89
RNA	-	Ribonucleic acid	12
ROC	-	Receiver operating characteristic	90
rRNA	-	ribosomal RNA	13
sGA	-	Simple genetic algorithm (conventional genetic algorithm)	34
SNPs	-	Single nucleotide polymorphisms	xxx
SNR	-	Signal-noise-to-ratio	47
SOM	-	Self-organizing maps	51
SVM	-	Support Vector Machine	16
TN	-	True negative	105
TP	-	true positive	105
TPR	-	True positive rate	91
tRNA	-	Transfer ribonucleic acid	13
TWRBF	-	A transductive inference based radial basis function	45
TWNFI	-	Neural Fuzzy Inference System with Weighted Data Normalization	45
WKNN	-	Weighted distance KNN method	55
WTCCC	-	Wellcome Trust Case Control Consortium	167
WWKNN	-	Weighted distance and weighted variables K-nearest neighbours	56

Acknowledgment

First and foremost, I am heartily thankful to my supervisor, Professor Nikola Kasabov, whose encouragement, guidance and support from the initial to the final stage enabled me to develop a deep understanding of the subject. Nikola has a remarkably good taste in research and an excellent sense of strategy. He showed me how to approach a research problem in different ways and find the best solution. I greatly appreciate for his open personality, patience, enthusiasm, and immense knowledge that, taken together, make him a great supervisor.

I would like to thank Dr. Qun Song, for his advices in the experimental design and data analysis, which forms important parts of this research. He has been actively interested in my work and has always been available to advise me during his stay in our institute.

While conducting this research, I have had a great time to work and learn from many people, thanks in large part to the stimulating environment of the Knowledge Engineering and Discovery Research Institute (KEDRI) . I would like to thank the past and present members of KEDRI for their support, straight-talking honesty, which ranks among one of the most profound aspects of my study. Stefan Schliebs deserves special acknowledgment for his thoughtful advice, friendship and a lot of insightful discussions. I thank Dr. Peter (Yuan-Chun) Hwang, for always being around for helpful discussions and for providing technical supports. Dr. Michael Defoin-Platel, he helped me a lot with his own background in bioinformatics and evolutionary computation. Vishal Jain whom I enjoyed collaborating with for a GRN project. Because they deserve it and are not thanked nearly enough, I would also like to thank the staff of the KEDRI, Dr. Shaoning (Paul) Pang, Dr. Simei Gomes Wysoski, Harya Widiputra, Marin Karaivanov, Haza Nuzly, Gary Chen, Lei

Song, and Kshitij Dhoble. Their helpful influence is clear, and have enriched my educational experience immeasurably.

Particularly, I am indebted to Joyce D'Mello who always was ready to help me in whatever situation I confronted and encouraged me at the times I was about to give up. No doubt my study would have looked very different if there was no support from Joyce.

I would like to thank Diana Kassabova who kindly proofread my thesis and offered grammatical assistance.

Thanks also to the Tertiary Eduction Commission of New Zealand for the financial support through the Top Achiever Doctoral Scholarship.

On a personal level, I thank my wife Li Shen for her support. She gives me strength and confidence. Without her love, patience and encouragement, this work would never be completed. I owe to my grandmother and my parents who definitely cannot be thanked enough.

Lastly, I offer my regards and blessings to all of those who have supported me in any respect during the completion of the study.

Abstract

The core focus of this research is at the development of novel information methods and systems based on personalised modelling for genomic data analysis and biomedical applications. It has presented a novel personalised modelling framework and system for analysing the data from different sources and discovering the knowledge through an evolving and adaptive way. The main idea of personalised modelling is based on the assumption that every data sample has its unique pattern only being represented by a certain number of similar samples with a small set of important features. The proposed personalised modelling system (PMS) is an integrated computational system that combines different information processing techniques, applied at different stages of the data analysis, e.g. feature selection, classification, discovering the interaction of genes, outcome prediction, personalised profiling and visualisation, etc.

In summary, this research has presented the main following contributions:

(1) This study has implemented the idea of personalised modelling framework (PMF) introduced by Kasabov (2007b);

(2) This study has developed novel algorithms and methods for PMS, which are described in Chapter 6;

(3) I have addressed the issuess in personalised modeling for data analysis and proposed solutions in Chapter 5;

(4) I have analysed the proposed PMS on 6 types of cancer gene expression data in Chapters 6, 7 and 8;

(5) This thesis has presented the case studies of 4 types of cancer gene expression data analysis in Chapter 7 ;

(6) This study proposed a method using a coevolutionary algorithm for personalised modeling to select features and optimise relevant parameters for data analysis in Chapter 8.

(7) I have applied the proposed PMS on a SNPs dataset for Crohn's disease risk evaluation in a real world case study in Chapter 9;

(8) The thesis gives the future research directions for personalised modelling study.

To construct a PMS for knowledge discovery, new algorithms and methods have been developed in the course of this study: (1) personalised modelling based gene selection, (2) increment search based approach for personalised modelling (iPM) , (3) genetic algorithm search based approach for personalised modelling, (4) compact GA search based personalised modelling, and (5) co-evolutionary algorithm based method for gene selection and parameter optimisation in personalised modelling (cEAP) .

Using these developed algorithms and methods, I have implemented a personalised modelling system for data analysis and knowledge discovery from a simple approach to the more sophisticated approaches. The implemented PMS is illustrated on bench mark data sets and applied on real data: gene expression data of 6 types of cancer; SNPs data for Crohn's disease risk analysis (from the UK Welcome Trust Repository).

The experimental results from the proposed PMS have shown the improved performance in terms of classification accuracy. More importantly, such a framework and system create an optimal personalised model combining informative features (e.g. genes) and optimised relevant parameters. The knowledge elicited from the created personalised model allows us to profile every new input data sample, which is very useful for the problems that need precise information for each individuals, e.g. the design of tailored treatment for a cancer patient.

This study is a feasibility analysis for personalised modelling on different sources of data, such as gene expression data, proteomic data and SNPs data. To the best of my knowledge, it is the first comprehensive study of personalised modelling from the

point of view of computational intelligence. The findings from this study also encourage us to carry out in-depth study for solving open questions in future research. The developed algorithms and models are generic which can be potentially incorporated into a variety of applications for data analysis and knowledge discovery with certain constraints, such as financial risk analysis, time series data prediction, to name only a few.

Publication Lists

The following is a list of my published papers based on the algorithms and techniques presented in this thesis during my PhD study:

- **Book Chapters**

  1. **Hu, Yingjie**, Kasabov, N. (2009). Coevolutionary Method for Gene Selection and Parameter Optimization in Microarray Data Analysis. In C. S. Leung, M. Lee & J. H. Chan (Eds.), Neural Information Processing (Vol. 5864, pp. 483-492). Berlin/Heidelberg: Springer.
  2. **Hu, Yingjie** , Nikola Kasabov, 2008). Ontology-based framework for personalized diagnosis and prognosis of cancer based on gene expression data. In M. Ishikawa, K. Doya, H. Miyamoto & T. Yamakawa (Eds.), Neural Information Processing (pp. 846 - 855). Berlin, Heidelberg: Springer-Verlage.
  3. Nikola Kasabov, Qun Song, Lubica Benuskoval, Paulo Gottgtroy, Vishal Jain, Anju Verma, Ilkka Havukkala, Elaine Rush, Russel Pears, Alex Tjahjana, **Yingjie Hu**, Stephen MacDonel, Integrating Local and Personalised Modelling with Global Ontology Knowledge Bases for Biomedical and Bioinformatics Decision Support, Chapter 4, In: Smolin et al (eds) Computational Intelligence in Bioinformatics, Springer, 2008
  4. Pang, S., Havukkala, I., **Hu, Yingjie**., Kasabov, N.: Bootstrapping Consistency Method for Optimal Gene Selection from Microarray Gene Expression Data for Classification Problems. Chapter 4, In: Zhang, Y.-Q., Rajapakse, J.C. (eds.): Machine Learning for Bioinformatics. John Wiley & Sons, Inc., New Jersey (2008)

- **Journal Papers**

  1. Pang, S., Havukala, I., **Hu, Yingjie**, Kasabov, N.: Classification Consistency Analysis for Bootstrapping Gene Selection. Neural Computing and Applications 16 (2007) 527-539
  2. Chan, Z.S.H., Havukkala, I., Jain, V., **Hu, Yingjie**, Kasabov, N.: Soft Computing Methods to predict Gene Regulatory Networks: An Integrative approach on Time-Series Gene Expression Data. Applied Soft Computing 8 (2007) 1189-1199

- **Conference Papers**

  1. Kasabov, Nikola, **Hu, Yingjie**, Liang, L.: Personalised Modelling for Risk and Outcome Prognosis on a Case Study of Brain Disease. 1st International Congress on Clinical Neurology & Epidemiology, Munich, Germany (2009)
  2. **Hu, Yingjie**, Song, Q., Nikola Kasabov: Personalized Modeling based Gene Selection for Microarray Data Analysis. In: M.Koeppen, N.Kasabov, G.Coghill, M.Ishikawa (eds.): ICONIP 2008. Springer LNCS, Auckland (2009)

- **Submitted Journal Papers**

  1. Kasabov, Nikola, **Hu, Yingjie** : Globally Optimised Personalised Models for Medical Decision Support. International Journal of Functional Informatics and Personalised Medicine. Submitted.

Datasets Used in the Thesis

Four benchmark cancer gene(protein) expression datasets ares used in this study:

1. Colon cancer data (Alon et al., 1999)
   (available at `http:/microarray.princeton.edu/oncology/`)
   The data consist of 62 samples collected from colon cancer patients, in which 40 samples are labeled as diseased and 22 are labeled as normal. Each sample consists of 2,000 genes.

2. Leukaemia data (Golub et al., 1999)
   (available at `http://www-genome.wi.mit.eud/MPR/`)
   The biology task on this data is to distinguish two types of leukaemia - Acute Lymphoblastic Leukaemia(ALL) and Acute Myeloid Leukaemia(AML). Leukaemia data contains 72 samples(47 ALL vs. 25 AML), each sample represented by the expression levels of 7,129 probes from 6,817 genes.

3. Lymphoma data (Alizadeh et al., 2000)
   (available at `http:/llmpp.nih.gov/lymphoma/`)
   The data contain the expression levels of 4026 genes across 96 samples in lymphoma patients. Among them, 42 samples belong to Diffused large B cell lymphoma (DLBCL) group while 54 are from other types. The objective of the study is to discriminate DLBCL and other types of lymphoma.

4. DLBCL data (Shipp et al., 2002)
   (available at `http://www.ailab.si/orange/datasets/DLBCL.htm`)
   The dataset contains 58 DLBCL samples and 19 Follicular lymphoma (FL) samples. Each sample is represented by 6817 genes. DLBCL and FL are

two B-cell lineage malignancies that have very different clinical presentations, natural histories and response to therapy. However, FLs frequently evolve over time and acquire the morphologic and clinical features of DLBCLs and some subsets of DLBCLs have chromosomal translocations characteristic of FLs. The biological objective of the analysis on this data is to distinguish between these two type of lymphomas.

5. Lung cancer data (Gordon, Jensen, Hsiao, Hsiaox, & JE, 2002)
   (available at `http://www.chestsurg.org/microarray.htm`)
   This dataset is originally used for classification between malignant pleural mesothelioma (MPM) and adenocarcinoma (ADCA) of the lung cancer diagnosis. The complete dataset has 181 tissue samples (31 MPM vs. 150 ADCA) and each sample is described by 12533 genes.

6. Ovarian cancer data (Petricoin et al., 2002)
   (available at `http://clinicalproteomics.steem.com/`)
   This dataset contains 253 samples in which 91 samples are labeled as healthy and 162 are ovarian cancer. There are total 15154 proteins for identifying tumor patterns.

7. Central Nervous System (CNS) cancer data (Pomeroy et al., 2002)
   (available at `http://www-genome.wi.mit.eud/mpr/CNS/`)
   The CNS cancer data used in this thesis is the dataset C in Pomeroy's work (Pomeroy et al., 2002). It consists of 60 patient samples, in which 39 are medulloblastoma survivors (class 2) and 21 are treatment failures (class 1). The learning objective of this gene expression data is to classify the patients who survive after the treatment and those who are succumbed to central nervous system cancer. Each sample is represented by 7,129 probes from 6,817 human genes.

8. Single nucleotide polymorphisms (SNPs) data for Crohn's disease risk prediction
   (available at `http://www.wtccc.org.uk`)
   The data consists of three subsets:
   Dataset A and B are the datasets for training. Dataset A contains 1049 samples in which 561 samples are diseased and 488 are controls. Dataset B contains 1045 samples in which 560 samples are crohn's disease cases, while 485 are

controls. Dataset C is the testing set that includes 1062 samples (577 diseased cases vs. 485 controls).

CHAPTER 1

Introduction

"The beginning of knowledge is the discovery of something we do not understand."

- Frank Herbert

1.1 Background: Contemporary Research in Life Sciences

The scale and the techniques of life science research have been changed significantly since human society entered genomics era in the mid 1980s. Microarrays have become one of the most important technological breakthroughs in biological science that enable scientists to understand difficult problems at a genomic level. For example, microarrays offer a new approach to discover the biological mechanisms that trigger normal genes to become cancerous. With the advancement of genomic technology and the support from computer and information science, system biology has progressed into a new paradigm where the research is shifting from studying single-variable (single-gene) to studying complex gene interactions.

Health informatics, clinical research and the widely-spread use of microarray technology have all contributed to the generation and accumulations of vast amount of data. This data comes from areas, such as functional genomics, proteomics, metabolomics, patients' clinical information, etc. The discovery of the hidden relationships and patterns in the available data could provide researchers with new knowledge in a variety of areas, e.g. new oncogenes discovery, disease diagnosis, therapeutic treatment design, drug response prediction, to name but a few. There has been an ever-increasing need for biological, medical and computer scientists to work together for data retrieval, analysis, visualisation and knowledge discovery.

Computational intelligent techniques have been therefore put forward to bioinformatics related tasks, such as modelling, diagnostic, learning and optimisation, with applications in several areas. The application of computational intelligent techniques in biomedical science is not as recent as we might think. In fact, the utilisation of computational intelligent techniques in medical research can be tracked back to the late 1970s. Many research projects attempted to use statistics and other simple techniques to investigate the feasibility for analysing large clinical databases during the 1970s and 1980s (Breiman, Stone, Friedman, & Olshen, 1984). Some of these works are: the project carried out at the Brighman and Women' hospital, in Boston, USA, to create decision trees using recursive partitioning methods in myocardial infarction for making clinical dicision (Goldman et al., 1988), and the study that created methodology for developing clinical prediction rules (Wasson, Sox, Neff, & Goldman, 1985). However, researchers found that it was difficult to acquire knowledge from medical expert systems in a specific domain using traditional statistical techniques (Anderson, 2000). Researchers moved on to the utilisation of computational intelligence methods such as machine learning techniques could be a new and effective approach to discover knowledge from medical datasets (Maojo, 2004).

KARDIO system (Bratko, Mozetic, & Lavac, 1989) is a pioneering study in terms of using computational intelligence for knowledge discovery in medical expert systems. The system is designed for cardiological diagnosis and treatment, where an inductive algorithm is used to extract rules from large clinical databases. Since that time, computational intelligent techniques have been extensively used for medical data analysis (Lavrac, Keravnou, & Zupan, 1997). The discovered knowledge can be used for various purposes, such as diagnosis, prognosis, visualising, monitoring, treatment decision supporting. Another study (Cooper et al., 1997) used several methods,

namely logistic regression, decision trees, Bayesian networks, neural networks and K-nearest-neighbour (KNN) to discover clinical predictors in pneumonia mortality.

The emergence of microarray technology provides a new platform to study complex diseases, such as cancer. The technology assists researchers to untangle the vast complexity of the relationships among geneotypes, phenotypes development, environment and evolution (Baldi & Hatfield, 2002). For clinical purposes, microarray technology plays an important role in understanding the pathway of disease (especially for cancer), for designing tailored diagnostic strategies and for creating personalised molecular medicine.

The contemporary life sciences research requires integrated computational intelligent models and systems for the study of medical problems related to diseases that kill hundreds of thousands of people every year, such as cancer. Ideally, the models should combine:

1. Different sources of information, such as gene expression microarray data, proteomics data, human expertise knowledge, clinical data, etc.
2. Different information processing techniques, applied at different stages of the data analysis, e.g., data pre-processing, feature selection, clustering, classification, discovering the interaction of genes, outcome prediction, risk evaluation, etc.

Despite the availability of large genetic and clinical databases and the enormous human expertise related to diseases, there are very few specific information procession methods and systems that have been successfully used for gene expression data modelling, for disease prognosis and for drug target discovery, specifically for new individual patients who have complex disease, such as cancer.

1.2 Why Personalised Modelling?

In order to develop an understanding of personalised modelling for gene data analysis and biomedical applications, we must answer the question: *why do we need*

personalised modelling for gene data analysis and for biomedical applications? Contemporary medical and other data analysis and decision support systems use predominantly inductive global models for the prediction of a person's risk, or of the likely outcome of a disease for an individual (Anderson et al., 2006; Levey et al., 1999). In such models, features are pre-processed to minimise learning function's error (usually a classification error) in a global way to identify the patterns in large databases. Pre-processing is performed to constrain the features used for training global learning models. In general, global modelling is concerned with deriving a global formula (e.g. a linear regression function, a "black box neural network", or a support vector machine) from a large group of data samples. Once an optimal global model is trained, a set of features (variables) are selected and then applied to the whole problem space (i.e. all samples in the given dataset). Thus, the assumption is made that the global model is able to work properly on any new data sample. In clinical research, therapeutic treatment designed to target a disease is assumed to be useful for everybody who suffers from this disease. The drugs developed as a result of this global approach have been successful in revolutionising medicine over the past decades.

Statistic reports from the medical research community have shown that drugs developed by such global modelling methods are only effective for approximately 70% of people who need treatment, leaving a relatively large number of patients who will not benefit from the treatment at all (Shabo, 2007). Regarding aggressive diseases, such as cancer, any ineffective treatment of a patient (e.g. either a patient not being treated, or being incorrectly treated), can be the difference between life and death. Such global modelling based medical treatment systems are not always applicable to the individual patients, as the molecular profiling information is not taken into account. The heterogeneity of diseases (e.g. cancer), means that there are different disease progresses and different responses to the treatment, even when the patients have similar remarkably morphologically tumours in the same organ. Thus, a more effective approach is required that would use a patient's unique information, such as protein, gene or metabolite profile to design clinical treatment specific to the individual patient.

The advance of molecular profiling technologies, including microarray messenger ribonucleic acid (mRNA) gene expression data, proteomic profiling, and metabolomic information make it possible to develop "personalised medicine" based on new molec-

ular testing and traditional clinical information for treating individual patient. According to the United States Congress, the definition of *personalised medicine* is given as "the application of genomic and molecular data to better target the delivery of health care, facilitate the discovery and clinical testing of new products, and help determine a person's predisposition to a particular disease or condition" (Senate Health, Education, Labor, and Pensions, 2007). The personalised medicine is expected to focus on the factors affecting each individual patient and for help fight chronic diseases. More importantly, it could allow the development of medical treatment tailored to an individual's needs.

Motivated by the concept of personalised medicine and utilising transductive reasoning (Vapnik, 1998), personalised modelling was recently proposed as a new method for knowledge discovery in biomedical applications. For the purpose of developing medical decision support systems, it would be particularly useful to use the information from a data sample related to a particular patient (e.g. blood sample, tissue, clinical data and/or DNA) and tailor a medical treatment specifically for her/him. This information can also be potentially useful for developing effective treatments for another part of the patient population.

In a broader sense, personalised modelling offers a new and effective approach for the study in pattern recognition and knowledge discovery. The created models are more useful and informative for analysing and evaluating an individual data object for a given problem. Such models are also expected to achieve a higher degree of accuracy of prediction of outcome or classification than conventional systems and methodologies (Kasabov, 2007b).

Personalised modelling has been reported as an efficient solution for clinical decision making systems (Song & Kasabov, 2006), because its focus is not simply on the global problem space, but on the individual sample. For a new data vector, the whole (global) space usually contains much noise information that presents the learning algorithm working properly on this new data, though the same information might be valuable for other data samples. With personalised modelling, the noise (or redundant) information can be excluded within the local problem space that is only created for the observed data sample. This characteristic of personalised modelling makes it a more appropriate method for discovering more precise information specifically for the individual data sample than conventional models and systems.

1.3 Research Goal and Objectives

Evolving intelligent methods have been adopted as one of the major computational tools for optimisation problems in bioinformatics research, e.g. for constructing medical prediction models. In this research, evolving intelligent methods and systems refer to the methods and systems that are able to evolve towards better solutions for optimising tasks. Such methods and systems may include a variety of algorithms and methods, such as evolutionary algorithms, swarm intelligence systems and evolving connectionist systems (ECOS) (Kasabov, 2003, 2007a).

1.3.1 Research Goal and Objectives

The goal of this research is to develop novel information methods and systems for personalised modelling (PM) and specifically for genomic data analysis and biomedical applications. The main objective of this research is to investigate this new and promising area, and build a generic modelling environment using Personalised Modelling based Framework (PMF) for biomedical data analysis. This research will approach the task in the following way: Creating a methodology for gene expression data and biomedical data modelling and knowledge discovery using evolving intelligent computational techniques. This would involve gene expression data preprocessing and feature selection, building a model based on the learning process (e.g., classifiers); model testing and validation; outcome visualisation and integration.

1.3.2 Specific Research objectives

More specifically, the research include the following objectives:

1. To critically analyse the problems related to PM.
   Although plenty of computational intelligent models have been so far developed for genomic data analysis, there are few integrated systems that can be successfully used for constructing medical decision support system. There are still a variety of issues that have not been resolved. For example, identifying which genes are informative in the microarray gene expression data.

2. To develop a generic modelling environment based on the personalised modelling framework and to analyse its performance under different scenarios.

3. To develop new methods for personalised feature selection and personalised profiling.
   Personalised modelling creates a unique model using a small number of informative features that highly represent an individual data vector's pattern. Thus, feature selection is a fundamental step to create a personalised modelling system (PMS) for analysing different data sources, such as microarray gene expression data, protein data, single nucleotide polymorphisms (SNPs) data, etc.

4. To develop a PMS for gene expression data modelling and classification.
   One major task for bioinformatics research is to utilise gene expression data for complex human disease study, such as cancer and diabetes. This study aims to develop a PMS for gene expression data analysis and investigate its performance over bench mark microarray gene expression datasets.

5. To develop a PMS for SNPs data modelling and classification.
   This study will present a PMS for SNPs data modelling and risk of disease evaluation. It is a feasibility analysis of personalised modelling on SNPs data for clinical application.

In summary, the ultimate objective of this research is to develop new methods and systems for personalised modelling that leads to improved classification performance and personalised profiling. Such methods and systems integrate novel machine learning and modelling techniques for:

◊ feature selection;

◊ classification;

◊ adaptation to new data;

◊ knowledge discovery and model validation;

◊ data sample profiling and results visualisation.

1.4 Organisation of the Thesis

The remainder of this thesis covers the development of a new proposed framework and system for personalised modelling.

- Chapter 2 gives an introduction to genomic data analysis including gene expression data and SNPs data analysis. It also provides a literature review covering the related biological background;
- Chapter 3 presents an overview of a range of computational intelligent techniques that are relevant to this research. I provide a brief description of the widely used techniques that have been used for genomic data analysis and biomedical applications;
- Chapter 4 briefly reviews modelling approaches and techniques for data analysis and knowledge discovery. It gives the description of three main modelling approaches, namely global, local and personalised modelling. It also presents a comparison study where the three modelling approaches are applied on a benchmark gene expression dataset for a classification task.
- Chapter 5 presents a critical analysis of the problems related to PM. It addresses the issues related to PM and gives potential solutions for the problems;
- Chapter 6 gives a conceptual framework of PM. This framework is used for the creation of five new algorithms to implement functional modules and for the implementation of three personalised modelling systems for modelling and knowledge discovery. Also, this chapter has presented a general strategy for evaluating proposed algorithms and PMSs;
- In Chapter 7, a PMS is developed that can be used on cancer gene expression data. A GA based PMS is applied on four benchmark genomic datasets for cancer classification;
- Chapter 8 proposes a novel method and system for feature selection, neighbourhood selection and model optimisation. The new method uses a coevolutionary algorithm for optimisation;

- Chapter 9 describes a case study that uses a SNPs dataset for Crohn's disease (CD) risk prediction. This task is a real-world biomedical analysis problem that presents challenges to personalised modelling. This case study has demonstrated the strength of personalised modelling over global modelling when applied on specific SNPs data;
- Chapter 10 summarises the thesis and gives the conclusion followed by future research directions.

CHAPTER 2

Genomic Data Analysis

"Having a sequence of the human genome is good, but our ability to interpret it was limited. "

- Eric Lander

One major task for bioinformatics research is to analyse genome-wide transcription from genomic data, such as microarray gene expression data and single nucleotide polymorphisms (SNPs) data. Due to the inherently complex behavior of biological systems, the genomic data analysis process normally consists of several stages. For example, the analysis starts with data preprocessing, followed by feature selection to find informative features (e.g. informative genes), then discriminates the classes of given samples by using different techniques (e.g., classification or clustering). This chapter gives a brief review of genomic data analysis and related biological background.

2.1 Gene Expression Data Analysis

This section presents some relevant biological knowledge relevant to the thesis, along with a brief introduction of some terminology and problem definitions.

2.1.1 Biological Background

In molecular biology, *cells* are the fundamental organisational units of all living organism systems. The *deoxyribonucleic acid* (DNA) is the nucleic acid that contains all the genetic instructions for functioning cells' activities in all living systems. A DNA module is a double-stranded polymer of basic molecular units called *nucleotides*. Each nucleotide is made of a deoxyribose sugar, a phosphate group and one of the four types of molecules called *nitrogen bases*. The four nitrogen bases found in DNA are *adenine*(**A**), *guanine*(**G**), *cytosine*(**C**) and *thymine*(**T**). The halves of the double helix structures are joined with the hydrogen bonds between nitrogen bases through *complementary base pairing* (**A** bonds only to **T**, while **C** bonding to **G**). For example, the occurrence of **A** on one strand must be coupled with the occurrence of **T** on the other strand. Similarly, if there is a **C** on one strand, a **G** will be always as a partner on the other. A double helical structure of DNA is illustrated in Figure 2.1.

DNA molecules play a main role of long-term information storage in all living organisms. A DNA sequence is a particular arrangement of the base pairs in a DNA strand (e.g., **ACAAGATGCC**), with the capacity to carry the exact instructions required to create a particular organism with its own unique characteristics. DNA is often called the blueprints of all living organisms, since it contains all the information required to construct and maintain the life from simple bacteria to complex human beings (Lu & Han, 2003). The properties characterised by the double helix structure of DNA molecules offer a special way to preserve and pass the information stored in DNA from one cell to another and from parental generation to their offsprings.

A complete DNA sequence that characterises a living organism is called its *genome*. The genome does not function as one genetic sequence, but is separated into a number of sections - *genes*. The size of genomes can be very different: the genome of Candidatus Carsonella ruddii (an obligate endosymbiotic Gamma Proteobacteria) contains only about $160,000$ base pairs of DNA, which is the smallest genome of living creature discovered so far, while the haploid human genome is approximately 3 billion DNA base pairs long and and has about $20,000 \sim 25,000$ genes (Minkel, 2006; wikipedia, 2009).

In contemporary biology, a *gene* is defined as "a locatable region of genomic sequence,

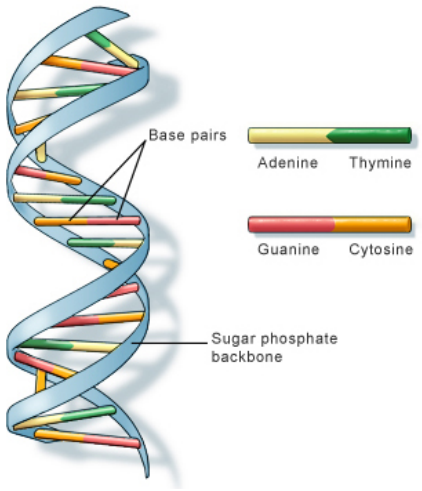


Figure 2.1: *A double helical DNA structure formed by base pairs attached to a sugar-phosphate backbone (U.S. the National Library of Medicine, 2009).*

corresponding to a unit of inheritance, which is associated with regulatory regions, transcribed regions, and/or other functional sequence regions" (Pearson, 2006). The physical development and phenotype of organisms are generally considered a product of genes interacting with each other. Taking into account complex patterns of regulation and transcription, genic conservation and non-coding RNA genes, an updated definition of a gene is thereby proposed by Gerstein et al. (2007): "A gene is a union of genomic sequences encoding a coherent set of potentially overlapping functional products".

There are two general types of genes in the human genome:

1. protein-coding genes:
   Protein-coding genes are the majority in the complete genome and are the templates for generating molecules - proteins. They are expressed in two stages: transcription and translation.

2. non-coding RNA (ribonucleic) genes:
   Non-coding RNA genes represent only $2 \sim 5\%$ of the total number of genes which provide the template for the synthesis for encoding functional RNA molecules. A large proportion of RNAs are involved in the control of gene expression, particularly protein synthesis.

An organised structure of DNA within a cell is a *chromosome*. Before cells dividing, *chromosomes* are duplicated in a process called *DNA replication* (Russell, 2009).

2.1.2 Gene Expression and DNA microarray Technology

DNA serves as a template not only for making copies of itself but also for producing a blueprint of a RNA molecule. A genome provides templates for the synthesis of a variety of types of Ribonucleic acids (RNAs) that may involve some most prominent examples of non-coding RNAs, such as messenger RNA (mRNA), transfer RNA (tRNA) and ribosomal RNA (rRNA) . RNA is a molecule consisting of a set of nucleotide units, each nucleotide consisting of a nitrogenous base, a ribose sugar, and a phosphate. Although RNA is very similar to DNA, the main differences are in the important structural details:

- Within a cell, DNA is usually double-stranded. By contrast, RNA is usually single-stranded;
- The sugar in DNA is deoxyribose, while the sugar in RNA is ribose that is the same as deoxyribose but with one more oxygen-hydrogen atom.
- RNA molecules have a much greater variety of nucleic acid bases, while DNA has only 4 different bases in most cases.

Over the last decades, a number of DNA array-based technologies have been developed for determining gene expression levels in living cells. There are a number of types of DNA arrays currently available for gene expression profiling. Two popular developed array technologies are summarised as follows:

- Oligonucleotide arrays:
  The main proponent of this technology is Affymetrix whose GeneChip arrays consist of small glass plates with a number of Oligonucleotide DNA probes recorded on the surface. Using this approach, massive number of mRNAs can be probed simultaneously. However, it is an expensive technology because specific equipment is required to manufacture and access genechips. A representative work using Oligonucleotide array data is presented by Golub et al. (1999) where it is used to classify bone marrow samples for cancer diagnosis.
- complementary DNA (cDNA) microarray:
  This is another solution for mRNA measurement developed by Stanford University, which is cheaper to manufacture and easy to read. Owing to the

non-proprietary right of this technology, cDNA is currently the most prevalent technology for microarray data analysis in academia. An important milestone achieved by using this technology was when Alizadeh et al. (2000) revealed a previously unknown sub-classification within diffuse large B-cell lymphoma (DLBCL) based on the analysis of cDNA microarray data.

There are two stages in which the expression of the genetic information stored in DNA molecule occurs: (Lu & Han, 2003):

1. *transcription* stage in which DNA molecule is transcribed into mRNA;
2. *translation* stage in which mRNA is translated into the amino acid sequences of the proteins for creating cellular functions.

Hence, *gene expression* is defined as the process of transcribing a gene's DNA sequence into RNA. During the transcribing, the information from genes is used in the synthesis of functional gene products (usually proteins). Functional RNAs are the products of transcribing non-protein coding genes, such as rRNA genes or tRNA genes. A gene's expression level indicates the approximate number of copies of the observed gene's RNA that are produced in a cell. Additionally, the level is in relation with the amount of corresponding proteins produced.

The measurement of gene expression has become an important part of life sciences research, owing to its ability to quantify the level at which a particular gene is expressed within a cell or an organism. The analysis based on such information can be a powerful tool for the study of the development in multicellular organisms and the identification of protein functions in single cells.

Empirical research has shown that specific patterns of gene expression occurring at different biological stages can cause response in tissues and cells (Russell, 2009). Therefore, gene expression level could be used to gauge the activity of a gene under specific biochemical conditions and can be very useful for:

- detecting virus infection in a cell;
- estimating the risk of an individual to develop cancer (oncogene expression);

- evaluating the cell's response to a drug.

Similarly, the analysis of the location of expression protein is a technical breakthrough that allows the analysis to be performed on an organismic or cellular scale. The measurement of localisation is particularly important for the development in multicellular organisms and as an indicator of protein function in single cells.

Microarray technology has emerged as one of the most powerful tools for measuring thousands of genome-wide expression levels simultaneously, especially in the realm of complex disease research (Ehrenreich, 2006; Draghici, Khatri, Eklund, & Szallasi, 2006). Complex and aggressive diseases, such as cancer, is known to be reflected in the mutation of certain genes. Normal cells can be mutated to malignant cancer cells under certain circumstances, e.g., the mutation in genes that influence the cell cycle, apoptosis, genome integrity, etc (Ben-Dor, Bruhn, Frideman, Schummer, & Yakhini, 2000).

Many microarray-based technologies have been developed for bioinformatics research over the last decades. They make it possible to observe the complex interactions among a large number of molecules, such as DNA, protein and combinatorial chemistry arrays, using a prespecified library of molecular probes (Baldi & Hatfield, 2002). Specifically designed for determining the expression levels of genes in living cells, DNA microarray (also known as DNA microarray chip) has taken a central stage in bioinformatics research, since it gives a possibility to investigate complex biological problems using some interdisciplinary approaches.

At a very basic level, a DNA microarray provides a snapshot of enormous amount of genes in a tissue sample. DNA microarray can be simply defined as "orderly arrangement of tens to hundreds of thousands of unique DNA molecules (probes) of known sequence" (Baldi & Hatfield, 2002, p7). A DNA microarray chip is produced by recording a large number of DNA segments (called *probes*) in spots arranged on a solid surface, such as a glass slide, a quartz wafer or a nylon membrane. Each spot is further labeled and hybridised to an array from a given objective interest, e.g., tumor biopsy (Huber, Von Heydebreck, & Vingron, 2003). The value yielded by measuring the labels of spots is then correlated to the abundance of the RNA transcript of the given tissue sample. The commonly used DNA microarray manufacturing methods for gene expression profiling include (Schena, 2000):

1. *In situ* synthesised oligonucleotide arrays (e.g., Affymetrix Inc.);
2. Pre-synthesised DNA arrays (e.g., Brown laboratory at Stanford University, Corning (NY, US) and Interactive (Ulm, Germany));
3. Filter-based DNA arrays (e.g., Clonetech.)

Although a variety of techniques have been proposed for analysing gene expression data, the field is still evolving and the developed methods have not reached a maturity level. Gene expression data can be analysed on three different levels (Baldi & Hatfield, 2002):

1. Single gene level. On this level, the analysis technique aims to identify whether each individual gene behave differently and isolatedly in an experiment;
2. Multiple gene level. Different clusters of genes are analysed to observe whether there exist common functionalities, interactions, co-regulation, etc.
3. The third level analysis attempts to discover whether the underlying gene and protein networks are responsible for observed patterns.

Many computational algorithms and models have been applied to gene expression data analysis. The typical algorithms and models used for analysis include k-means clustering, hierarchical clustering, principal component analysis (PCA) , self-organizing maps (SOM), decision trees, Bayesain networks, neural networks and support vector machine (SVM) , etc. There is no single method or algorithm that favor different gene expression data analysis tasks, because each method or algorithm has its own strength depending on the specific task and unique properties of the data to be analysed. In addition, microarray gene expression data is inherently high-dimensional, so that the outcome from data analysis is highly dependant on the methods of dimensionality reduction (known as feature selection in machine learning). The dimensionality reduction methods is one of the core parts in this research, and will be described in later chapters.

2.1.3 Recent Research in Microarray Gene Expression Data Analysis

Gene expression data analysis has become an indispensable part of system biology research. Currently, the majority of gene expression data research is conducted in the realm of cancer classification. Cancer diagnosis used to primarily rely on the histological appearances of tumours, which has been proved unreliable and inaccurate. Now the medical science community demands systematic and unbiased methods that are able to successfully classify cancers. Microarray technology has been consequently put forward as a new aid in treating various cancers and related complex diseases, owing to its ability of profiling differential gene expressions of tissue samples.

Over the last two decades, the remarkable progress achieved in microarray technology has helped researchers to further develop optimised treatment of cancer and other complex diseases, as well as the evaluation of prognosis based on genetic knowledge. Dozens of microarray research papers have shown that this technology is highly sensitive and efficient for detection and prognosis. For example, cDNA microarray is used to assess Parkinson disease samples and examine the drug intervention (Mandel, Weinreb, & Youdim, 2003). Microarray gene expression data has been employed in several studies of Alzheimer disease to predict different stages, including preclinical and prognosis stages (Galvin & Ginsberg, 2004; Galvin et al., 2005).

With the advance of microarray technology, biological data is being created and collected at a phenomenal rate (Beckers & Conrath, 2006). For example, the GenBank repository of nucleic acid sequences and the SWISSPROT database of protein sequences are doubling in size every 15 months on average (Benson et al., 2002). Contemporary bioinformatics research therefore needs assistance from computer science to design and implement new systems for data collection, storage, retrieval, analysis, etc. Nowadays, bioinformatics has become an integrated part of molecular biology and computer science to discover information and knowledge from the analysis of large-scale data, such as gene expression, protein expression and clinical data analysis (Luscombe, Greenbaum, & Gerstein, 2001).

Extensive studies have been carried out on classification problems related to complex diseases, such as cancer, in the last decades. New methods and systems are developed by statistical, computer science and biological research communities. However,

cancer classification using gene expression data is still a relatively new research area that has encountered many challenges due to its unique nature, such as:

1. How to develop effective and efficient algorithms and modelling systems for cancer classification?
2. How to exclude a large number of irrelevant features (genes) because the presence of these irrelevant genes can interfere with the discrimination power of relevant genes?
3. How to remove the technical noise that could be introduced at the stage of data collection or data pre-processing?
4. How to discover and interpret the important biological information with the use of gene expression data analysis?

2.1.4 Cancer - a Disease of Genes

Cancer is the result of cumulative genetic mutations disrupting the biological pathways, which results in the uncontrolled cell replication. Simply, cancer originates from a combination of an individual's genetic factors and influences from the surrounding environment and the personal history and lifestyle (DiChristina, 2008). The mutations affect two groups of cancer genes (Gibbs, 2003). One group is known as the tumor suppressors that normally restrain cells' ability to divide. The mutations may permanently disable these genes. The other group of genes are called oncogenes that stimulate the cell division, i.e. they prompt the tumor cells' growth.

Cancer arises because of "the accumulation of defects in certain classes of genes" (Bartek & Lukas, 2001, p1001). In 2008, more than 1.4 million people were newly diagnosed with cancer in the United States alone and cancer was the second leading cause of death in the United States and moving towards number one (*Cancer Facts & Figures 2008*, 2008). The statistics show that more than 500,000 Americans lost their lives to cancer in 2008, and almost one out of two men and one out of every three women will be diagnosed with cancer during their lifetime (Reuters, 2009).

The advent of microarray technology has made it possible to monitor the expression levels for thousands of genes simultaneously, which can help clinical decision making

in complex disease diagnosis and prognosis, especially for cancer classification, and for predicting the clinical outcomes in response to cancer treatment. Microarray technology offers a powerful tool for monitoring cancer prophylaxis and for clinical decision making (Kroczak et al., 2006).

A substantial number of methods and models for cancer diagnosis and risk management have been proposed. However, cancer is still thought of as an extremely frightening disease, as some types of cancer are still incurable and inoperable, such as epithelioid hemangioendothelioma (EHE) . The patients who have these type of incurable cancer are usually suggested "watch and wait" by doctors (Collins & Barker, 2008).

It is not a new idea that some specific gene mutations can increase the risk of a normal to develop into a tumor cell. In the late 1970s, John M. Bishop and Harold Varmus discovered that oncogenes existed in a wide variety of living organisms, including humans. They were awarded the Nobel Prize in Physiology or Medicine in 1989 for their discovery of the cellular origin of retroviral oncogenes. By early 2007, 350 cancer-related genes had been identified and since then plenty of insights into this disease have been reported (Collins & Barker, 2008). However, different genes cause the disease in different people, thus there is the need for personalised modelling.

Following the discovery of these cancer genes, treatment strategies based on specific gene mutations have been extensively studied in the medical research area. A number of new gene-based drugs have been invented for different types of cancers, e.g., Gleevec™ - the drug for complex malignancies treatment has been proved effective against chronic myelogenous leukemia (Denis, 2008; Henkes, Kuip, & Aulitzky, 2008). Another example of genetic information based personalised medicine is Iressa®. It can significantly benefit a small population of patients with non-small-cell lung cancer who have not responded to other treatments with both platinum-based and docetaxel chemotherapy (Tamura & Fukuoka, 2005). Genome-wide expression data analysis using microarray technology has an important role to play for the better understanding of complex human diseases, especially for cancer diagnosis and prognosis. The knowledge discovered from gene expression data analysis experiments brings a new paradigm for further developing new therapeutic approaches and identifying novel diagnostic biomarkers.

2.1.5 Microarray Data Analysis for Cancer Research

A substantial number of research studies have shown that microarray gene expression data analysis could be in some cases 100% sensitive and specific to detect cancer and predict prognosis, such as the ovarian cancer study (Petricoin et al., 2002; Zhu et al., 2003). Microarray technology is considered revolutionary for studying complex human diseases and has been claimed that "all human illness can be studied by microarray analysis, and the ultimate goal of this work is to develop effective treatments of cures for every human disease by 2050" (Schena, 2002).

However, there is an increasing concern that many published research findings from microarray gene expression data analysis experiments are not reproducible. This issue has been addressed as as one of the most important bias problems in microarray research (Ioannidis, 2005; Ransohoff, 2005a), and has become a big threat to the reliability of contemporary bioinformatics research for cancer gene data analysis (Ransohoff, 2005b; Eklund & Szallasi, 2008). Marshall (2004) disputed the reliability of the outcomes of microarray experiments: "Thousands of papers have reported results obtained using gene arrays, ... But are these results reproducible?".

Thus, reproducibility of microarray experiments becomes a big concern for microarray gene expression data study for contemporary cancer research. One example is the study of proteomic microarray data for ovarian cancer diagnosis: Petricoin et al. (2002) and Zhu et al. (2003) claimed that their methods could accurately identify ovarian cancer using proteomic data. However, Baggerly, Morris, Edmonson, and Coombes (2005) questioned their approaches because he and his colleagues were unable to reproduce highly accurate results reported in the paper (Petricoin et al., 2002). Regarding this issue, Petricoin suggested that other researchers should communicate the original data provider to correctly process data if they intended to have a meaningful analysis of reproducibility.

Recently, the academic community has recognise that evaluation criteria must be established to ensure researchers to choose proper methodologies leading to more efficient and reliable outcomes. Consequently, plenty of literature has been so far published focusing on the solution to improve the validity of microarray data analysis experiment from different aspects, including estimating bias error, using unbiased validation schemes and better laboratory controlling techniques (Eklund & Szallasi,

2008; Allison, Cui, Page, & Sabripour, 2006; M. Zhang et al., 2008; Varma & Simon, 2006; Shi, Perkins, Fang, & Tong, 2008).

2.2 Single Nucleotide Polymorphisms (SNPs) Data Analysis

2.2.1 Single nucleotide polymorphisms - SNPs

SNPs genotypes are of great importance for understanding of the human genome, and are the most common genetic variations between human beings. An example of a SNP can be the alternation in the DNA segment AAGCCTA to AAGCTTA, where the fifth nucleotide - 'C' in segment 1 is replaced with a 'T' in segment 2. Figure 2.2 demonstrates a SNP occurring in two DNA fragments from different individuals.

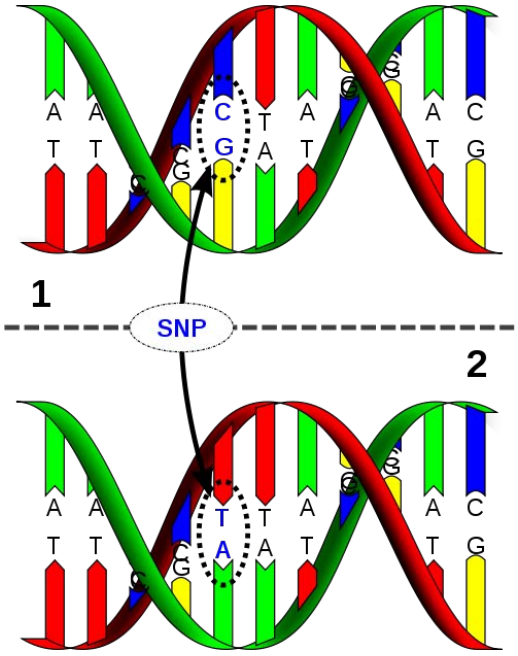


Figure 2.2: *DNA molecule 1 differs from DNA molecule 2 at a single base-pair location (a C/T polymorphism), copied from (Hall, 2007)*

On average, SNPs occur in nucleotides at the rate of $3 \sim 5\%$, which means there are approximately 10 million SNPs in human genome. SNPs are found in the DNA

among genes, and more of them have no effect on human health or disease development. However, when SNPs occur within a gene or in a regulatory region near a gene, they may have a direct impact on disease development through affecting genes function. Therefore, some SNPs act as biomarkers that allow scientists to locate the genes associated with disease.

Research has shown that some of these genetic variances are very important in the study of human health (Kato, Kamoto, Hyuga, & Karube, 2007). The variations in the human DNA sequences may paly an important role in disease development by affecting genomic functions, e.g. influence the development of diseases and the response to drugs, chemicals, pathogens, etc. Moreover, SNPs are thought to be a key factor in understanding the concept of personalised medicine (Carlson, 2008).

At present, there is no effective way to measure how a patient will respond to a particular drug treatment. In many cases, a treatment can be effective for a group of patients, but is not effective for others at all. Findings related to SNPs can help researchers build clinical decision support systems that predict an individual's response to certain drugs and environmental factors (e.g. toxins) and the risk of particular disease development. Also, SNPs offer a new way to track the inheritance of disease genes within societies, especially for studying complex diseases, such as Coronary heart disease, cancer and diabetes.

It is generally agreed that the most efficient way to associate a SNP with phenotype is through a genome-wide association (GWA) study. With GWA scans, hundreds of thousands, or even millions can be screened using DNA microarray technology, also known as *SNP array*. The first SNP array was developed in 1998, containing only 558 loci (Wang et al., 1998). The SNPs in the sample were amplified in a single multiplex polymerase chain reaction that contained primer pairs for different loci (Boyd, Mao, & Lu, 2009). Amplified DNA was then hybridised on a SNP array to analyse the genotype of 558 SNPs. A challenge for information science is to develop efficient methods for personal SNPs data analysis.

2.3 Conclusion

This chapter has briefly reviewed genomic data analysis in bioinformatics study. It has introduced biological background and some commonly used terminology related to this research. It has also identified some issues in microarray data research, such as the reproducibility of the microarray data experiments and bias issues occurring in experiments. It has posed an open question to be discussed and answered in this study:

- How to create an efficient framework and a system for developing efficient clinical decision support system using personal genomic data?

To deal with this problem, the next chapter will discuss some computational intelligent models and systems that will be used in this thesis.

CHAPTER 3

Computational Intelligence: Methods and Systems

"Intelligence is a basic property of life "

- J. W. Atmar

This study focuses on the development of personalised modelling for gene data analysis and biomedical applications using evolving intelligent methods and systems. We hereby give an introductory overview of some popular computational intelligent methods and systems that will be used throughout the thesis. Computational intelligence is a branch of computer science that develops methods and systems that are considered to be in some sense, intelligent, for solving a variety of complex problems in the areas of science and engineering area. The methods and systems of computational intelligence embrace the techniques from statistical methods, neural networks, fuzzy systems, evolutionary computation, swarm intelligence, evolving connectionist systems, etc.

In order to provide more precise information for data analysis, personalised modelling creates a unique model for each data sample. This type of research problems need the algorithms and models that are able to adapt to new data sample and

evolve the structure of learning system. In literature, evolutionary computation and evolving connection system are often suggested to be the good choices for solving the problems that need adaptive and evolving learning, owing to their capability of evolving candidate solutions towards optimal target (Michalewicz & Fogel, 2004; Kasabov, 2007a). In computer science, evolutionary is an iterative progress related to the development of populations of individual systems. Such process is usually inspired by the biological mechanism of evolution. Evolving computation may include evolutionary process, because the evolutionary processes do require evolving and adaptive development of single individuals. This chapter gives a brief review of these two computational techniques and related algorithms that will be used for personalised modelling in this thesis.

3.1 Evolutionary Computation

This section provides some insights into the applications of the most commonly used algorithms and models in the field of evolutionary computation. The experiment part demonstrates the implementation of some extensively studied algorithms of evolutionary computation for solving a benchmark problem.

3.1.1 Introduction to Evolutionary Computation

Evolutionary computation is a subfield of artificial intelligence that usually involves combinational optimisation problems. Basically, evolutionary computation uses iterative progress where populations of individuals are evolved during the development. Evolutionary computation is inspired by the biological mechanism of evolution, and uses intelligent computational techniques to mimic Darwinian principles for solving optimisation problems.

The understanding of evolution was advocated by Charles Darwin and Alfred Russel Wallace in their joint publication (Darwin & Wallace, 1858) in which compelling evidence was presented for the theory of evolution. The early attempts to use evolutionary theory for automated problem solving date back to the 1950s. From the observation of the famous Turing test, Turing commented on “an obvious connection between the process (the test for artificial intelligence) and evolution” (Turing, 1950,

p450). Friedman (1959) recognised that artificial intelligence ("thinking machines") can be fulfilled by a simulation of mutation and selection. The study carried out by Friedberg and his colleagues (Friedberg, 1958; Friedberg, Dunham, & North, 1959) was a pioneer trial to implement simulated evolution for finding solutions to optimisation problems. In their work they focused on the improvement of a machine language computer program through an evolved learning strategy. However, the limitation of this work lay in the difficulties of choosing highly interactive separate program instructions. Another important work during the early stages of evolutionary computation was the Bremermann's experiment (Bremermann, 1958). He indicated that the principle of evolution is " most useful as a key to the understanding of creative thinking and learning" (Babovic, 1996, p118) and conjectured that evolution could be an efficient tool for solving optimisation problems.

3.1.2 Main Methods and Techniques for Evolutionary Computation

Although simulated evolution has a long history, it was only recently that the current main techniques for evolutionary computation were formalised. Evolutionary algorithm and swarm intelligence are probably the most popular and representative techniques for evolutionary computation. Evolutionary algorithm is a population-based optimisation algorithm firstly introduced by Fogel in 1960s (L. Fogel, Owens, & Walsh, 1966). Candidate solutions to the target optimisation problem represent the individuals in a population, and a fitness function evaluates the candidates and determines which solutions will survive. Then the heuristic process evolves the above steps until terminating conditions are reached. Distinguished by the implementation details and the target of particular applied problems, genetic algorithm (GA) , evolution strategy and evolutionary programming are the three major methods (techniques) used in evolutionary algorithms design.

3.1.3 Genetic Algorithm (GA)

GA might be the most popular technique that has been used for implementing evolutionary algorithm. GA has been extensively explored for solving complex practical

problems and as computational models for constructing natural evolutionary systems (M. Mitchell, 1996), since it was developed by Holland (1975) in the early 1970s. Most commonly, genetic algorithms are mainly adopted as an evolutionary model for finding the exact or approximately best solutions to optimisation problems in science and engineering.

The classical form of genetic algorithm involves three types of operators: selection, crossover and mutation. Selection is an operator that selects individuals in the population for reproduction. The higher the fitness of the individual is, the more chances it has to be selected. The second operator is crossover that randomly determines a locus at the parent individuals, and then swap the subsequences at the locus of parents individuals to generate two offsprings. The third operator is mutation that randomly flips some bits in an individual. The simplest mutation is one bit flipping, e.g. the individual (chromosome) 10010101 might be mutated at the third position to create an offspring 10110101. Mutations should occur with a very low probability (*e.g.* 0.001), otherwise they will disrupt the fitness of the overall population. Figure 3.1 illustrates these two operators.

Genetic encoding

Encoding candidate solutions (*individuals*) is one of the major factors that impacts a GA performance. The efficiency of a GA's search usually depends very much on the choice of an appropriate encoding way to the populations of chromosomes. The simplest way to encode the chromosome is to employ binary bit-value. Binary encoding uses a binary value (either 0 or 1) to represent the possible values of the genes in the chromosome. Binary encoding is usually effective and works well in a simple searching problem space. However, using binary encoding can be very difficult when the optimisation involves complicated encoding, such as real values, category data, etc. In addition, for some optimisation problems requiring domain knowledge, binary encoding cannot be well adapted. Thus, other more sophisticated encoding techniques have been developed for different types of optimisation problems, such as permutation encoding, real-value encoding, tree encoding, etc.

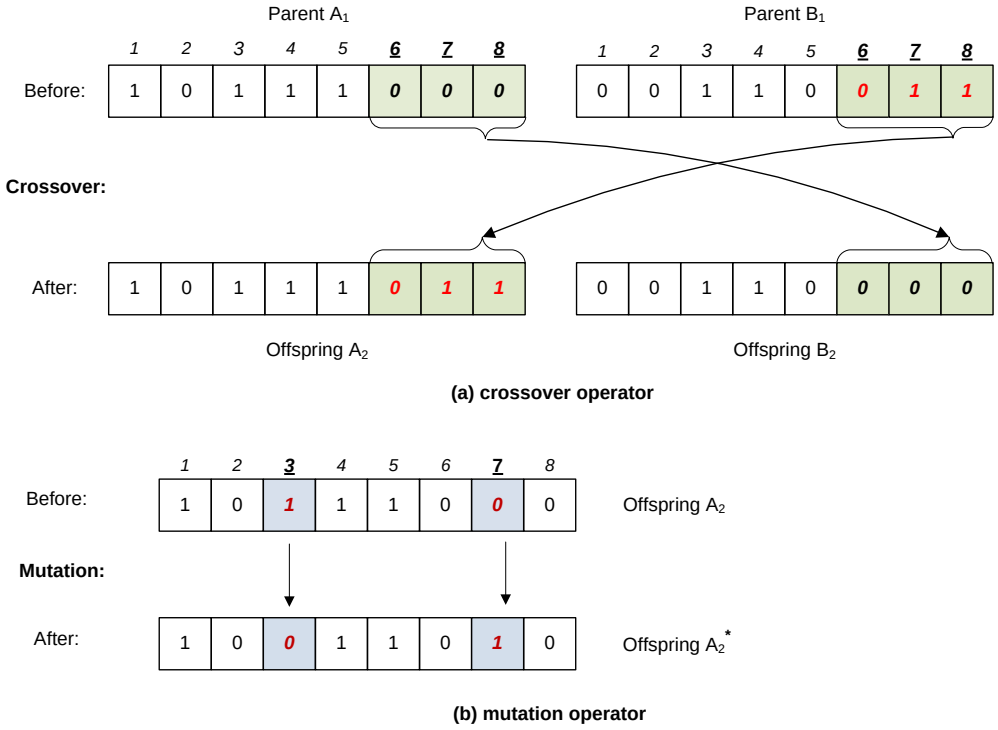


Figure 3.1: *The illustration of crossover and mutation operators. (a) The crossover operator chooses the* 6^{th} *bit as the locus. Parents* A_1 *and* B_1 *swap three bits' value starting from bit6 to produce offsprings* A_2 *and* B_2*. (b) Mutation occurs at the position of bit 3 and 7 in individual* A_2 *where the bit value is flipped.*

Selection

A common selection method in GAs is *fitness-proportionate selection* (M. Mitchell, 1996), which replicates the natural selection principle - "fittest to survive", i.e., a fitter individual will tend to have a higher probability of be selected to produce the next generation. Roulette-wheel sampling (Goldberg, 1989) is one of the most popular methods for fitness-proportionate selection. The method offers each individual a chance to be selected based on its chromosome string fitness value. Suppose we have a randomly created generation of individuals (population size $\mu = 4$) as follows:

Chromosome label	Chromosome	Fitness	Percentage of Total(%)
A	000101	3	4
B	010001	17	24
C	001010	10	14

D	101001	41	58

The fitness is measured by the sum of individual's bit string. The concept of roulette-wheel selecting method is illustrated in Figure 3.2.

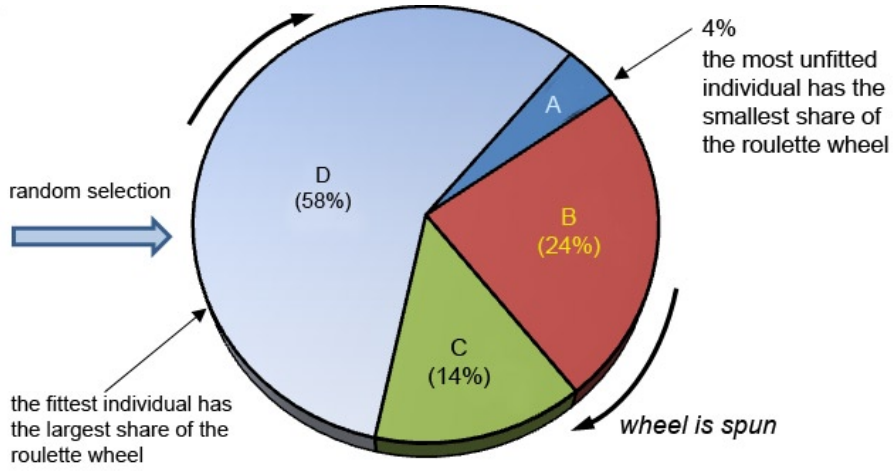


Figure 3.2: *The illustration of roulette-wheel selection based on fitness.*

The number of times that the roulette wheel will be spun is equal to the population size. Since in the example the population size is 4, the roulette wheel will spin 4 times. The first two spins might select individual B and C as the parents, and the next two spins might select B and D. If the roulette wheel spins many times (usually at least several hundreds), the selection is clearly biased towards fitter individuals.

Other selection methods include: Elitism method (De Jong, 1975) retains some of the fittest individuals at each generation. Rank selection is a method that selects the individuals based on their rank rather than their absolute fitness (Baker, 1985). Steady-State selection is often used in evolving rule-based GA systems (J. Holland, 1986), where a small number of the most unfitted individuals are replaced by the offsprings from GA operations of the fittest individuals.

A simple GA

Typically, a simple GA starts with a random population of encoded candidate individuals (also known as chromosomes). Chromosomes are encoded in binary bit-

streams in which each bit is denoted by either 0 or 1. The evolution then starts with a population of randomly generated chromosomes. In each generation, a fitness function evaluates all chromosomes in the population. Chromosomes are stochastically selected from the current population based on their *fitness* and will be recombined through crossover and mutation to form the offsprings for the next generation. The *new generation* will be evolved in the iterative process that usually involves $1,000$ or several thousands iterations. A GA terminates when at least one of the following conditions is met:

- the maximum number of generations has been produced
- a solution is found that satisfies the pre-specified fitness level
- a highest fitness level is reached

The pseudo code for a classical (simple) GA is given in Algorithm 5 in Appendix A.

Plenty of published work has shown that GAs are capable of solving difficult optimisation problems through an appropriate choice of candidate individuals in the searching space and efficient operators (M. Mitchell, 1996). The successful practical implementations of GAs found in literature include: applications in computer programming and engineering optimisation problems (Forrest & Mayer-Kress, 1991; Krishnakumar & Goldberg, 1992), rule-based classification systems (Liepins, Hilliard, Palmer, & Rangarajan, 1989), artificial life simulation (J. H. Holland, 1992), and parallel computing (Muhlenbein, Bendisch, & Voigt, 1996; Lazarova, 2008).

3.1.4 Evolution Strategy

Evolution strategy was developed by Rechenberg (1973) and Schwefel (1974) for evolving optimal shapes of minimal drag bodies in a wind tunnel using an evolution-inspired principle. Evolution strategy can be applied for a variety of optimisation problems, including continuous, discrete and combinatorial search spaces with or without constraints (Schwefel, 1981, 1995). Since evolution strategy is mainly applied to parameter optimisation problems, real-valued encoding is usually employed for representing candidate solutions (individuals/chromosomes). Each individual

contains a number of strategy parameters that are used to control the behavior of mutation operator during the evolution.

An optimisation problem can be presented as follows:

$$\mathbf{y}^* = \underset{\mathbf{y}\in\mathcal{Y}}{\operatorname{argopt}}\, \mathbb{F}(\mathbf{y}), \tag{3.1}$$

where $\mathcal{Y}$ is a search space, and $\mathbb{F}$ is the function to be optimised. One typical example of $\mathcal{Y}$ is a real-valued *n-dimensional* search space $\mathbb{R}^n$.

Evolution strategy primarily applies mutation and selection to a population of individuals to evolve solutions iteratively. At the very beginning (generation *gen*=0), evolution strategy randomly generates a population(μ) of individuals $(\alpha_1, \ldots, \alpha_\mu)$. To create the new generation, λ offsprings are bred from the set of parent individuals $(\alpha_1, \ldots, \alpha_\mu)$. The parental individuals are randomly selected, which means the selection is independent of the parent objective function $\mathbb{F}$. Each individual α_i consists of not only the objective function $\mathbb{F}_i = \mathbb{F}(\mathbf{y}_i)$, but is usually defined by a few parameters (known as *endogenous strategy parameters*) $\mathbf{s}_i$:

$$\alpha_i = (\mathbf{y}_i, \mathbf{s}_i, \mathbb{F}(\mathbf{y}_i)) \tag{3.2}$$

where i is the the individual's index in the population.

The size of λ should be unequal to the size μ of the parent population. The offspring population is generated by the method that can be mathematically formulated by:

$$(\mu/\rho \ \overset{+}{,}\ \lambda) - evolutionstrategy \tag{3.3}$$

where ρ is the number of individuals to be involved in the offspring reproduction, and the "$\overset{+}{,}$" denotes two types of selection, *plus selection* and *comma selection*, respectively. The strategy-specific parameters μ, λ and ρ are called "*exogenous* strategy parameters" and are kept constant through the evolution process (Beyer & Schwefel, 2002).

The selection in evolution strategy gives the evolution a direction in which only the fittest individuals get the chance to reproduce. The parents are deterministically selected (i.e., deterministic survivor selection) from the multi-set of either the offspring referred to as comma-selection ($\mu < \lambda$ must hold), or both the parents and offspring,

referred to as plus-selection. Two kinds of selection techniques - *comma* and *plus* selection are commonly employed in evolution strategy depending on whether the parental population is included or not during the selection process.

In the case of *comma* selection (μ, λ), the individuals of parent population are excluded for recombining the new generation even if they have higher fitness value than all offsprings. The selection pool size here is λ. Such selection schema requires $\lambda > \mu$ to drive the evolving process towards an optimal solution. If $\lambda = \mu$, the evolution would not work because all the offsprings would be selected as parents, which would result in the selection providing no search-revelent information (Beyer & Schwefel, 2002).

In contrast to *comma* selection, *plus* selection$(\mu+\lambda)$ takes the parent individuals into account. It selects the individuals for the new population not only from λ offsprings but from μ parent individuals, i.e., the size of selection pool is $\lambda + \mu$. Hence, there is no restriction on the size of offspring population (λ). The special case of $\lambda = 1$ is notated as "*steady-state*" evolution strategy. Plus selection promises the survival of fittest individuals.

Each selection techniques is favoured for evolution strategy implementation in different application areas. Comma selection is suggested to search unbounded spaces $\mathcal{Y}$ (Schwefel, 1987), while plus selection is recommended for searching discrete finite problem spaces, especially for combinatorial optimisation problems (Herdy, 1992).

The prime genetic operator in evolution strategy is mutation. The design of mutation operator is problem-dependent. It usually applies a normal distribution probability function to each component of an individual. The mutation process is often controlled by some strategy parameters, e.g., the mutation strength parameter. A simple evolution strategy algorithm is given in Algorithm 6 in Appendix B.

3.1.5 Evolutionary Programming

Evolutionary programming was originally developed by Fogel (1962) in a simulated evolution for investigating artificial intelligence. The individuals in evolutionary programming are often encoded by real numbers. The evolution is simply driven by the mutation operator that commonly adopts a probability distribution function to op-

timise objective variables. Evolutionary programming is principally applicable to all areas where evolutionary algorithms can be implemented. Evolutionary programming has been used in a range of combinatorial optimisation problems in different areas, e.g., pharmaceutical design (Duncan & Olson, 1996), molecular docking analysis (Gehlhaar & Fogel, 1996), cancer diagnosis study (D. Fogel, Wasson, Boughton, & Porto, 1997, 1998), control systems modelling (Jeon, Kim, & Koh, 1997) and system identification (D. Fogel, Fogel, & Porto, 1990).

3.1.6 Comparison of Three Methods: GA, Evolutionary Strategy and Evolutionary Programming

The three main types of evolutionary algorithms - GA, evolutionary strategy and evolutionary programming are broadly similar in principle, though they have significant differences in terms of implementation. The individuals of the population are fixed-length-string based in all three algorithms. However, evolutionary strategy and evolutionary programming commonly use real-valued encoding for individual representation, while GA generally adopts binary bitstream encoding schema. The prime genetic operator in GA is recombination (crossover), while mutation is the main driving force for evolutionary strategy and evolutionary programming. Evolutionary programming differs from GA and evolutionary strategy in that it does not use recombination operator (crossover), and its evolution is entirely dependent on mutation. The three algorithms also differ in the type of selection control: the selection in GA and evolutionary programming is probabilistic, while it is deterministic in evolutionary strategy.

3.1.7 An Implementation of GA: Compact Genetic Algorithm

Compact genetic algorithm (cGA) (Harik, Lobo, & Goldberg, 1999) is an optimisation algorithm that represents the population as a probability distribution over a set of solutions with a specified population size and selection rate. Compact genetic algorithm can be an alternative GA solution for complex optimisation problems, because it requires much less computational power than a simple (classical) GA (sGA)

. This algorithm will be used to construct the systems and models for personalised modelling in this research. Therefore, below we explain the basic principle of cGA and design an experiment to demonstrate the evolving process of cGA on a simple benchmark problem in the following section.

The Principle of Compact Genetic Algorithm

In compact GA, the evolving process is driven by the iterated competitions between two candidate individuals and tends to converge towards a near-best solution. The algorithm starts with a randomly created probability vector to be used for generating a population of individuals. Two individuals from the current population compete with each other and the winner will survive. cGA then makes the decision to select the winner from these two competitors according to their fitness evaluated by a fitness function. The winner's information will be used for producing the next generation, and the process will repeat until the terminating criterion is reached.

Suppose there is a task pertaining to finding an optimal solution. Firstly, cGA randomly creates a probability vector p with l bits where each bit represents the probability that it will be selected or not. The bigger the bit value, the higher probability the bit to be selected. From the very beginning, each bit must have the equal probability of being selected or not, i.e. all bit values should be 0.5. Hence, the probability vector p should look like: [0.5 0.5 0.5 ... 0.5].

Two individuals A and B are randomly generated from the probability vector p, and may look like: [*0.41 0.63* 0.52 0.50 ... 0.82]. Each bit denotes the probability of the gene to be selected or not, the larger the value, the higher the probability for the gene to be selected. For example, bit 1 indicates less likely to be selected ($0.41 < 0.5$), while bit 2 with value (0.63) indicates a higher probability to be selected. Based on such assumptions, two individuals a and b are updated by comparing their bit value with probability vector p. If a bit value is larger than 0.5, then it becomes 1, otherwise 0. For example, bit 1 will be 0 after the comparison, while bit 2 will be 1. So far, cGA has generated two individuals a and b with all bits either 1 or 0. Let A and B compete, and cGA makes the decision which one is the winner according to the evaluation by a fitness function. Probability vector p is then updated to produce the next generation based on the competition result through the following updating strategy: check whether $winner(i) = loser(i)$

if they are same, then there is no need to update the i^{th} bit in vector p;
otherwise do the following updating for $p(i)$:

if winner(i)==1 **then**
  $p(i) = p(i) + \frac{1}{\mu}$
else
  $p(i) = p(i) - \frac{1}{\mu}$
end if

where μ is the population size. The probability vector p is checked whether it has converged in each generation. It has converged when each bit value is either 1 or 0. Once p is converged, it represents the optimal solution. Otherwise, cGA repeats the process from the step of generating two new individuals. Algorithm 7 illustrates the form of a cGA in Appendix C.

3.2 Evolving Connectionist Systems (ECOS)

Evolving computation is a general term that denotes several computational techniques in relation with evolving process where a modelling system is able to adapt to changes. The term 'evolving' is often thought to have the same meaning as the term 'evolutionary', and they do have quite a lot of overlap to some extent. However, they should be distinguished in terms of designing different problem solutions. Evolving process is a process that "is mainly concerned with the development of the structure and functionality of an individual system during its lifetime" (Kasabov, 2007a, p3). Evolving process is further defined by Kasabov as "a process that is developing, changing over time in a continuous manner" (Kasabov, 2003, p7). Evolutionary is concerned with the development of a population of individual systems evolved over generations (J. H. Holland, 1992; Kasabov, 2003).

Evolving intelligent system is an information system that "develops its structure, functionality, and knowledge in a continuous, self-organized, adaptive, and interactive way from incoming information, possibly from many sources, and performs intelligent tasks typical for humans thus improving its performance" (Kasabov, 2007a, p9). The distinction of evolving intelligent system is that it emphasises the dynamic and knowledge-based structure and adaptiveness to the new coming information.

An evolving process is difficult to model because:

- There might be no prior knowledge for some parameters;
- Unexpected changes may happen at a certain stage during development
- The results may not be strictly predictable in long term

Hence, to model an evolving process is a challenging task that needs well designed applications in life and computing sciences. The most typical example of an evolving process is life. Modeling living systems require continuous and adaptive changes and at the same time preserves some features and principles in a life long way. The representative work for evolving modelling system is evolving connectionist systems (ECOS) developed by Kasabov (1998).

3.2.1 Principles and Architectures of ECOS

Evolving connectionist systems (ECOS) are defined as "multi-modular connective architectures that facilitate the modelling of evolving processes and knowledge discovery" (Kasabov, 2003, p26). An evolving connectionist system consists of a collection of neural networks (can be a single one) that work continuously and evolve their structure and functionality through a dynamic interactions within the system itself or with other systems. Generally, an evolving connection system involves the following functional parts (Kasabov, 2003):

1. Data acquisition.
2. Data pre-processing and feature evaluation.
3. Connectionist modelling.
4. Knowledge discovery.

ECOS learn local models from data through a set of clusters, each being associated to a local output function. The creation of clusters is based on the similarity between data samples in the input space or in both input and output space. The former

case is shown in some models of ECOS, such as the dynamic neuro-fuzzy inference system DENFIS model (Kasabov & Song, 2002), and the latter is shown in the model of evolving fuzzy neural network (EFuNN) (Kasabov, 2001). Let $X = \{x, y\}$ be a sample, and $r = (\omega_1, \omega_2)$ be an existing rule node defined by two vectors of connection weights - ω_1 and ω_2, thus the similarity between X and r can be measured by a normalised Euclidean distance:

$$d(X, r) = \frac{\sum_{i=1}^{n}(x_i - \omega_1(i))^2}{n} \tag{3.4}$$

where n is the number of input variables. Given a threshold R_{max}, those samples that have a distance to an existing cluster center (rule node) r less than R_{max} are allocated into the same cluster. New clusters are formed by the samples that fall into the cluster r. Cluster centers are continuously adjusted and new clusters are incrementally created when new data samples come.

ECOS learn from data and consequently create a local output function f_c for each cluster. For a given data vector x, the output function f_c creates the local models represented by a set of rules with clusters as:

if $x \in r$, ***then*** the output is calculated by f_c

3.2.2 Evolving Fuzzy Neural Networks (EFuNN)

EFuNN (Kasabov, 2002) is a connectionist model with neuro-fuzzy inference systems for implementing ECOS. EFuNNs are fuzzy neural network structures that evolve based on ECOS principles. Fuzzy neural networks are connectionist structures that can be interpreted by a set of fuzzy rules and a fuzzy inference system (Roger Jang, 1993; Lin & Lee, 1996). EFuNN has a five-layer structure in which all nodes represent *membership functions* (MF) and can be modified during learning. Figure 3.3 illustrates an example of an EFuNN with a short term memory and feedback connections. The detailed algorithm for evolving EFuNNs from incoming data vectors is illustrated in Appendix D.

The input layer is the first layer that contains input variables. The second layer is a fuzzy input layer where each input variable is represented by a group of neurons.

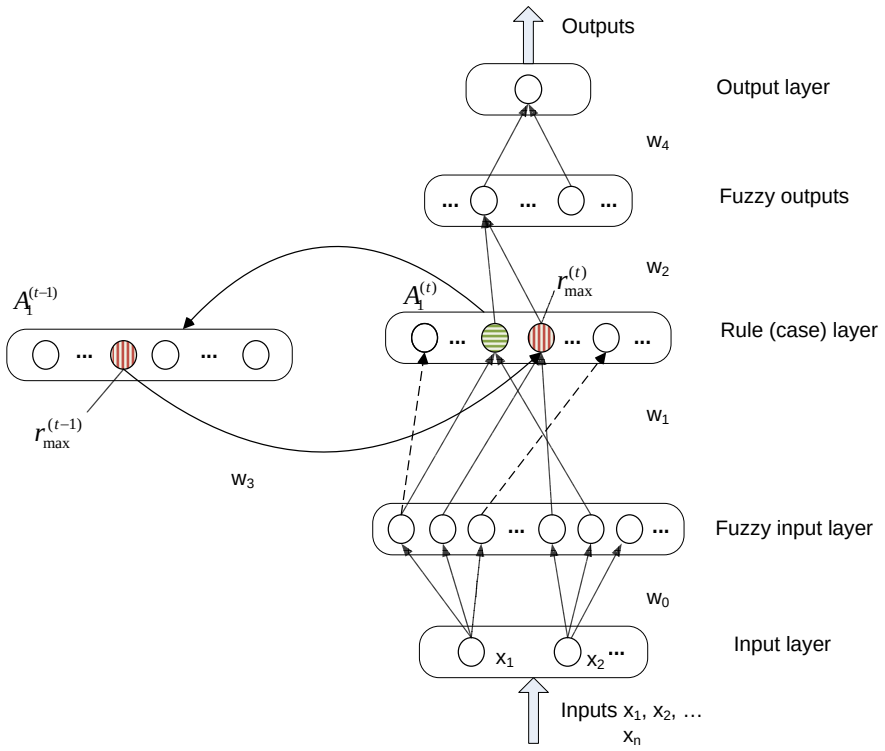


Figure 3.3: *An example of an EFuNN with a short term memory and feedback connections, adapted from Kasabov (2001)*

These neurons denote the fuzzy quantisation of the input variable, e.g. three neurons can be used to represent "best", "good" and "bad" fuzzy values of a variable. Different MFs can be attached to the neurons, such as triangular or Gaussian MF. This layer aims to transfer the input variables into membership degrees to which they belong to the corresponding MF. Within this layer, new neurons are created, when the corresponding variable value of a given input vector does not belong to any of the existing MFs. An optional short-term memory layer can be introduced through feedback connections from the rule node layer.

Rule (case) layer is the third layer in EFuNN which contains rule nodes that evolve through supervised or unsupervised learning. The rule nodes represent prototypes of the associations between input and output data. Each rule node r is defined by two vectors of connection weights: $\omega_1(r)$ and $\omega_2(r)$. The former is adjusted by an unsupervised learning model based on the similarity measurement within a local problem space, while the latter is adjusted by a supervised learning model based

on the estimation of output error. The neurons in fourth layer represents the fuzzy quantization for the output variables. Finally, the fifth layer gives the value of the output variables.

Evolving classification function (ECF) is a simple implementation of ECOS that is used in this study. The learning algorithm of ECF is described in Appendix E.

3.3 Support Vector Machine (SVM)

Support vector machine (SVM) is a popular algorithm used for the creation of learning models in machine learning. A SVM model consists of a set of vectors described by a kernel function that separates the data samples belonging to different classes (the vectors are called support vectors). SVM has been widely employed to build models for machine learning problems (Vapnik, 1998; Shah, Oehmen, & Webb-Robertson, 2008; Q. Wu, 2009). In many cases, SVM models can be efficient classification models and produce reliable results (Bozic, Zhang, & Brusic, 2005).

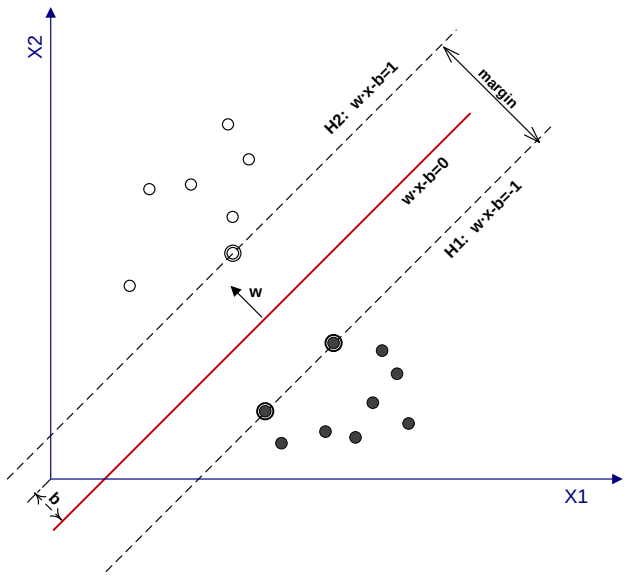


Figure 3.4: *An example of the linear separating hyperplanes in SVM. Note: the support vectors are encircled*

Support vector machine (SVM) was firstly introduced by Vapnik in the mid-1960s. It has been successfully applied in different fields of computer science and engineering

for classification and regression problems (Burges, 1998). Given a bi-class problem in an *m-dimensional* space, a SVM builds a separating hyperplane in that space, which aims to maximise the margin between the two groups of data sets. The training data is given as $X = \{x_i, y_i\}, i = 1, \ldots, n, y_i \in \{-1, 1\}, x_i \in \mathbb{R}^m$, where x_i is an *m-dimensional* data vector, y_i is the corresponding class label. Assume there exist some hyperplanes that separate positive (label '+1') and negative (label '-1') samples. The data points x_i falling on such a hyperplane should satisfy:

$$w \cdot x_i + b = 0 \tag{3.5}$$

where w is a normal vector perpendicular to the hyperplane, a parameter $|b|/|w\|$ specifies the perpendicular offset from the hyperplane to the origin, and $\|w\|$ is an Euclidean normal vector of w. The shortest distances from the separating hyperplane to the closest positive and negative data points are denoted by d_+ and d_-, respectively. Let d_+ and d_- be the "margin" of a separating hyperplane. Then, the given problem is simplified by using a SVM algorithm to find the separating hyperplane with the largest margin. If the training data are linearly separable, all the training data samples should satisfy the following constraints:

$$x_i \cdot w + b \geq +1, \forall y_i = +1 \tag{3.6}$$

$$x_i \cdot w + b \leq -1, \forall y_i = -1 \tag{3.7}$$

They can be further combined and written as:

$$y_i(x_i \cdot w + b) - 1 \geq 0, \forall i \in \{1, 2, \cdots, n\} \tag{3.8}$$

The data points satisfying the equality in Eq.3.6 will fall on the hyperplane $H1 : x_i \cdot w + b = +1$, with vector w and perpendicular distance from the origin $|1 - b|/\|w\|$. In the same way, the data points satisfying the equality in Eq.3.7 will fall on the hyperplane $H2 : x_i \cdot w + b = -1$, with vector w and perpendicular distance from the origin $|-1-b|/\|w\|$. The margin can be calculated by $2/\|w\|$, as $d_+ = d_- = 1/\|w\|$. Thus two parallel hyperplanes $H1$ and $H2$ are constructed, and there are no data points lying between them. Consequently, the pair of hyperplanes giving the maximum margin through minimising $\|w\|^2$ will be found and subjected

to Eq.3.8. Finally, an optimal separation can be achieved by the hyperplane that has the greatest distance to the neighbouring data points of both classes, as is illustrated in Figure 3.4. The data points are referred as *support vectors*, if they satisfy the equality in Eq.3.6 or 3.7 and their removal would change the solution to the discovered hyperplane. In Figure 3.4, support vectors are indicated by *extra circles*. Generally, the larger the margin, the lower the generalisation error of the classifier (Burges, 1998).

For nonlinear classification problems, a kernel function is introduced into SVM to find the maximum-margin hyperplane (Boser, Guyon, & Vapnik, 1992). The SVM based classifiers can be mathematically formulated by:

$$y(x) = \text{sign}\left[\sum_{i=1}^{n} a_i y_i \Phi(x, x_i) + b\right] \tag{3.9}$$

where a_i is a positive real constant and b is a real constant, Φ is a mapping function used for SVM kernel function construction (Suykens & Vandewalle, 1999),which typically has the choices from linear, polynomial and radial basis function(RBF) function. The solution to a nonlinear optimisation problem with inequality constraints is given by the saddle point of the Lagrangian, which is computed by:

$$\max_{\alpha_i, \upsilon_i} \min_{w,b,\xi_i} \mathfrak{L}(w, b, \xi_i; \alpha_i, \upsilon_i) \tag{3.10}$$

where $\mathfrak{L}$ is the Lagrangian constructed by:

$$\mathfrak{L}(w, b, \xi_i; a_i, \upsilon_i) = \mathfrak{J}(w, \xi_i) - \sum_{i=1}^{n} a_i\{y_i[w^T\varphi(x_i) + b] - 1 + \xi_i\} - \sum_{i=1}^{n} \upsilon_i \xi_i \tag{3.11}$$

where $a_i \geq 0$, $b_i \geq 0 (i = 1, \cdots, n)$ are Lagrange multipliers, $\mathfrak{J}$ is the risk bound minimized by:

$$\min_{w,\xi_i} \mathfrak{J}(w, \xi_i) = \frac{1}{2} w^T w + c \sum_{i=1}^{n} \xi_i \tag{3.12}$$

where the parameter ξ_i is introduced by:

$$y_i[w^T\varphi(x_i) + b] \leq 1 - \xi_i, \quad i = 1, \cdots, n \xi_i \leq 0 \tag{3.13}$$

Although SVM has been extensively used for solving real world problems in different

research areas, there are some issues that we have to consider if we would like to have a successful implementation. One main limitation of SVM methods lies in the choice of kernel for solving real world problems, which remains an open research question in computer and engineering science. Another concern of SVM implementation for real world problems is speed and size, especially during training stage. This issue might make the learning process for a very large dataset (a large number of support vectors) particularly difficult (Burges, 1998). Additionally, SVM is difficult to adapt to new data and the knowledge discovered by it is very limited (Kasabov, 2007b).

3.4 Conclusion

The chapter has presented a brief review of intelligent computational methods, including EA, ECOS and SVM. Genetic algorithms discussed here will be employed into the proposed PMS in later chapters for optimisation problems.

Evolutionary computation and ECOS seem to be the very attractive techniques that are applicable for optimising models and systems, owing to their ability to evolve the structure and function of the created models. In addition, SVM is a robust and reliable algorithm widely used in the development of computational intelligent systems for machine learning. Chapter 4 will propose a new modelling technique, namely personalised modelling that comprises different computational intelligent methods for data analysis and knowledge discovery.

CHAPTER 4

Global, Local and Personalised Modelling Approaches to Data Modelling and Knowledge Discovery

"That is what learning is. You suddenly understand something you've understood all your life, but in a new way."

- Doris Lessing

4.1 Inductive vs. Transductive Reasoning

Knowledge discovery is the process using computer technology to search large volumes of data for patterns that can be considered informative and useful. It offers a powerful tool to transform data into information and knowledge that can be used for a wide range of profiling practices, such as marketing, disease diagnosis, risk evaluation and new scientific knowledge discovery.

Most of the learning models and systems in artificial intelligence that have been developed and implemented are based on two approaches: inductive and transductive

inference. The original theory of inductive inference proposed by Solomonoff (1964a, 1964b) in early 1960s was developed to predict the new data based on observations of a series of given data. In the context of knowledge discovery, the inductive reasoning approach is concerned with the construction of a function (a model) based on the observations, e.g., predicting the next event (or data) based upon a series of historical events (or data) (Bishop, 1995; Levey et al., 1999). Many of the statistical learning methods, such as: SVM, Multi Layer Perceptron (MLP) and neural network models have been developed and tested on inductive reasoning problems.

Inductive inference approach is widely used to build models and systems for data analysis and pattern discovery in computer science and engineering. This approach creates the models based upon known historical data vectors and applicable to the entire problem space. However, the inductive learning and inference approach is only efficient when the entire problem space (global space) is searched for the solution of a new data vector. Inductive models generally neglect any information related to the particular new data sample, which raises an issue about the suitability of a global model for analysing new input data.

In contrast to inductive learning methods, transductive inference introduced by Vapnik (1998) is a method that creates a model to test a specific data vector (a testing data vector) based on the observation of a specific group of data vectors (training data). The models and methods created from transductive reasoning focus on a single point of the space (the new data vector), rather than on the entire problem space. Transductive inference systems emphasize the importance of the utilisation of the additional information related to the new data point, which brings more relevant information to suit the analysis of the new data. Within the same given problem space, transductive inference methods may create different models, each of them specific for testing every new data vector.

In a transductive inference system, for every new input vector x_v to be processed for a prognostic or classification task, the following steps are performed:

1. The N_v nearest neighbours derived from an existing dataset D will form a subset D_x. If necessary, some data in D_x can also be generated by an existing model M (e.g. the information and knowledge retrieved from an existing clinical model);

2. A new model M_x is dynamically created based on these samples to approximate the function in the locality of x_v;

3. Model M_x is then specifically used to calculate the output value y_v corresponding to the input vector x_v;

Transductive inference systems have been applied to a variety of classification problems, such as heart disease diagnostics (D. Wu, Bennett, Cristianini, & Shawe-taylor, 1999), promoter recognition in bioinformatics (Kasabov & Pang, 2004), microarray gene expression data classification (West et al., 2001). Other examples using transductive reasoning systems include: evaluating the predicting reliability in regression models (Bosnic, Kononenko, Robnik-Sikonja, & Kukar, 2003), providing additional reliability measurement for medical diagnosis (Kukar, 2002), transductive SVM for gene expression data analysis (Pang & Kasabov, 2004) and a transductive inference based radial basis function (TWRBF) method for medical decision support system and time series prediction (Song & Kasabov, 2004). Most of these experimental results have shown that transductive inference systems outperform inductive inference systems, because the former have the ability to exploit the structural information of unknown data.

Some more sophisticated transductive inference approaches have been developed including: Transductive Neural Fuzzy Inference System with Weighted Data Normalization - TWNFI (Song & Kasabov, 2006) and Transductive RBF Neural Network with Weighted Data Normalization - TWRBF (Song & Kasabov, 2004). These methods create a learning model based on the neighbourhood of new data vector, and then use the trained model to calculate the output.

Transductive inference approach seems to be more appropriate to build learning models for clinical and medical applications, where the focus is not simply on the model, but on the individual patient's condition. Complex problems may require an individual or a local model that best fits a new data vector, e.g. a patient to be clinically treated; or a future time moment for a time-series data prediction, rather than a global model that does not take into account any specific information from the object data (Song & Kasabov, 2006). However, in order to implement transductive modelling for data analysis problems, we must address some open questions, for example:

- How many variables should be used and what is their importance of them in terms of modelling construction?
- How to measure the distance between the data points when finding the neighbours in the given data set?
- What classification method to use?

These issues will be discussed in Chapter 5.

4.2 Global, Local and Personalised Modelling

Global, local and personalised modelling are currently the three main techniques for modelling and pattern discovery in the machine learning area. These three types of modelling techniques are derived from inductive and transductive inference and are the most commonly used learning techniques for building the models and systems for data analysis and patter recognition (Kasabov, 2007b, 2009). This section will investigate these three techniques for data analysis and model design.

4.2.1 Definitions

- **Global modelling** creates a model from the data that covers the entire problem space. The model is represented by a single function, e.g. a regression function, a radial basis function (RBF) , a MLP neural network, SVM, etc.
- **Local modelling** builds a set of local models from data, where each model represents a sub-space (e.g. a cluster) of the whole problem space. These models can be a set of rules or a set of local regressions, etc.
- **Personalised modelling** uses transductive reasoning to create a specific model for each single data point (e.g. a data vector, a patient record) within a localised problem space.

4.2.2 Experiment Setup

To illustrate the concepts of global, local and personalised modelling, we hereby present a comparative study in which we have applied each type of model to a benchmark gene expression dataset, namely colon cancer data (Alon et al., 1999) for cancer classification.

The main objectives of this comparative study are:

1. To illustrate the differences among global, local and personalised modelling for data analysis and knowledge discovery;
2. To present a brief review of several popular algorithms used for data modelling and knowledge discovery;.
3. To investigate several popular algorithms that are used for global, local and personalised modelling.

The data used in the comparative experiment originates from Colon cancer data proposed by Alon et al. (1999). The dataset consists of 62 samples of colon epithelial cells from colon cancer patients. 40 samples are collected from tumors and labeled as "diseased", and 22 samples are labeled "normal" and are collected from a healthy part of the colon of the same patient. Each sample is represented by $2,000$ genes selected out of total $6,500$ genes based on the confidence in measured expression levels.

Since this experiment is mainly designed for demonstrating the difference of classification performance of three modelling techniques, we simply select 15 out of 2,000 genes by a signal-noise-to-ratio (SNR) method according to their statical scores for the purpose of reducing computational cost. SNR algorithm is later described in detail in section 5.1.4. Thus, the preprocessed subset used in the experiment presented in this chapter constitutes 62 samples. Each sample contains 15 top genes based on their statistical SNR ranking scores. The subset is denoted by $D_{colon15}$.

As our interest for this experiment is mainly in the comparison of the classification performance obtained from three different modelling techniques, we have applied a simple validation approach (*Hold-out* method) to the classification on data $D_{colon15}$:

the given data is split into training and testing data with a specified ratio, i.e. 70% of samples are used for training and the remaining 30% for testing (classification problem see also Section 5.3.

The experiment is carried out in Matlab environment, and some functional modules, such as visualisation of Multiple linear regression (MLR) model and SVM model are derived from **NeuCom** and **Siftware** (refer to `http://www.theneucom.com`). **NeuCom** and **Siftware** are two generic intergraded systems for data analysis, modelling, profiling and knowledge discovery developed by the Knowledge Engineering and Discovery Research Institute - KEDRI, AUT (`http://www.kedri.info`). These two systems consolidate a variety of statistical algorithms, artificial intelligent models and evolving intelligence methods, that can be used for solving complex data analysis problems.

4.2.3 Global Modelling

Linear and logistic regression models might be the most popular global modelling techniques. They have been implemented in a variety of global methods for modelling gene expression data (T. Furey et al., 2000), and for modelling gene regulatory networks (D'haeseleer, Liang, & Somogyi, 2000).

Multiple linear regression

MLR is a global modelling technique that is among the simplest of all statistical learning algorithms. MLR analysis is a multivariate statistical technique that examines the linear correlations between a single dependent variable and two or more independent variables. For multiple linear regression analysis, the independent variable X is described by an m-dimensional vector: $X = \{x_1, x_2, \cdots, x_m\}$. Thus, we can obtain a MLR model:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_m x_{im} \varepsilon_i, \quad i = \{1, 2, \cdots, n\} \tag{4.1}$$

where:

- β is an m-dimensional parameter vector called effects or (regression coefficients);
- ε is the *"residual"* representing the deviations of the observed values y from their means $\overline{y}$, which are normally distributed with mean 0 and variance;
- n is the number of observations.

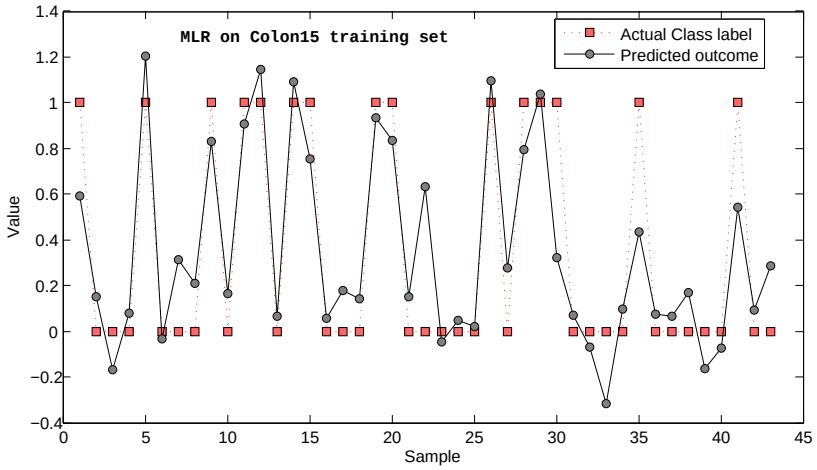
For the purpose of investigating the global modelling for classification problems, an MLR based approach is applied to the subset of colon cancer gene expression data ($D_{colon15}$). A global MLR-based classifier is created from the training data (70%) analysis, which is given as:

$$\begin{aligned}\mathcal{Y} \;=\; & 0.1997 + 0.1354 * \mathbf{X_1} + 0.70507 * X_2 + -0.42572 * X_3 - 0.19511 * X_4 \\ & +0.0943 * \mathbf{X_5} - 0.6967 * \mathbf{X_6} - 1.0139 * X_7 + 0.9246 * \mathbf{X_8} \\ & +0.1550 * \mathbf{X_9} + 0.6190 * X_{10} + 0.1793 * X_{11} + 1.123 * \mathbf{X_{12}} \\ & -0.1615 * X_{13} - 0.4789 * X_{14} - 0.4910 * X_{15}\end{aligned} \tag{4.2}$$

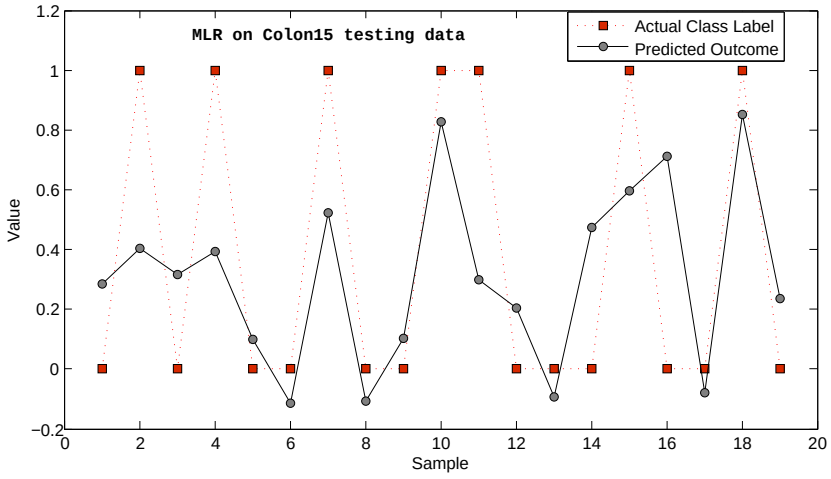
where $\mathcal{Y}$ is an MLR model to predict the new input data vector (here is to predict whether a patient sample is "diseased" or "normal"), and $X_i, i = 1, 2, \ldots, 15$ denotes each variable (feature).

Function 4.2 constitutes a global model to be used for evaluating the output for any new data vector in the 15-dimensional space regardless of where it is located. This global model extracts a 'big' picture for the whole problem space, but lacks an individual profile (Kasabov, 2007b). It indicates to certain degree the genes' importance: X_6, X_8 and X_{12} show strong correlation to the corresponding output, while X_5, X_1, X_9 are less important in terms of outcome prediction.

Figure 4.1 shows the prediction result from the global multi-linear regression model over colon data with selected 15 genes. The results plotted in Figure 4.1 (a) and (b) demonstrate the inconsistent issue in microarray gene expression data analysis: the accuracy from testing data is significantly lower than that from training data - 95.3% vs. 73.7%, when the threshold of disease distinction is set to 0.5. Such inconsistency issue will be discussed in detail in Section 5.7.



(a) The classification result using a global MLR model on $D_{colon15}$ training set (the training accuracy is 95.3%);



(b) The classification result using a global MLR model on $D_{colon15}$ testing set (the testing accuracy is 73.7%).

Figure 4.1: *An example of global modelling: the classification results from a multi-linear regression model(MLR) over colon cancer gene data, where x axis is the sample index, y axis represents the value of the actual class label and predicted outcome for each sample. The red square points represent the actual class labels of the samples, while the black circle points present the predicted outcome.*

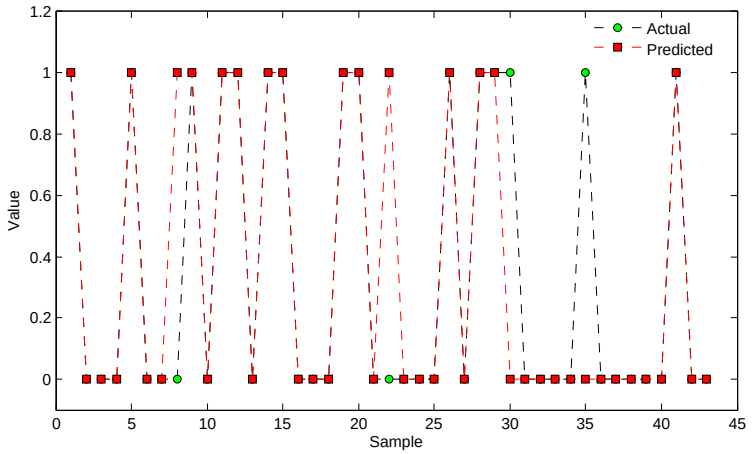
A Global SVM Modelling

For comparison, we applied a global SVM classifier on the colon data $D_{colon15}$, using the same sampling method (70% for training, 30% for testing). As described in Chapter 3, SVM is a robust algorithm that can be implemented into different modelling approaches. Here, the experiment uses a classical SVM to perform a classification on the given colon cancer data through a global modelling approach. The experiment result is illustrated in Figure 4.2. The accuracy on the training set is 90.7% (39 out of 43 samples are successfully classified), while the accuracy on testing set is still significantly lower - 79.0%.

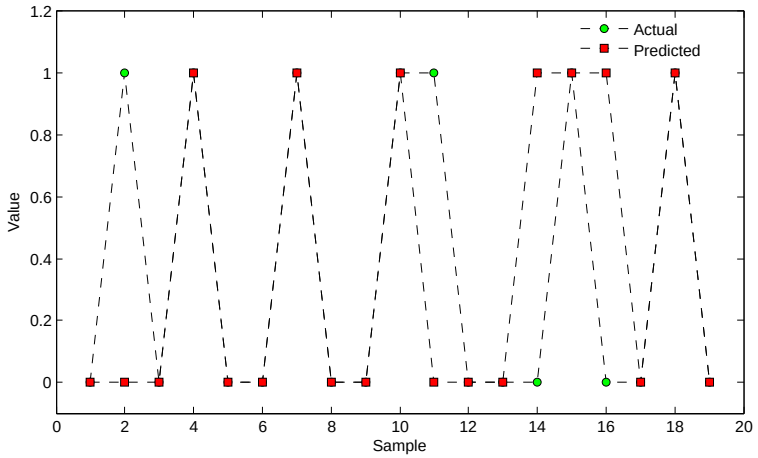
4.2.4 Local Modelling

Unlike global models, local models are created to evaluate the output function especially within a sub-space of the entire problem space (e.g. a cluster of data). Multiple local models can consist of the complete model across the entire problem space. Local models are usually based on clustering techniques. A cluster is a group of similar data samples, where similarity is measured predominantly as Euclidean distance in an orthogonal problem space. Clustering techniques can be found in the literature: classical k-means (Lloyd, 1982), Self-Organising Maps (SOM) (Kohonen, 1982; Graepel, Burger, & Obermayer, 1998), fuzzy c-means clustering (Bezdek, 1982), hierarchical clustering for cancer data analysis (Alon et al., 1999), a simulated annealing procedure based clustering algorithm for finding globally optimal solution for gene expression data (Lukashin & Fuchs, 2001). Fuzzy clustering is a popular algorithm used to implement local modelling for machine learning problems. The basic idea behind it is that one sample may belong to several clusters to a certain membership degree, and the sum of membership degree should be one.

Local learning models adapt to new data and discover local information and knowledge, that provide provide a better explanation for individual cases. However, these local modeling methods do not select specific subsets of features and precise neighbourhood of samples for individual samples that require a personalised modelling in the medical area. Evolving classification function (ECF) (Kasabov, 2002; Kasabov & Song, 2002) is a representative technique for local modelling (the detailed algorithm of ECF refers to Appendix E). The classification result from ECF local model over



(a) The classification results of SVM model over $D_{colon15}$ training set (the training accuracy is 90.7%);

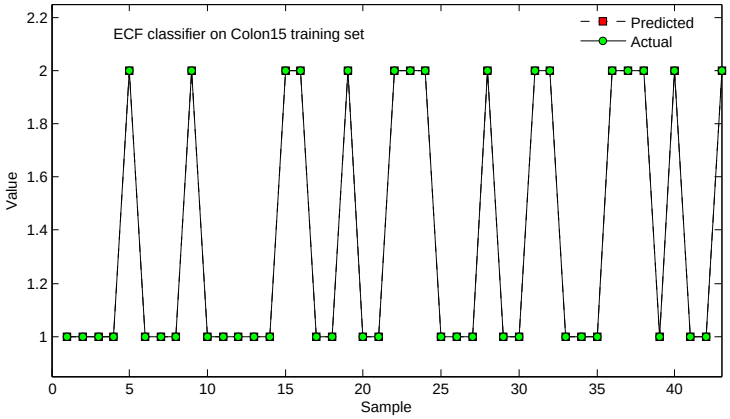


(b) The classification results of SVM model over $D_{colon15}$ testing set (the testing accuracy is 79.0%).

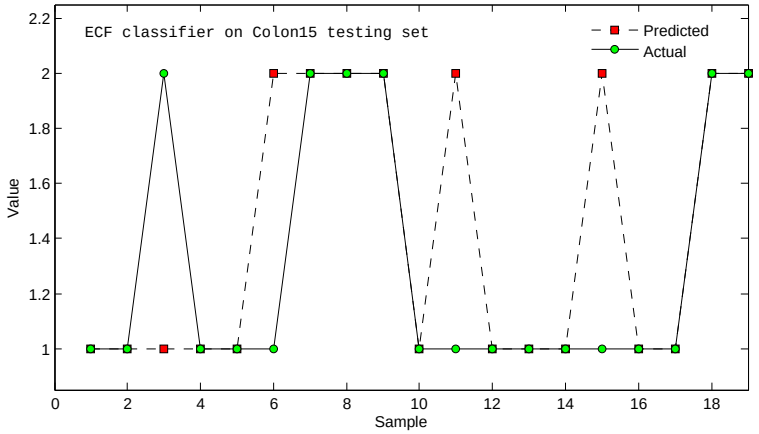
Figure 4.2: *An example of global modelling: the outcomes from a polynomial SVM model, where x axis is the sample index, y axis represents the value of the actual class label and predicted outcome for each sample. The green circle points represent the actual class label of the sample, while the red squared points are the predicted outcome.*

dataset $D_{colon15}$ is shown in Figure 4.3(a) and 4.3(b). The classification accuracy from ECF model on the training set (70% of the whole data) appeared excellent - 100% accurate, but the classification result from the testing set (30%) is only 78.95% (15 out of 19 samples are correctly classified). It seems that local modelling might

not be an effective approach for analysing this particular gene expression dataset. Moreover, it is difficult to optimise the parameters during the learning process.



(a) A local modelling: the outcomes from ECF model on the training set of colon cancer data (70%), the training accuracy is 100%.



(b) A local modelling: the outcomes from ECF model on the testing set of colon cancer data (30%), the testing accuracy is 79.0%.

Figure 4.3: *An example of local modelling: the experimental results from a local modelling method (ECF) on the training and testing set from data ($D_{colon15}$), respectively. Black solid line represents the actual label of the sample, while red dotted line is the predicted outcome.*

4.2.5 Personalised Modelling

In contrast to global and local modelling, personalised modelling creates a model for every new input data vector based on the samples that are closest to the new data

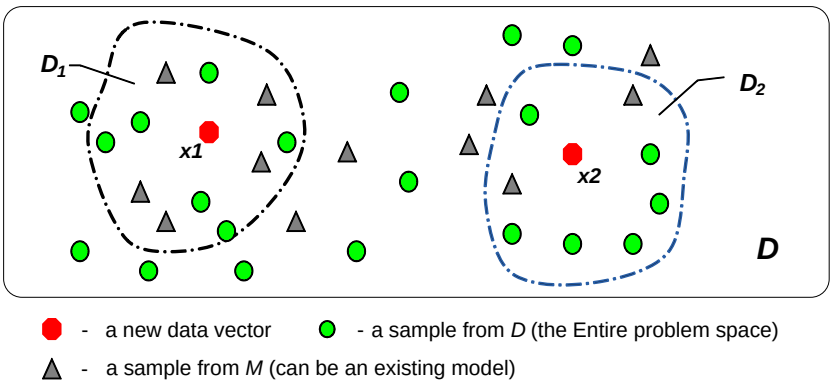


Figure 4.4: *An example of personalised space, where x_1 and x_2 represent two new input vectors, D is the entire (global) problem space, D_1 and D_2 denote the two personalised spaces for x_1 and x_2, respectively.*

vector in the given dataset. Figure 4.4 gives an example for personalised problem spaces. KNN method is probably the simplest techniques to use for personalised modelling. In a KNN model, the K nearest samples for every new sample x_i are derived from the given dataset through a distance measurement (usually Euclidean distance), and the class label for the new sample x_i is assigned based on a voting scheme (T. Mitchell, Keller, & Kedar-Cabelli, 1986). The classical KNN method calculates the output value y_i according to the determination made by the majority vote of its neighbours, i.e. the new data vector is assigned to the class most common amongst its k nearest neighbours.

KNN algorithm is one of the most popular algorithms in machine learning, because it is simple to implement and works fast and effectively on many machine learning problems. However, the parameter selection is a critical factor impacting on KNN classifier's performance, e.g., the choice of value for K. In general, more nearest neighbours (K) used in KNN method can reduce the effect of noise over the classification, but would make the boundaries between classes less distinct. If too few neighbours are selected, there can be insufficient information for decision making. Also, the performance of the KNN algorithm can be severely degraded by the presence of noisy features which is a very common issue in biomedical data.

Weighted Nearest Neighbour Algorithms for Personalised Modelling: WKNN & WWKNN

In a weighted distance KNN algorithm (WKNN) , the output y_i is calculated not only based on the output values (e.g. class label) y_j, but is also dependent on the weight w_j measured by the distance between the nearest neighbours and the new data sample x_i:

$$y_i = \frac{\sum_{j=1}^{K_i} w_j \cdot y_j}{\sum_{j=1}^{K_i} w_j} \tag{4.3}$$

where:

- y_i is the predicted output for the new vector x_i;
- y_j is the class label of each sample in the neighbourhood of x_i.
- K_i is the number of K nearest samples to x_i;
- w_j is the is the weight value calculated based on the distance from the new input vector x_j to its K nearest neighbours.

The weight w_j can be calculated as follows:

$$w_j = \frac{\max(d) - (d_j - \min(d))}{\max(d)}, \quad j = 1, \cdots, K \tag{4.4}$$

where:

- the value of weights w_j ranges from $\frac{\min(d)}{\max(d)}$ to 1;
- $d = [d_1, d_2, \cdots, d_K]$ denotes the distance vector between the new input data d_i and the its K nearest neighbouring samples;
- $\max(d)$ and $\min(d)$ are the maximum and minimum values for vector d.

The distance vector d is computed as:

$$d_j = \sqrt{\sum_{l=1}^{m}(x_{i,l} - x_{j,l})^2}, \quad j = 1, \cdots, K \tag{4.5}$$

where m is the number of variables (features) representing the new input vector x_i within the problem space; $x_{i,l}$ and $x_{j,l}$ are the l^{th} variable values corresponding to the data vector x_i and x_j, respectively.

The output from a WKNN classifier for the new input vector x_i is a "*personalised probability*" that indicates the probability of vector x_i belonging to a given class. For a two-class classification problem, a WKNN classifier requires a threshold θ to determine the class label of x_i, i.e., if the output (*personalised probability*) is less than the threshold θ, then x_i is classified into the group with "small" class label, otherwise into the group with "big" class label. For example, in a case of a two-class problem, the output from WKNN model for sample#1 of data $D_{colon15}$ is 0.1444, so that this testing sample is classified into class **1** ("small" class label) when the threshold θ is set to 0.5.

Weighted distance and weighted variables K-nearest neighbours (WWKNN) is a personalised modelling algorithm introduced by Kasabov (2007b). The main idea behind WWKNN algorithm is: the K nearest neighbour vectors are weighted based on their distance to the new data vector x_i, and also the contribution of each variable is weighted according to their importance within the local area where the new vector belongs (Kasabov, 2007b). In WWKNN, the assumption is made that the different variables have different importance to classifying samples into different classes when the variables are ranked in terms of their discriminative power of class samples over the whole *m-dimensional* space. Therefore, it will be more likely that the variables have different ranking scores if the discriminative power of the same variables is measured for a sub-space (localised space) of the entire problem space. The calculation of Euclidean distance d_j between a new vector x_i and a neighbour x_j is mathematically formulated by:

$$d_j = \sqrt{\sum_{l=1}^{K} c_{i,l}(x_{i,l} - x_{j,l})^2}, \quad j = 1, \cdots, K \tag{4.6}$$

where: $c_{i,l}$ is the coefficient weighing x_l in relation with its neighbourhood of x_i, and K is the number of the nearest neighbours. The coefficient $c_{i,l}$ can be calculated

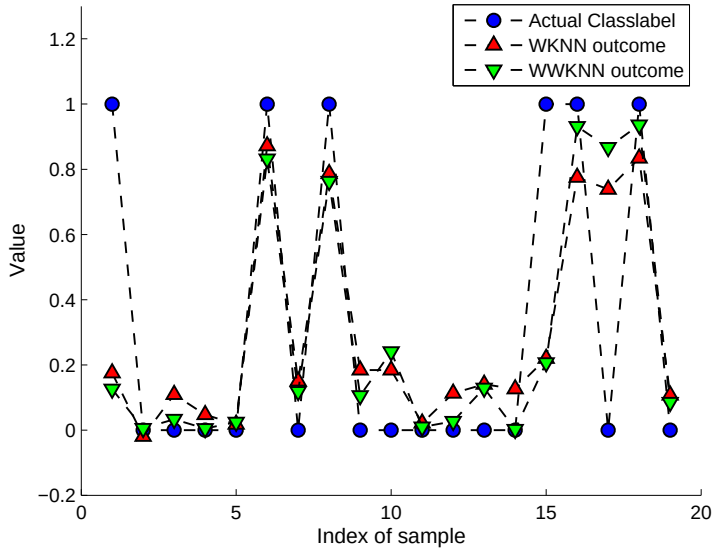


Figure 4.5: *The experimental results computed by two personalised models - WKNN and WWKNN on the colon cancer $D_{colon15}$ testing set (it contains 19 samples). $K = 15$ and the classification threshold is 0.5. The classification accuracies from WKNN and WWKNN are 84.2% and 78.9%, respectively.*

by a SNR function that ranks variables across all vectors in the neighbourhood set $D_{nbr}(x_i)$:

$$\begin{aligned} c_{i,l} &= \{c_{i,1}, c_{i,2}, \cdots, c_{i,K}\} \\ c_{i,l} &= \frac{|\bar{x_l}^{class1} - \bar{x_l}^{class2}|}{\sigma_l^{class1} + \sigma_l^{class2}} \end{aligned} \tag{4.7}$$

where:

- $\bar{x_l}^{classi}$, $i = \{1, 2\}$: the mean value of the l^{th} feature belonging to class i across the neighbourhood $D_{nbr}(x_i)$ of x_j;
- σ_l^{classi}, $i = \{1, 2\}$: the standard deviation of l^{th} feature belonging to class i across the neighbourhood $D_{nbr}(x_i)$ of x_j.

Comparing to a conventional KNN algorithm, the contribution of WWKNN lies in the new distance measurement: all variables are weighted according to their importance as discriminating factors in the neighbourhood area (personalised sub-space), which might provide more precise information for classification or prediction of the new data vector.

The experimental results from the classification of $D_{colon15}$ data using WKNN and WWKNN are summarised in Figure 4.5. It shows that WWKNN outperforms WKNN (84.2% vs. 78.9%) for colon cancer data classification. Both WKNN and

WWKNN can create an outcome vector indicating the testing sample's probability of being diseased, which provides the important information for clinical decision making.

4.3 A Case Study of Comparing Global, Local and Personalised Modelling Approaches

The previous section 4.2 provides a detailed description and comparative analysis of the three modelling approaches. This section presents a case study where an incorporated personalised modelling approach is used for cancer diagnosis. The case study mainly aims to investigate the classification performance obtained from different algorithms using global, local and personalised modelling techniques over a benchmark gene expression datasets - the diffuse large B-cell lymphoma (DLBCL) datasets (Shipp et al., 2002).

4.3.1 Experiment Setup

The objective of this experiment is to compare the global, local and personalised models for lymphoma classification. Five classification models - MLR, KNN, SVM, ECF, and WWKNN are applied to the cancer data analysis experiment.

Data

The diffuse large B-cell lymphoma (DLBCL) dataset contains genetic data of patients with one of the two types of lymphoma - diffuse large B-cell lymphoma (DLBCL) and Follicular lymphoma (FL). The dataset has 58 DLBCL samples and 19 FL samples, and each sample contains 6,817 genes.

4.3.2 Results and Discussion

Each of the models used in this experiment was validated through a leave-one-out cross validation (LOOCV) . Originally, to remove the noise and irrelevant genes,

Shipp and her colleagues applied a SNR-based gene selection method on the whole dataset and selected the top 30 genes (Shipp et al., 2002). We also used a SNR-based method to select the top 30 genes based on their SNR ranking scores (i.e. we used the same 30 genes as those selected in Shipp's work), and applied different classifier models on the lymphoma data.

The overall classification accuracy (in %) obtained by applying five models (global, local and personalised) is presented in Table 4.1. In the last two columns, k is the number of neighbours used in the WWKNN algorithm.

Table 4.1: *The classification results obtained from 5 models on Shipp's DLBCL data using 30 genes*

Model	MLR	KNN	SVM	ECF	WWKNN (k=5)	WWKNN (k=15)
Number of selected genes	30	30	30	30	30	30
Overall accuracy	85.71%	84.42%	84.42%	88.31%	84.42%	**89.61%**

The 30 genes selected in our experiment can be found in the list of 50 biomarker genes finally reported by Shipp et al. (2002) for distinguishing two types of lymphoma: DLBCL and FL. However, comparing to Shipp's biomarker gene list, the importance of these 30 genes from our SNR method is different. For example, the top 2 genes we selected - gene HG1980-HT2023 and M14328 are described as the two marker genes ranked at the 8th and 2nd position based on their biological importance to DLBCL (Shipp et al., 2002).

The best classification accuracy (89.61%) achieved on Shipp's data is from the personalised WWKNN model - 69 out 77 samples are successfully classified. The local model ECF performs better than other models (MLR, KNN and SVM). In the experiment, it is found that the WWKNN performance is sensitive to the selection of some parameters, e.g. the number of the nearest neighbours (k).

In the experiment, 12 genes always appear among the top 16 selected by the personalised modelling method WWKNN, across the whole sample population. In other words, for every individual lymphoma sample, these 12 genes have a very high probability to be selected as the informative genes for distinguishing lymphoma types. Also, the same 12 genes are found important and ranked among the top 20 in Shipp's experiment. Table 4.2 summarised these 12 genes with their biological information.

Table 4.2: *12 selected genes from Shipp's DLBCL data*

Gene Index	Biological description
$HG1980-HT2023_{at}$	Tubulin, Beta 2
$M14328_{s_{at}}$	ENO1 Enolase 1,(alpha)
$X56494_{at}$	PKM2 Pyruvate kinase, muscle
$X02152_{at}$	LDHA Lactate dehydrogenase A
$M57710_{at}$	LGALS3 Lectin, galactoside-binding, soluble, 3 (galectin 3)
$L17131_{rna1_{at}}$	High mobility group protein (HMG-I(Y)) gene exons 1-8
$J03909_{at}$	GAMMA-INTERFERON-INDUCIBLE PROTEIN IP-30 PRECURSOR
$HG417-HT417_{s_{at}}$	Cathepsin B
$HG2279-HT2375_{at}$	Triosephosphate Isomerase
$M63138_{at}$	CTSD Cathepsin D (lysosomal aspartyl protease)
$D82348_{at}$	5-aminoimidazole-4-carboxamide-1-beta-D-ribonucleoti de
$M22382_{at}$	HSPD1 Heat shock 60 kD protein 1 (chaperonin)

The DLBCL dataset has been studied extensively, and consequently many models and approaches have been developed. Most of the studies are focused on the performance in terms of computational results. However, it is generally agreed that currently no model or approach can always perform well on different gene expression data for cancer diagnosis and prognosis. In many cases, the performance of prediction can not be the only factor to judge whether a model is superb than others. Other factors, such as the consistency of prediction performance, and reproducibility of the experimental results should be taken into account.

In this case study, we are more interested in what knowledge can be discovered by these three different modelling techniques and which one is more appropriate for cancer gene expression data analysis. For example, one of our findings is that the 12 selected genes are among the most important genes reported by other published paper, which means these genes should be further studied to evaluate whether they are contributive to other cancer diagnosis and prognosis.

In Shipp's work, the best accuracy they achieved is 92.2% using a weighted voting algorithm with 30 selected genes based on cross-validation testing, which is slightly better than the result from our WWKNN model (89.61%). However, regarding their data sampling and validation approaches, there exist some open questions, e.g. how many genes are best fit for the classification over DLBCL data, because their method does not involve parameter optimisation.

4.4 Conclusion and Open Problems

In order to describe the notion of personalised modelling, this chapter has presented a brief review of inductive and transductive reasoning method for data analysis in machine learning. It also discusses a preliminary study through a comparison of three major modelling approaches, namely global, local and personalised modelling for microarray data analysis.

Global models reveal the trend in data that is valid for the whole problem space, while local models capture local patterns from the clusters of data. Both global and local models can discover some useful information and knowledge from the analysis of available data. Local models are also adaptive to new input data through forming new clusters and applying new functions to capture data patterns (Kasabov, 2007b). In short, these two modelling approaches assume a fixed set of variables, which makes it difficult to modify and accommodate the new variables along with new input data.

Personalised modelling approach can be a solution to the issues raised by global and local modelling, since it spontaneously creates the models that accommodate any new variables to fit the new data. The experiment results also show that the strength of personalised modelling is not only providing a competitive way for data analysis.

This chapter has discussed the issues of personalised modelling for data analysis. The personalised modelling construction is a complex process that requires evolving and adaptive computational techniques. The chapter raises the questions and open problems that need to be discussed and solved in the rest of this thesis:

1. How to determine the appropriate personalised problem space for a new input data sample? For example, how many samples (K) should be included in the neighbourhood (personalised problem space), and which samples are best to represent the pattern of object sample?

2. How to find the best combination of parameters for the learning functions (e.g. a classifier)?

3. How many and which features are highly differentially expressed between different samples and are of benefit to assessing the outcome for the new input

data sample?

4. How to build the profile from the analysis on different data sources, such as gene expression data, protein data, clinical data, SNPs data, etc?

5. How to effectively visualise the outcomes and results to help understand the information discovered from data analysis?

In fact, the above issues and open questions motivate us to find better solutions to personalised modelling for genomic data analysis. Chapter 5 gives a detailed discussion of these issues and questions.

CHAPTER 5

Critical Analysis of Problems Related to Personalised Modelling

"Knowledge is power. Rather, knowledge is happiness, because to have knowledge – broad, deep knowledge – is to know true ends from false, and lofty things from low."

- Helen Adams Keller

Despite the increasing interest in the concept of personalised modelling, especially for biomedical applications, the methods and systems are still far away from their mature stage. There are issues related to personalised modelling that are of significant concern to researchers. These issues can be related to the types of data, biological relevance of features, data classification problems, parameters tuning, overfitting, etc. I believe that in order to develop efficient personalised modelling framework and systems for data analysis and modelling, it is necessary to study and acquire an in-depth understanding of the problems and the related issues.

5.1 Feature Selection - a Critical Step in Personalised Modelling

5.1.1 Introduction

In this thesis, a main application of the proposed personalised modelling is for the development of disease prediction system that uses microarray gene expression data. Owing to the ability to observe thousands of gene expression levels simultaneously, microarray data technology is a scientific breakthrough in the realm of complex disease research, and provides a powerful way to study life science at genomic level. Over the last two decades, microarray gene expression data has been extensively studied in medical research, especially for the diagnosis and prognosis of complex diseases, such as cancer. Many research studies have claimed excellent results achieved using microarray data, especially for cancer diagnosis and prognosis (Alizadeh et al., 2000; Asyali, Colak, Demirkaya, & Inan, 2006; Cho & Won, 2003). However, some concerns about the reliability of microarray experiments have been raised recently, because many published impressive experimental results are found difficult to replicate in other laboratories.

Empirical research has revealed that the issue is mainly caused by the extremely imbalanced structure of microarray datasets (Chuang et al., 2004; Pawitan et al., 2005; Li & Yang, 2002). In a typical microarray dataset, each row represents a tissue sample, and each column represents a gene's expression level. The number of samples to be analysed is very small comparing to the number of the genes on the chip. In most real microarray datasets, the number of genes (usually thousands or tens of thousands) far exceeds the number of samples (usually tens or several hundreds). For example, there are 78 samples vs. 24,482 genes in the breast cancer dataset proposed by van't Veer (2002). Figure 5.1 shows an example of a typical microarray gene expression dataset.

In machine learning research, in order to get a satisfactory classification accuracy, the sample size of a dataset should be sufficiently large comparing to the number of features (Ambroise & McLachlan, 2002; Glymour, Madigan, Preigbon, & Smyth, 1996; Hosking, Pednault, & Sudan, 1997; Varma & Simon, 2006). A good classifica-

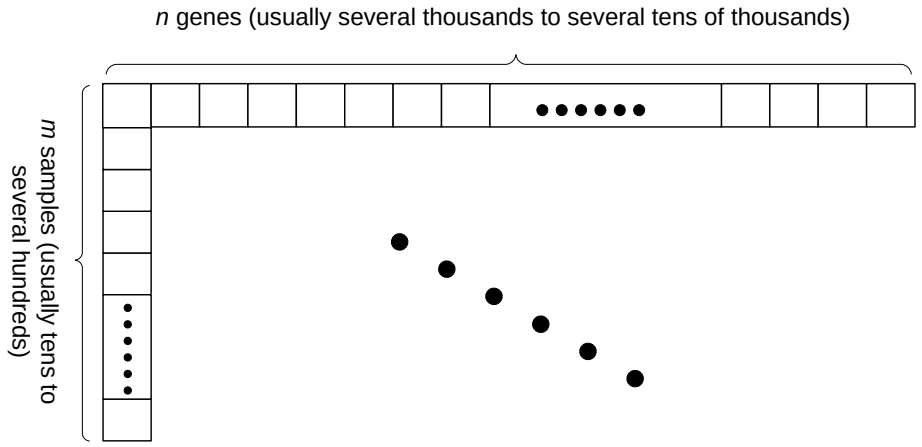


Figure 5.1: *An example of the typical imbalanced structure of a microarray gene expression dataset (m-by-n, $m \ll n$)*

tion model usually comes from a dataset with a balanced structure, i.e. the sample size should be appropriate to the number of features (Raudys, 1976). Generally, the generalisation error in machine learning area decreases when the sample size increases (Hamamoto, Uchimura, & Tomita, 1996).

However, it is difficult to get a microarray dataset with reasonably large sample size, compared to the number of features (genes). At present, microarray data is still expensive to collect and manufacture, due to the issues of intellectual property protection and the huge quantity of data points recorded into a microarray chip (a high throughput dataset usually includes more than one million data points). In most microarray datasets, only a very small proportion of features (genes) contribute to computational models (e.g. a classifier), while the rest of genes are noise genes that confuse learning models. The amount of relevant genes is typically small, as "the majority of the active cellular mRNA is not affected by the biological differences" (Wolf, Shashua, & Mukherjee, 2004, p1).

Previous disease classification work on microarray datasets has demonstrated that using a small number of informative genes can successfully discriminate the patterns of tissue samples, e.g. diseased or normal (Dudoit, Fridlyand, & Speed, 2000; Golub et al., 1999; Hu, 2008). Feature selection is thus proposed to eliminate the influence of noise genes, and to find the informative genes related to a disease.

5.1.2 Feature Selection

Identifying the features that are informative for the classification is a critical issue for understanding the biology behind the classification and for achieving promising and reliable experimental results. Feature selection is concerned with discovering a small number of most informative features that can represent the objective patterns. Gene selection is the application of feature selection in microarray gene expression data analysis research. There are plenty of reasons to employ feature selection in contemporary bioinformatics, especially for cancer classification. The main benefits of using feature selection are summarised as follows:

- Enhance the reproducibility of gene expression data analysis experiment. Gene selection will extract a compact subset of genes so that most noise genes will be eliminated. Hence, the computational models can work more properly on gene expression data, and will be more likely to produce better experiment results;
- Ease the computational burden for gene expression data analysis. It is much cheaper to focus on a small number of informative genes that can differentiate express the patterns of disease from the whole gene set.
- Improve data understanding and model interpretability. Gene selection can assist the system to reveal and visualise data more precisely in a less dimensional space.

The problem of a typical feature selection method in bioinformatics can be briefly described as follows: given a microarray gene expression dataset $D = \{X, Y\}$, where $X = \{x_i, \mid i = 1, \cdots, n\}$, $Y = \{y_i, \mid i = 1, \cdots, n\}$. Each sample is characterised by a vector of expression levels of m genes $x_i = \{x_{i1}, x_{i2}, \cdots, x_{im}\}$, and has a label $y_i = \{0, 1\}$ indicating which class it belongs to, e.g. "normal" vs. "diseased" (*Note*: here we use a two-class classification just for simplicity and convenience in terms of description). Each gene is a vector of their expression values across the samples and is denoted by $G = \{g_j, \mid i = 1, \cdots, m\}$. The goal is to find a subset of genes $S = \{s_i \mid i = 1, \cdots, l\}$ that leads to the best and reliable analysis performance. Let S^* be the optimal subset with l genes ($S^* \in G$). A learning function $\mathfrak{F}$ (a classifier or other computation models) evaluates the selected genes (candidate genes) and

computes a generalisation error p_e. The smaller the p_e, the more informative the selected gene set S^*:

$$\exists S^* \in X : \min(\mathbf{p_e}) = (\mathfrak{F}, S, X) \tag{5.1}$$

The most straightforward method of gene selection is the exhaustive search in the whole problem space:

1. Examine all the possible combination of genes;
2. Select a subset of genes (S^*) when the smallest p_e is achieved.

However, the exhaustive search in Step 1 becomes impracticable when the number of features becomes very large.

Selecting informative genes, as a critical step for cancer classification, has been implemented using a diversity of techniques and algorithms. Simple gene selection methods come from statistical models, such as t-statistics, Fisher's linear discriminate criterion and PCA (Ding & Peng, 2003; T. Furey et al., 2000; Jaeger, Sengupta, & Ruzzo, 2003; Tusher, Tibshirani, & Ghu, 2001). Statistical methods select genes by evaluating and ranking their contribution or redundancy to classification (C. Zhang, Lu, & Zhang, 2006), and are able to filter out informative genes very quickly. This type of methods usually run quickly and may achieve acceptable classification performance in some cases.

More sophisticated algorithms are also available, such as noise sampling method (Draghici et al., 2003), Bayesian model based approach (Efron, Tibshirani, Storey, & Tusher, 2001; Lee, Sha, Dougherty, Vannucci, & Mallick, 2003), significance analysis of microarrays (SAM) (Tibshirani, 2006), artificial neural networks based approach (N.Kasabov, Middlemiss, & Lane, 2003), and rough sets based approach (L. Sun, Miao, & Zhang, 2008). All these methods define a loss function, e.g. a classifier or cluster, to evaluate the goodness of candidate gene sets. Most of them claim to be capable of extracting out a set of highly relevant genes (Wolf et al., 2004), however their computational cost is much higher than that of statistical methods.

5.1.3 Main Approaches for Feature Selection: Filter, Wrapper and Embedded methods

Feature selection methods in bioinformatics literature basically fall into three categories - filter, wrapper and embedded methods, depending on whether the learning algorithm is used as a part of the selection criteria (Guyon & Elisseeff, 2006). The three types of feature selection methods are illustrated in Figure 5.2. The three types

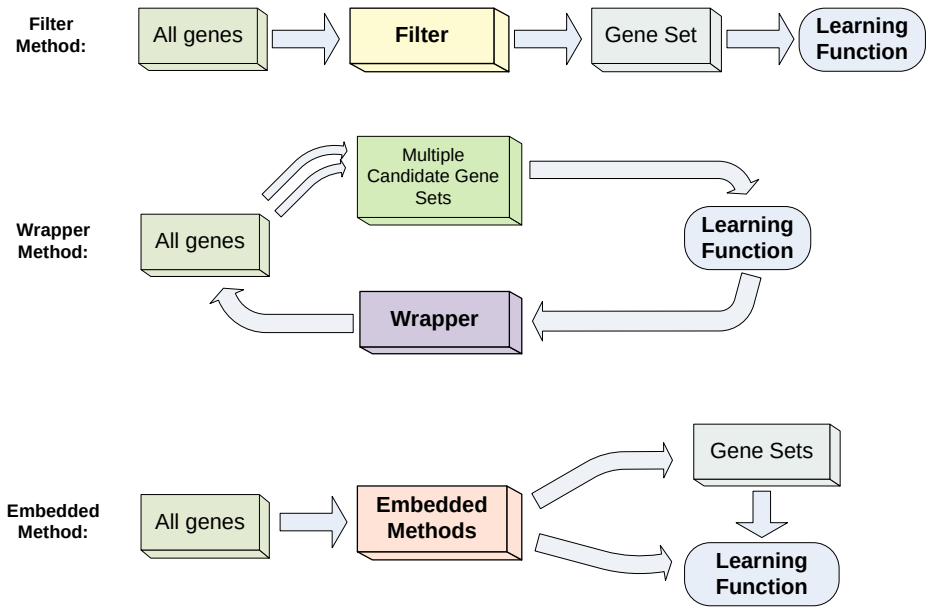


Figure 5.2: *The illustration of three feature selection approaches: filter, wrapper and embedded methods.*

of feature selection methods are discussed in the next three sections.

5.1.4 Filter Methods

Filter methods follow the methodologies of statistical models, such as t-test and SNR, as the measuring criterion to examine the intrinsic characteristics of genes. In filter methods, the feature selection procedure is independent to the prediction process, i.e. filter methods select and evaluate genes only based on their statistical scores.

A typical filter feature selection method ranks all genes according to their individual relevance. For example, the Pearson correlation coefficient can be used as the statistical relevance scores for ranking genes individually. Let $x_{i,j}$ represent the i^{th} sample (vector) with the values of the j^{th} gene from a training dataset D (n-by-m), and y is the n dimensional vector with the values of target objectives (e.g. the desired class labels in classification problems). The Pearson correlation coefficient r_{xy} for gene ranking is thus defined as:

$$r_{xy} = \frac{\left|\sum_{i=1}^{n}(x_{i,j} - \bar{x_j}) \cdot (y_i - \bar{y})\right|}{\sqrt{\sum_{i=1}^{n}(x_{i,j} - \bar{x_j})^2 \cdot \sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{5.2}$$

where $\bar{x_j}$ and $\bar{y}$ represent the mean of vectors x_j and y, respectively.

T-test is another popular choice to implement filter feature selection methods. T-test based feature selection methods evaluate to what extent each gene in a sample is in relation with a particular gene in other samples. The relationship is evaluated by a t-test algorithm and each gene is assigned a t-test statistic score calculated by:

$$T_i = \frac{\bar{x_i} - \bar{y_i}}{\sqrt{\frac{1}{n_x} + \frac{1}{n_y}} \cdot \sigma_i} \tag{5.3}$$

where T_i is the t-test statistic value of the i^{th} gene in D, $\bar{x_i}$, $\bar{y_i}$ denote the mean value of the i^{th} gene corresponding to each classes(e.g. class 0 and class 1) respectively, n_a and n_b are the number of samples of two classes, and σ_i is the pooled standard deviation for the i^{th} gene:

$$\sigma_i = \sqrt{\frac{(n_a - 1) \cdot \sigma_a^2 + (n_b - 1) \cdot \sigma_b^2}{df}} \tag{5.4}$$

where σ_a^2 and σ_b^2 is the variance of two subsets, each corresponding to one of two different classes, and df is the degree of freedom of the t-distribution under null hypothesis, which is calculated by:

$$df = n_a + n_b - 2 \tag{5.5}$$

Hence, a small number of genes with high ranking scores (t-test statistic scores) are

considered highly informative to classification problems.

One thing to bear in mind when using t-test method for feature selection is that it works well only when the data is normally distributed and the population variances are equal for the two classes. If variances are unequal in a two-class problems, the degrees of freedom (df) can be computed by a different version of T-test algorithm - Welch's T-test (Welch, 1938). The value of degrees of freedom obtained by Welch's T-test is usually smaller than that calculated by Eq.5.5.

One notable application of t-test algorithm for feature selection was presented by Dudoit, Yang, Callow, and Speed (2002). The method was based on a two-sample t-test which made the assumption that the samples in the given dataset were randomly selected from normally distributed population with equal variances. Firstly, the differentially expressed genes were evaluated by the T-statistic value (Eq. 5.3). The method also takes into account the absolute expression level of a gene (ω_i):

$$\bar{\omega_i} = \frac{\sum_{j=1}^{n} \log_2 \sqrt{R \cdot G}}{n} \tag{5.6}$$

where R and G are the intensity measurements for each gene spotted in a single-slide cDNA microarray chip, n denotes the number of hybridisations performed. Other different versions of t-test can be found in literature, such as Levene's test (Levene, 1960) and Bartlett's test (Snedecor & Cochran, 1989). Both of them are two sensitive methods when the samples have equal variances (homogeneity of variances).

T-test based feature selection methods are often found in preliminary studies as a benchmark to compare with newly developed methods, as t-test is an extensively studied algorithm and easy to implement. One of its major advantages is the simplicity and robustness, which leads to a fast computation process for feature selection.

T-test based feature selection algorithms usually make the assumption that two samples have equal variances and the genes are independent. These assumptions can have a significant negative impact on real microarray datasets, because the interaction among genes are neglected. Empirical studies have indicated that the genes selected by simple T-test based algorithms are not reliable in terms of expressing disease patterns, and are more likely to be generated by chance. For example, even if the P-value (a probability associated with a test statistic) is significantly small

(0.01) in a microarray experiment with 10,000 genes, 100 genes might be identified by chance.

Another widely used statistical algorithm, SNR is often adopted to conduct a search for discovering informative genes. This approach starts with the evaluation of a single gene and iteratively searches the candidate genes in the rest of dataset based on a statistical criterion. SNR, as a simple algorithm, is usually found generally effective to identify the difference between two normal distributed samples (Lai, Reinders, & Wessels, 2004; Veer et al., 2002). Let $\bar{x}_i$ and $\bar{y}_i$ denote the mean values of the i^{th} gene in the samples in class 1 and class 2 respectively, σ_{xi} and σ_{yi} are the corresponding standard deviations. The SNR score of each gene can be calculated by:

$$SNR(i) = \frac{|\bar{x}_i - \bar{y}_i|}{\sigma_{xi} + \sigma_{yi}}, i = 1, 2, \cdots, m \tag{5.7}$$

where m is the number of genes in the given dataset. The greater the SNR value, the more informative the gene.

SNR based algorithms for feature selection have been widely used. Examples include a univariate ranking method (Lai et al., 2004), and a weighted-voting (WV) algorithm combined with SNR method (Iwao-Koizumi, Matoba, Ueno, Kim, & al., 2005) for selecting genes in a study of human breast cancer. SNR-based feature selection usually ranks the correlated genes in the dataset according to their discriminative levels towards the classes. The genes with high SNR scores are selected as the informative variables of each class.

Filter methods can be a good choice for selecting genes when the number of genes is very large. They are usually fast and effective. Filter feature selection methods can be found in many published works: A Noise sampling method based on an ANOVA approach (Draghici et al., 2003), minimum redundancy - maximum relevance (MRMR) gene selection method (Ding & Peng, 2003), Self Organizing Maps (SOM) based method (Tamayo et al., 1999), a Singular Value Decomposition (SVD) based method (Alter, Brown, & Botstein, 2000), a.k.a gene shaving method (Hastie et al., 2000), max-surprise method (Ben-Dor, Friedman, & Yakhini, 2001), etc.

The main limitation of filter selection methods is that they ignore the possible interactions among genes. Most techniques used in filter methods are univariate. The genes are considered separately so that the interactions among genes are not taken

into account. The combination of selected genes may not follow the performance of the genes evaluated individually. Another issue is the number of the selected genes is subjectively determined by trial-and-error, since the gene ranking is based on a univariate scoring metric and the genes are selected independently from the learning function. Such schema may worsen classification performance compared to other feature selection methods.

5.1.5 Wrapper Methods

To avoid the weakness of filter methods, wrapper methods define a loss function, such as a classification model, to recursively evaluate the goodness of candidate gene subsets. The final learning function for data analysis consolidates a compact set of selected features and an optimal classifier. Figure 5.3 illustrates a simple flowchart of a wrapper feature selection method.

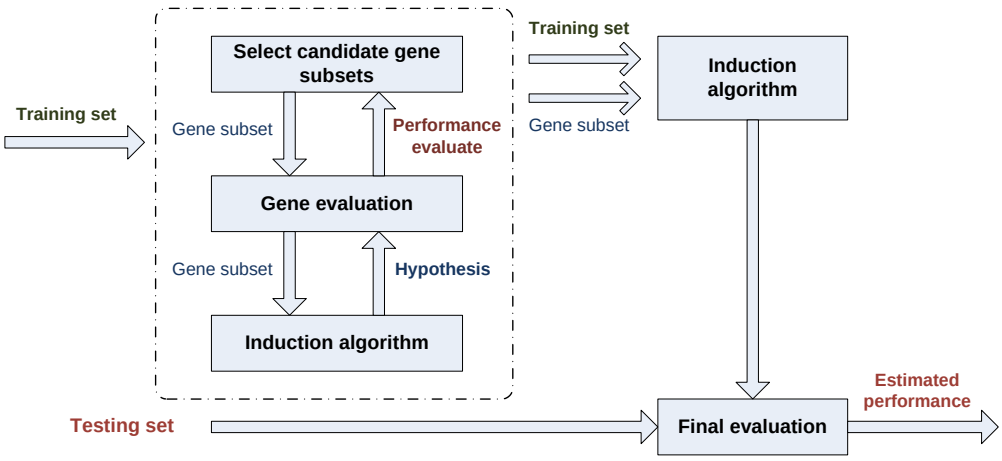


Figure 5.3: *A flowchart of a wrapper feature selection method, adapted from Kohavi and John (1997)*

The procedure of a typical wrapper feature selection method is roughly summarized as follows: for a given training dataset $D = \{x_i, y_i \mid x_i \in X, y_i \in Y, i = 1, \cdots, n\}$, the objective of feature selection (as generally understood) is to find a subset of genes that are able to assist a computational model to minimise the generalisation error. In other words, an optimal computational model using a small number of selected

genes should give a prediction as accurate as possible, and maximise the correctness of the mapping of an input set X to the output set Y. Thus, the generalisation error p_e in Eq. 5.1 is replaced with a wrap feature selection with the expecting risk $\mathbb{R}$:

$$\mathbb{R}(f_\sigma) = \int L(y, f_\sigma(\sigma \cdot x))dP(x, y) \tag{5.8}$$

where:

1. L is a loos function;
2. f_σ is a family of functions that can be a set of classifiers or regression models;
3. σ_i is an indicator vector indicating whether the gene i $(i = 1, 2, \cdots, m)$ is selected $(\sigma_i = 1)$ or not $(\sigma_i = 0)$;
4. P is a measurement function over training data $D(X, Y)$.

A constraint function s is introduced to evaluate the sparsity of σ. Therefore, a feature selection problem with a wrapper method can be rewritten as:

$$\min \mathbb{R}(f^*, \sigma, X, Y) \leftarrow \begin{cases} s(\sigma) \leq \sigma_0 \\ f^* = f_l(f_\sigma, \sigma, X, Y) \end{cases} \tag{5.9}$$

where:

1. $\mathbb{R}$ is the risk value measured by a learning function f^*, e.g. a classification or regression function. The smaller the $\mathbb{R}$ value, the better the performance;
2. σ_0 is a pre-specified parameter denoting the desired sparsity of σ;
3. f^* is the optimal function learned from the training over data $D(X, Y)$.

From Eq.5.9, it is easy to elucidate that a wrapper feature selection method is actually used to seek an appropriate criterion to drive the optimisation task of feature selection.

If $\mathbb{R}$ is allowed to be dependent on the learning model f_l and on the parameters of f^*, Eq.5.9 can further be reformulated by:

$$\min \mathbb{R}(\alpha^*, f_l, \sigma, X, Y) \leftarrow \begin{cases} s(\sigma) \leq \sigma_0 \\ \alpha^* = f_l(\sigma, X, Y) \end{cases} \tag{5.10}$$

where α^* is a function for evaluating the learning model f_l directly and can be defined as:

$$\alpha^* = \operatorname{argmin} f_l(\alpha, \sigma, X, Y) \tag{5.11}$$

In the past years, wrapper methods have become a popular choice for feature selection. Some of the works include: a GA/SBM method (Huerta, Duval, & Hao, 2006), a sequential search wrapper approache for feature selection in microarray cancer class prediction (Inza, Sierra, Blanco, & Larranaga, 2002), the FR-Wrapper approach for discovering biomarker genes for cancer classification (Peng, Li, & Liu, 2006), etc. One representative work of wrapper method for feature selection is SVM-RFE (Guyon, Weston, Barnhill, & Vapnik, 2002). This method uses a linear SVM to classify samples and ranks the contribution of the features in the classifier by their squared weights.

5.1.6 Embedded Methods

In contrast to filter and wrapper methods, embedded methods process feature selection inside the training procedure and are specific to a particular induction algorithm. The features that are finally selected by embedded methods can be seen as a by-product of the classifier training. One recently developed embedded method for feature selection can be found in my previous work - a bootstrapping consistency method for feature selection (Hu, 2008; Pang, Havukala, Hu, & Kasabov, 2007). Using this method, the candidate gene subsets are selected and evaluated by a GA based learning model based on their consistent performance through generations (usually several thousands). In each generation, the consistency is measured via a comparison between two subsets from resampled training datasets. The informative genes are finally selected when a criterion is satisfied (a balanced ratio of a consistency value to classification accuracy is achieved).

5.1.7 Discussion

Personalised modelling, especially for gene expression data analysis and biomedical applications requires efficient feature selection. The feature selection is a fundamental step towards the construction of personalised modelling, because a compact set of informative features will significantly benefit the testing performance.

Filter feature selection methods are simple and fast, but the selected features are usually only based on their statistical importance and are not evaluated by the learning model. Consequently, the selected features cannot be informative for an individual data sample and may lead to unsatisfactory classification performance. Wrapper and embedded feature selection methods are favoured in many works, since generally they can yield better classification accuracy than filter methods. A recursive searching schema for wrapper and embedded methods is usually involved to identify the optimal gene subsets. However, the good performance from wrapper and embedded methods always comes with expensive computational cost when the dataset has a high dimensionality (Kohavi & John, 1997; Guyon & Elisseeff, 2006; Saeys, Inza, & Larranaga, 2007).

In this study, to balance the computational complexity and classification performance, I have applied a combined method to select features for building personalised models. The method has two main steps: (1). use filter method to exclude the features that are significantly statistically irrelevant; (2). use wrapper method to find the informative feature from the rest. The selection process is optimised by a learning function (e.g. a classifier). The details of the implementation of a combined feature selection method will be described in Chapter 6.

5.2 Imbalanced Data Class Distribution Problem

The imbalanced class distribution problem is a critical concern for the data mining community, since it is encountered in many domains, such as in the analysis of clinical, environmental and financial data. The imbalanced class problem corresponds to the objective domains in which one class (*the majority class*) is represented by a significant large portion of samples, while the other (*the minority class*)

is represented by a very small portion of samples. For example, the SCOPE data (http://www.scopestudy.net/) used for predicting diseases in early pregnancy only contains around $5 \sim 10\%$ samples from the diseased group (class 2), while leaving most samples from normal (healthy) group (class 1). The imbalanced class issue poses a bottleneck regarding the prediction performance attainable by traditional learning algorithms that assume the objective dataset having a balanced sample class distribution.

5.2.1 Imbalanced Class Distribution Issue in Personalised Modelling

Previous studies have shown that the imbalanced class distribution issue often causes poor performance from standard classification models in many applications (Japkowicz & Stephen, 2002; Japkowicz, 2000). These standard classification models usually create classifiers that maximise the overall classification accuracy. When dealing with an imbalanced class distribution dataset, standard classification models usually lead to the training completely ignoring the minority class samples, i.e. the training is performed on all samples from the majority class. In this case, the classification over majority class samples can be very successful, while it may fail over the minority class samples. To construct a system for personalised modelling, it is crucial to find an appropriate neighbourhood of a new data vector to train candidate personalised models. However, it is often found that most or all the samples in the neighbourhood (personalised problem space) are from the majority class, especially for building the model for a sample belonging to majority class. Hence, finding a personalised problem space with reasonably balanced class distribution is of crucial importance for constructing personalised models in our study.

5.2.2 Previous Attempts at Dealing with the Imbalanced Class Distribution Problem

There have been some attempts at dealing with the imbalanced class distribution problem. Robert, Holte, Acker, and Porter (1989) reported various approaches to the problem with small disjuncts and proposed an approach based on a bias difference

evaluation. Y. Sun (2006) developed a cost-sensitive boosting algorithm for a multi-class classification problem over imbalanced data. Japkowicz (2000) addressed the imbalanced class issue for classification tasks and presented different solutions in her work.

Generally, there are three types of methods that are mainly employed for tackling the imbalanced class distribution problem:

1. Methods that use over-sampling of the minority class samples to match the size of majority class samples. One method can be found in the work presented by Ling, , Ling, and Li (1998).

2. Methods that use down-sizing the majority class samples to match the size of minority class samples. Kubat and Matwin (1997) applied a simple technique called *one-sided* selection of examples for the classification over imbalanced data.

3. Methods that use a recognition-based learning scheme. Such methods may ignore one of the two classes and the learning is often from the minority class. This scheme has been applied for different classification tasks over imbalanced datasets (Japkowicz, Myers, & Gluck, 1995; Kubat, Holte, Matwin, Kohavi, & Provost, 1998). This type of method is inspired by the auto-association based classification approach proposed by Japkowicz et al. (1995). The training process involves a MLP neural network to reconstruct its input at the output layer. After training, an auto-associator is used for classification based on the idea that the network can reconstruct the input at the output layer accurately, i.e. if MLP can create a novel instance, then the instance must belong to the class that was used for training; otherwise, if the instance creation fails, then the instance must belong to the other class.

Although this issue of the classification with imbalanced class distribution data has been known for a long time, it is still an open research question. There is no universal method that can work for the classification on all different imbalanced class distribution datasets. Down-sizing methods work efficiently in large problem spaces, while over-sampling method may perform well in small problem space. Recognition-based methods have been reported to be a better alternative in some cases (Japkowicz &

Stephen, 2002). The solution to imbalanced data classification problem depends on each given task and the object dataset.

In the context of a personalised modelling study, we have designed a simple schema to balance the class distribution for neighbourhood construction. The ratio between majority class samples and minority class samples is pre-specified. In short, the schema checks the class distribution of the neighbourhood for every candidate solution. It will extend the neighbourhood size, if there are not enough minority class samples included. Chapter 7 will use this schema to implement the proposed PMS.

5.3 Classification Models

Classification is of critical importance in PM. A number of classification algorithms have been developed in the past, such as artificial neural network based algorithms, decision tree methods, Naive-bayes classifier, nearest neighbour based algorithm, Bayesian statistics, SVM, etc.

5.3.1 Classification Models in Medical Applications

This section gives a review of some representative classification methods used for medical applications.

Correlation based Classification Method

The weighted voting method is proposed by Golub et al. (1999) for classifying DLBCL data and is known as GS method. This method is one of the pioneer studies in microarray gene expression research, and is based on the correlation evaluation. GS method assigns the class for the testing sample based on the weighted voting calculated by the expression values of a subset of informative genes from the testing pool.

The informative genes are selected based on their correlation values with class labels. Let the expression values of a gene in n training samples be represented by a vector

$g = (x_1, x_2, /cdots, x_n)$, where x_i is the expression value of gene i. Another vector $y = (y_1, y_2, /cdots, y_n)$ is the class label vector responding to each sample. Let $\mu_1(g), \sigma_1(g)$ and $\mu_2(g), \sigma_2(g)$ be the mean and standard deviation of the $\log_{10}$ of the value of g in class 1 and class 2, respectively. Thus, the correlation - $r(g, y)$ between the expression values of gene g and the class label y is calculated by SNR function as follows:

$$r(g, y) = \frac{\mu_1(g) - \mu_2(g)}{\sigma_1(g) + \sigma_2(g)} \tag{5.12}$$

The value of $\mid r(g, y) \mid$ is proportional to the correlation between g and y. The correlation $r(g, y)$ identifies to which class the gene g is more correlated. The larger the weight, the stronger the correlation.

GS method selects $L/2$ genes with the highest positive r values and $L/2$ genes with the highest negative values to consolidate a set of informative genes, where L is a pre-specified value.

Then, the class label of a testing sample x_γ is determined by a voting schema:
for each informative gene g_{inf} in the testing sample x_γ, the value of g_{inf} is normalised by $\log_{10}$ and denoted as $g_{nor} = log_{10}((x_{inf} - \mu)/\sigma)$, where x_{inf} denotes the value of an informative gene of a testing sample. The vote from gene g_{inf} is given as:

$$v_g = r(g_{inf}, y_\gamma)(g_{nor} - \mu_1(g_{inf}) - \mu_2(g_{inf})) \tag{5.13}$$

where the sign of the vote indicates the class.

Therefore, informative genes will create a "weighted vote" vector for one class. The final vote is calculated by:

$$D_s = \frac{V_{win} - V_{lose}}{V_{win} + V_{lose}} \tag{5.14}$$

where V_{win} is the number of votes of the winning class (i.e. the class that has the higher number of votes), while V_{lose} is the number of votes for the losing class, D_s denotes the degree of prediction strength.

To calculate the final prediction result, a threshold θ_γ for classification is specified for determining class to which the testing sample belongs. If $D_s \geq \theta_\gamma$, then the testing sample x_γ is assigned the winning class label. Otherwise, the weighted voting is thought to be not strong enough to make a decision, thus the class label of testing

sample remains uncertain.

Probabilistic Based Classification Method - Naive Bayes Method

Naive bayes Classifier employs probabilistic learning to classify the testing samples. Assume that a data vector $X = (f_1, f_2, \cdots, f_m)$, where f_i is the features to represent the data x_γ and all of them are conditionally independent of one another. Y is the class label responding to X. Thus, the probability model for a classifier is formulated as:

$$P(f_1, f_1, f_2, \cdots, f_m) \mid Y) = \prod_{i=1}^{m} P(f_i \mid Y) \tag{5.15}$$

For each new data sample x_γ to be classified, the prediction class label y_γ is determined by:

$$y_\gamma = \underset{y_k}{\operatorname{argmax}} P(Y = y_k) \prod_i P(X_i \mid Y = y_k), \quad i = 1, 2, \cdots, m \tag{5.16}$$

where y_k denotes class k. Keller, Schummer, Hood, and Ruzzo (2000) used Naive bayes algorithm for DNA array expression data analysis, where the class was modeled by a Gaussian probability function.

The main limitation of Naive bayes classifier is that it neglects the relationship among features, because the algorithm of bayes rule is made by the assumption that all features are conditionally independent. Such issue may cause the testing accuracy from a Naive bayes classifier to be inconsistent with the training accuracy over some difficult datasets.

Nearest Neighbour Based Classification Method - KNN

The main idea of this category of classification methods is based on the similarity measurement for the testing and training samples. KNN is probably the most well-known algorithm for classification. Using KNN classifier, the class labels of the testing samples are assigned by the majority vote from K samples from the training set which are most similar to the testing sample according to the distance (usually an Euclidean distance) measurement.

It is obvious that the value of K impacts the classification performance. How to determine the best value of K for KNN classifier is still an open research question. The potential solution for optimising K in KNN will be discussed in later sections.

Max-Margin based Methods

Max-Margin based classification methods aim to find an hyperplane that is able to separate the problem space into different groups according to the number of classes. The margin of the hyperplane is defined as the distance from the hyperplane to the closet groups of data points. The larger the margin, the better the hyperplane. Thus, if a classifier is able to separate the data points with a maximized margin, it can be less subjective to overfitting and gain better classification results (Lu & Han, 2003).

Max-Margin based classifiers can be a good choice for dealing with microarray gene expression data that has very sparse data points in a large dimensional space (Smola et al., 1999; Freund & Schapire, 1998). Here we give an introduction to Support Vector Machine algorithm - a popular Max-Margin based classifier.

Suppose for a data set pertaining to a binary classification task, each data point is represented by $X = (f_1, f_2, \cdots, f_m)$, f_i is the features, and Y is the class label corresponding to X, $Y \in 1, -1$. For small training data set with large feature space, SVM classifier constructs a hyperplane with maximum margin that is able to separate the positive data points from the negative ones. The classification performed by a SVM classifier on a new testing sample x_γ is given by:

$$Cls(x_\gamma) = sign(y_\gamma(\langle \omega_0, \phi(x_\gamma) \rangle - b_0) \tag{5.17}$$

where ω_0 and b_0 represent the vector and scalar in SVM (refer to description of SVM algorithm in Chapter 3). If the calculated sign is positive, it means y_γ is correctly classified, otherwise is misclassified. A number of SVM based algorithms have been proposed for classification problems. Such works include Soft margin and margin-distribution classification method developed by Shawe-taylor and Cristianini (1999), and the classification method for ovarian cancer gene expression data analysis (T. S. Furey, Cristianini, Duffy, W, & Haussler, 2000).

5.3.2 The Challenges of Classification for Personalised Modelling

Classification problems have been extensively studied in the research community of statistical, machine learning and data mining. However, the application of classification in personalised modelling poses new challenges due to its unique nature.

The first challenge comes from the structure of microarray gene expression data. As we have already discussed in Section 5.1, the unique structure of cDNA microarray gene expression data prevents traditional classification algorithms working properly. In most available gene expression datasets, the sample size if very limited, while the dimensionality of features (genes) is enormous. Traditional classification algorithms are not designed to deal with this kind of datasets. Such a characteristic of sparseness and large dimensionality becomes a big challenge for most existing classification algorithms. The large dimensionality of features often introduces an overfitting issue, which may result in increase of the validation error while the training error steadily decreases. The small size of samples makes the situation worse.

The second challenge involves the improvement of the effectiveness and efficiency of classicalism algorithms. Within the scope of personalised modelling system (PMS), every new data vector will have its own unique model that usually contains a classifier. Such scenario makes the computation very costly if the classifier is not efficiently designed. The performance from the classifier is another critical factor for predicting new coming data vectors.

The third challenge arises from the application domain of classification. Accuracy is generally considered most important for classification problems, but it is not the only goal to achieve in personalised modelling study. For medical purposes, biological relevancy is a critical factor, because any biological information discovered during the learning can be used for further study, including tailored treatment for individual patients, designing new drug based on the findings, etc. Useful information might be gained from the classification process, e.g. the identification of a group of genes working together in determining the cancerous tissues or cells (Lu & Han, 2003). All the information would assist researchers in gaining deeper insight about the genes and how they interact with each other. Therefore, biological or medical researchers are often more interested in those classifiers that not only yield high classification

accuracy but reveal important biological information.

One way to overcome the first two challenges is to incorporate feature selection methods to identify a compact set of informative features (e.g. highly differentially expressed genes). The classifiers can be built based on these informative features, which will significantly improve the classification accuracy and reduce the computational difficulty.

Regarding the third challenge, personalised modelling can produce a good platform for classifiers to discover important biological information, along with the classification accuracy measurement scheme. The proposed PMS creates a model that comprises a classifier and relevant parameters, and contains useful information for the testing data sample, such as the potential improvement of gene expression level, the most important features for disease diagnosis specifically for the patient to be tested, etc.

A number of classification models have been developed for different types of classification tasks. Lu and Han (2003) have summarised some popular classification algorithms in Table 5.1.

Table 5.1: *The summary of some commonly-used classification algorithms. Adapted from Lu and Han (2003)*

Classification algorithm	Category	Multi-class	Biological meaningful	Scalability
GS (Weighted voting)	Correlation based	No	Yes	Fair
Naive Bayes	Probability	Yes	No	Fair
SVM	Max-Margin	No	No	Good
KNN	Similarity	Yes	No	Not Scalable
Decision Tree	Entropy Function	Yes	Yes	Good
Neural Network	Perceptrons	Yes	No	Fair

Nevertheless, one thing we need to bear in mind is that there is no single classifier that can be always superior over others. Some classifiers work efficiently over well-balanced structured datasets, while others may perform properly on datasets with high dimensionality and small sample size. Therefore, to construct personalised models, the classifier needs to be specifically designed for the given problem.

5.4 Model Parameter Optimisation

It is a big challenge to optimise parameters for the development of personalised modelling. As described in Chapter 4, a global model builds a model that is expected to perform well on any given data with same scenario of analysis problems. Once relevant parameters are optimised for a trained model, such as the coefficients of a regression function, the maximum and minimum radius for a cluster, etc, there is no need to optimise these parameters again. However, unlike global modelling, personalised modelling builds a specific model for each individual data sample. To obtain an efficient and reliable personalised model, the relevant parameters should be optimised specifically for each individual data sample, i.e. the parameters used for different personalised models can be significantly different, even the models are built for the same classification problem (e.g. for same type of disease diagnosis from the same dataset).

5.4.1 Selecting the Appropriate Neighbourhood and Classification Threshold

The proposed personalised modelling framework and system (PMFS) require a set of parameters to be used for building personalised models. One important step for the creation of personalised models is to find an appropriate personalised problem space, i.e. the most appropriate number of nearest neighbouring samples (K_x) that can represent the pattern of the given testing sample. Also, some thresholds need to be optimised to suit the creation of personalised models. They can be the threshold for classification, the threshold in relation with clustering, etc. In traditional models, a threshold is usually specified before the learning process starts, and then is optimised by an optimising function. Once the optimal solution is obtained, the thresholds will be used for testing any new samples in the same problem category.

The most straightforward way to optimise different parameters is the exhaustive search, in which all the possible combinations of parameters will be assessed. However, this becomes a formidable challenge in practice, because the parameter optimisation brings huge computational complexity during the development of each personalised model. Hence, finding an efficient solution to the parameter optimisa-

tion is a fundamental step towards the successful implementation of PMS.

5.4.2 Discussion and Possible Solution

Heuristic learning can be a solution for parameter optimisation in the development of personalised models. It uses the reinforcement learning to seek an automate solution for determining a proper search direction when an optimisation task is given. The learning occurs while a search algorithm is solving an instance of a given problem.

In order to build a proper model, we have proposed a solution for parameter optimisation within the scope of personalised modelling study. It starts with a set of pre-specified parameters that can be obtained from historical experimental results or suggestions from literature. Then, these parameters will be tuned by a learning function within the training process. GA can be a good tool to use if the search algorithm and terminating criteria are carefully designed. Principally, GA-based approaches for parameter optimisation are able to find the optimal or near optimal solution for the parameters in relation with personalised modelling.

5.5 Data Sampling

When analysing microarray data, selection of a data sampling method is important for the verification of final experimental results (Allison et al., 2006; Braga-Neto, Hashimoto, Dougherty, Nguyen, & Carroll, 2004), because an improper sampling method often leads to biased and unreplicapable results (Zhu et al., 2003). A number of published studies claimed that they achieved a very high accuracy (close to 100%) from classification over different cancer gene expression datasets, such as the breast cancer study presented by Ramaswamy and Perou (2003), and the analysis on ovarian cancer data by Zhu et al. (2003). However many of them are reported unreplicapable by other laboratories. Ransohoff (2004) reported that these tests failed to be reproduced because the process of validation (i.e. the sampling method) was not well developed.

In the machine learning literature, several sampling methods are recognised as unbiased verification methods, such as resubstitution, cross-validation, and bootstrap

(Efron, 1979). A brief review of two popular sampling techniques, namely K-fold cross-validation and bootstrap is presented below. These two methods are discussed in terms of disadvantages and advantages.

5.5.1 Cross-validation

Cross-validation is a sampling technique extensively used in micorarray data analysis (Ambroise & McLachlan, 2002; Qiu, Xiao, Gordon, & Yakovlev, 2006). According to Ransohoff (2004), cross-validation is "a technique used in multivariable analysis that is intended to reduce the possibility of overfitting and of non-reproducible results. The method involves sequentially leaving out parts of the original sample ('split-sample') and conducting a multivariable analysis; the process is repeated until the entire sample has been assessed. The results are combined into a final model that is the product of the training step" (p. 312).

The advantage of cross-validation is that all the data can be used for cross training and testing, and the validation is totally independent of the training process. In the context of microarray data analysis, for cross-validation purposes, the dataset is randomly partitioned into two subsets, training and testing set. Indeed, the goal of implementing cross-validation is to evaluate whether the result is replicable or just caused by chance.

Cross-validation can be generally performed in two ways: K-fold cross-validation and leave-one-out cross-validation (LOOCV). In K-fold cross-validation, samples are randomly divided into K mutually exclusive subsets of approximately equal size. The validation process will be repeated for K rounds, where for each round, K-1 subsets are used for training (e.g. classifier training), and the remaining one subset for testing. For small dataset analysis (e.g. microarray gene expression data), 5 or 10 folds are generally suggested for cross-validation in literature (Breiman & Spector, 1992; Kohavi, 1995). LOOCV eventually is a K-fold cross-validation, where K equals the number of samples (N) in given dataset. In LOOCV, all the samples are separated N rounds, where for each round, all samples are used for training except one is left for testing. The final result is the average performance over N testing sets.

For many years, LOOCV has been suggested for evaluating classification performance over data with a very small number of samples, as it is a nearly unbiased method

and works well for estimating bias error, such as the mean squared error. However, Breiman and Spector (1992) have demonstrated that a high variance of LOOCV rises when the prediction rule of the method under verification is unstable. This is mainly because LOOCV sampling makes the training set very similar to the whole dataset.

5.5.2 Bootstrap Resampling

Bootstrap, first introduced by (Efron, 1979), is a sampling method for small sample size dataset. Empirical studies have shown that bootstrap is particularly effective for estimating bias error for very small sample size, such as microarray data (Efron, 1983). More recently many bootstrap estimators have been proposed, among which $e0$ and the .632 bootstrap are two popular methods that can yield good results when sampling in classification problems.

The principle of bootstrap method is data sampling with replacement. Suppose a dataset contains only 5 samples labeled A, B, C, D and E. The bootstrap sampling with replacement can be simply described as follows:

1. Randomly draw out one of 5 samples and record its label.
2. Put the sample back to the dataset.
3. Repeat **Step 1-2** N times (N is a constant integer) to have N labels in a sequence.
4. Randomly select a subsequence of 5 labels from the sequence obtained in **Step 3**, and extract the corresponding samples as the training set (the first round).
5. Repeat **Step 1-4**, to construct the testing set.

5.5.3 Comparison of Cross-validation and Bootstrap Methods

Cross-validation has a disadvantage that the training lacks sufficient information due to small size of the dataset. Therefore, in the case of partitioning a microarray dataset, cross-validation technique may increase the risk of *overfitting*. Critical

scientific issues are raised in literature in relation to the use of cross-validation for generalisation error estimation (Braga-Neto et al., 2004). However, cross-validation is still considered a robust and unbiased technique in microarray data analysis, if experiments are well designed and organised (Asyali et al., 2006).

Bootstrap uses a replacement resampling approach, and constructs training and testing sets with the exact same size as the whole dataset, while in cross-validation, both training and testing sets use only a subset of the whole dataset. Thus, the bootstrap method has an advantage of modelling the impacts of the actual sample size. The disadvantage is that the bootstrap method yields a good result only after hundreds of iterations, which makes it more computationally costly than cross-validation. In this study, cross-validation is employed as a validating method, due to its efficiency and robustness.

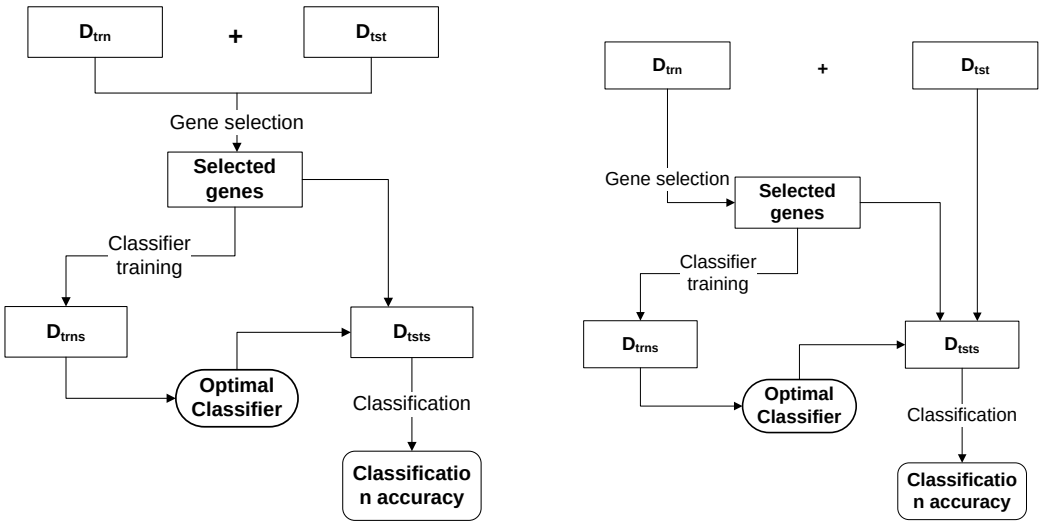
5.5.4 An Unbiased Validation Schema

An unbiased verification approach for microarray analysis should guarantee that generalisation errors occur in either feature selection or classification procedures as little as possible. To this end, an efficient data sampling method should be used in the two procedures to maximally decrease the generalisation error. In other words, the reliability and generalisability of the informative features should be evaluated on independent testing subsets, and then these features can be used for classification. The classification also needs to employ verification methods to estimate the bias error. Such procedure is shown in Figure 5.4(b). For comparison, a simple example of biased validation schema is demonstrated in Figure 5.4(a).

5.6 Error Measuring Methods

There are three commonly used error measuring methods to estimate the testing error in models related to classification problems:

- *The classification error* (the number or percent of the samples misclassified). This is probably the most straightforward and best-known method for validat-



(a) An example of biased validation scheme; (b) The proposed unbiased validation scheme

Figure 5.4: *The comparison between a biased and an unbiased verification scheme, where D_{trn} and D_{tst} are the training and testing set, D_{trns} and D_{tsts} are the training and testing set with selected genes, respectively. In case (a) (biased verification scheme), the testing set is used twice in gene selection and classifier training procedure, which introduces a bias error from the gene selection stage into the final classification step. Whereas in case (b) (the unbiased scheme), the testing set is only used in the final classification(validation) stage, i.e. the testing set is independent all through gene selection and classifier training procedures.*

ing classification models. It is simple and easy to interpret and has been widely accepted for classification experiments.

- *Root-mean-square-error (RMSE)* . The RMSE error for a testing data set can be calculated as follows:

$$rmse(\theta) = \sqrt{\frac{\sum_{i=1}^{n} e_i^2}{n}} \tag{5.18}$$

where e_i is the difference between the outcome and observed data, n is the number of cases. The RMSE error is used to characterise the posterior probability vector miscalculation.

- Receiver operating characteristic (ROC) curve is a technique used for visualising and selecting classifiers based on their performance.

Root-mean-square error (RMSE) is the mean square error of an estimator which quantifies the difference between the predicted value of a model (an estimator) and the actual value of the sample being estimated. Simply, RMSE is a risk evaluation function that corresponds to the observed value of the root squared error loss. RMSE is widely used to evaluate the performance of regression models, which allows to aggregate all variances between predicted value and actual values of observed samples into a single measure of predictive power.

5.6.1 ROC Curve: a Performance based Measuring Technique

ROC curve is a technique used for visualising and selecting classifiers based on their performance. It has long been used for evaluating classifier performance in signal detection (J. A. Swets, Dawes, & Monahan, 2000) and for visualising and analysing the behaviours of classification performance in diagnostic systems (J. Swets, 1988). Recently, ROC analysis has received extensive attention from the medical decision making community for diagnostic testing (Fawcett, 2004).

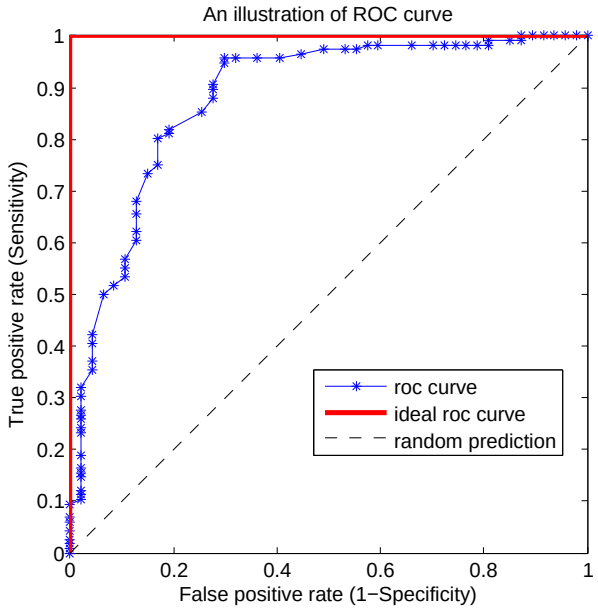


Figure 5.5: *An example of roc curve*

A ROC curve is plotted in a two-dimensional space in which *true positive rate* (TPR) is on the y axis while *false positive rate* (FPR) is on the x axis. ROC curve method is used to decide the shape and position of the line that separates the groups of 'normal' and 'abnormal' data samples. In the medical area, ROC curve is proposed to help clinicians make decisions for the calculation of the *sensitivity* and *specificity* of the test at different *cut-off* points. The definition of sensitivity and specificity is given as follows:

- Sensitivity: The proportion of people with the disease that the test successfully identifies as positive.
- Specificity: The proportion of people without the disease that the test successfully identifies as negative.

Figure 5.5 gives an example of ROC curve for a classification task, where the blue line with stars is a computed ROC curve, and the red solid line represents the perfect prediction.

It is clear that an easy decision can be made if all the control values are higher (or lower) than all the patient values. However, the overlap of two distributions makes the situation not so easy. If the threshold is high, many people who do not have the disease can be correctly diagnosed, but some of the people who have the disease are more likely to be misclassified into a healthy group. On the contrary, if the threshold is low, most of the people with the disease will be successfully identified, but more healthy people may be mistakenly diagnosed as diseased.

Based on ROC curve, researchers can calculate the sensitivity and specificity using each value in the data as the *cutoff* value. A number of pairs of sensitivity and specificity can be investigated. For example, with a high threshold, the specificity of a test is increased, while sensitivity is decreased. Similarly, the lower threshold may bring the issue of increases of sensitivity but decreases specificity.

The area under a ROC curve quantifies the overall ability of the test to discriminate between the diseased individuals and the healthy people. A truly useless test (one no better at identifying true positives than flipping a coin) has an area of 0.5. A perfect test has an area of 1.00, which means it has zero false positives and zero false negatives. Generally, a test will have an area between those two values.

5.6.2 Discussion

Classification accuracy is most commonly used in error measurement, owing to its simplicity and robustness. The comparison result based on accuracy is straightforward and easy to interpret. The limitation of this technique is that it may ignore the information from domain knowledge, e.g. biological reference, if the learning model is not carefully designed.

RMSE may be an inappropriate technique to measure generalisation error in personalised modelling study under some scenarios. Here is an example:
Suppose there are two models M_α and M_β, and a sample $x_\gamma = 0.35$ (from a healthy group) is given to be classified. The threshold for determining the class is set to 0.5 (if the predicted risk is less than 0.5, then the sample is classified as healthy, otherwise it is classified as diseased.). With two models M_α and M_β, the prediction risk of x_γ calculated by M_α is 0.6 ($RMSE_\alpha = 0.2$) , while the risk computed by M_β is 0.1 ($RMSE_\beta = 0.3$). In this case, it is incorrect to conclude that M_α performs better because of the smaller RMSE. On the contrary, model M_β correctly gives the prediction to sample x_γ, though it creates a large RMSE.

Although it is obvious that the area under a ROC curve indicates the overall testing ability to successfully discriminate between normal and abnormal samples, the interpretation of the area itself can be very intuitive. For example, if patients have higher test values than control threshold, then the area represents the probability that a randomly selected patient will have a higher test result than a randomly selected control, and vice vesa. If the area equals 0.75, a patient will have a more diseased test result than 75% of the controls on average (i.e. a higher diseased diagnosis probability). If the test is perfect, every patient will have a more abnormal test result and the area would be 1.00. If the test is useless (i.e. no better than the identification of normal versus diseased samples by chance), then the patient will have the equal possibility to be found diseased or healthy. Thus, the area under the curve would be 0.5. If the area is calculated less than 0.50, the definition of abnormal from a higher test value to a lower test value can be reversed. This adjustment will result in an area under the curve greater than 0.50.

In this research, mainly classification accuracy and ROC curve are used as error measuring methods during personalised modelling construction, due to their simplicity

and efficiency.

5.7 Inconsistency Problem and Local Accuracy

The reproducibility of microarray gene expression data analysis is a critical factor for determining the quality of cancer gene expression data experiment. For many cancer gene expression data analysis, the results of operations (such as clustering, classification, etc.) on the training dataset (a subset of a complete cancer microarray dataset), have been found often very different from those of the same operations on the testing dataset (another subset of the complete cancer microarray dataset). This is defined as the *inconsistency issue*. In practice, this inconsistent response becomes a critical issue for evaluating the reliability of cancer gene expression data experiment results.

In the context of PMS, we introduce another accuracy - *local accuracy*. Local accuracy is defined as the accuracy calculated by a classifier within the personalised problem space during training process. In this study, it is incorporated into the learning function to optimise candidate personalised models within training process. The local accuracy is different from the training accuracy. The latter is calculated based on the classification on all training samples and is usually significantly higher than the testing accuracy. The local accuracy is calculated based on the samples in the personalised problem space that can more precisely represent the patterns of new testing sample. Thus, local accuracy should be more likely to be close to the testing accuracy.

Being used for evaluating the candidate personalised models, local accuracy should be more consistent in relation to the testing accuracy. However, the inconsistency issue between local accuracy and testing accuracy still exists in our experiments, though it is less significant than that between training and testing accuracy. This issue has been demonstrated in the experiments in Chapter 4.

To deal with the inconsistency issue between local and testing accuracy, we need to find the appropriate personalised space for the testing sample, i.e. the space where the samples used for learning can highly represent the testing sample's pattern. Moreover, the appropriate size of personalised space need to be identified. Too few

data samples may not include sufficient information, while too many samples may introduce a lot of noise information that will confuse the classifiers. Therefore, how to choose the number of samples and which ones should be included in the personalised space are two fundamental factors for personalised model construction. One way to handle this problem is to incorporate an automate schema to find an optimal personalised space.

5.8 Profiling and Visualisation

Personalised profile is a major contribution that the personalised modelling approach offers. The profile comprises the information that may need to be modified for the design of personal scenarios improved that can be used for potential applications, such as personalised medicine, personalised drug design for complex diseases (e.g. cancer, diabetes and brain disease), finance risk evaluation, etc. The information may include: number of variables (features), which variables (features) are important for the given analysis problem; the predicting risk for the new testing data vector; the difference between the actual value and the desired value of important variables.

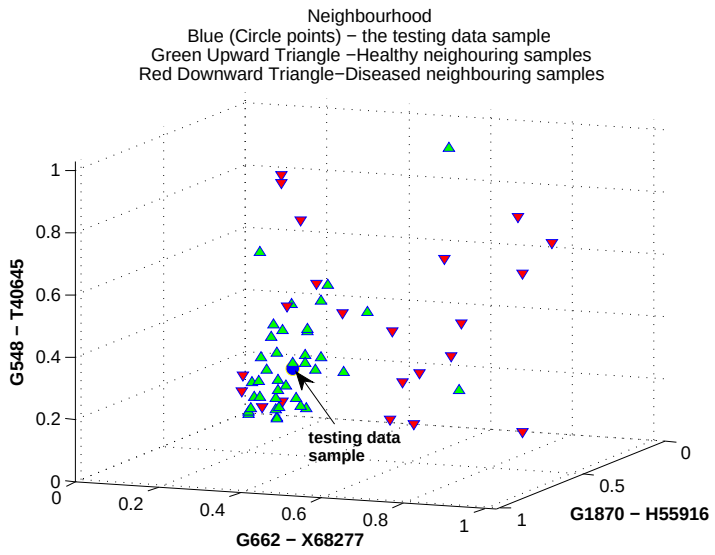


Figure 5.6: *The neighbourhood of sample 1 of colon cancer data visualised in a 3-D space (3 genes: gene 249, 267 and 1674), where blue circle points are the new input data sample, green up-triangle points are the normal neighbouring samples, and red down-triangle points represent the diseased neighbouring samples*

It is not an easy task to interpret the new data vector's profile that involves high diversity of variables and sparsity of data vectors. So far, there is no PM methods that offer profiling. This study attempts to develop some methods for PM profiling.

In the proposed PMS, visualisation creates a paradigm shift in the interpretation of important variables (features) to profile the new input data vector. Using the most informative variables, a created personalised profile allows to visualise the comparison of new data vector's important features against those associated with a desired outcome. For simplicity of interpretation, the visualisation is designed to be plotted in a 2-D or 3-D space. Figure 5.6 gives a 3D demo of the neighbourhood of a sample from a Colon cancer dataset using three features (genes).

The visualisation includes profiling the personalised space corresponding to the new input vector. The visualisation shows the distribution of new data vector's neighbouring samples. A scenario of potential improvement for new data can also be shown by the visualisation. The details will be described in Chapters 7 and 8.

5.9 Conclusion

In this chapter, we have addressed several issues that have arisen during the development of personalised modelling based framework. These issues can arise due to different factors, including the unique nature of the data structure, the optimisation of parameters, classification problems, to name but a few. In order to ensure a successful construction of personalised modelling for a given analysis task, it is necessary to study and understand these issues.

With the aim to find potential solutions for the issues raised by the study of personalised modelling, this chapter has reviewed the areas of feature selection, classification, data sampling, error measuring, etc. It has also explored and discussed a variety of algorithms and models in relation to this study. The next chapter will propose a personalised modelling system for data analysis and knowledge discovery, and will discuss a few case studies where this framework has been implemented.

CHAPTER 6

A Personalised Modelling Framework (PMF) and A Methodology for Implementing Personalised Modelling Systems (PMS)

"When solving a problem of interest, do not solve a more general problem as an intermediate step. Try to get the answer that you really need but not a more general one"

- Vladimir N. Vapnik

This chapter presents the methodology to build an integrated framework for personalised modelling and illustrate the data analysis and knowledge discovery on some particular benchmark data. In the previous chapters, I have addressed the issues that global models cannot provide precise and sufficient information for analysing a new incoming data vector under different circumstances, and the selected features are not informative enough to lead to a successful classification. Moreover, it is difficult to incorporate previously developed models and existing knowledge into global modelling methods. In order to find a more effective approach for analysing new data

vectors, this chapter proposes a personalised modelling framework (PMF) and the methodology for implementing a personalised modelling system (PMS). This PMF comprises several functional modules, such as feature selection, classification models, optimisation modules, etc. The chapter also gives an example to implement a PMS using a simple approach for knowledge discovery using biomedical data. The proposed PMS is applied on three case studies for cancer diagnosis using benchmark cancer gene expression datasets.

6.1 The PMF

The concept of personalised medicine has been promoted widely in recent years through the collection of personalised databases, establishment of new journals and new societies and publications in international journals. Despite the furor of interest in this area, there are at present no adequate data analysis methods and systems that can create highly accurate and informative personalised models from data.

The methods and systems particularly related to personalised data analysis and decision support system are based on the use of an individual's information, including gene expression level, proteomics patterns, clinical and cognitive data, etc. The methods are adaptive and evolving through incremental addition of new data for an adaptive learning. The method can be applied on different types of problems, such as cancer diagnosis and prognosis using benchmark microarray gene expression, proteomics pattern data analysis, and other types of data analysis. The framework comprises applications in computer science, mathematical modelling, profiling and prognostic systems to predict outcomes and evaluate risks for new data based on the information discovered from historic data.

The philosophy behind the proposed PMF is the realisation that every person is different, and preferably each individual should have their own personalised models and tailored treatment. In the context of medical research, it has become possible to utilise individual data for a person with the advance of technology, e.g., DNA, RNA, protein expression, clinical tests, inheritance, foods and drugs intake, diseases. Such data is more readily obtainable nowadays, and is easily measurable and storable in electronic data repositories with less cost.

With a transductive approach, each individual data vector that represents a patient in any given medical area obtains a customised, local model that best fits the new data. This is contrary to using a global modeling approach where new data is matched to a model (function) averaged for the entire dataset. A global model may fail to take into account the specific information particular to individual data samples. Moreover, there are no efficient methods for identifying important features that assist complex disease classification, e.g. which genes, SNPs, proteins and other clinical information contribute to the disease diagnosis. Hence, a transductive approach seems to be a step in the right direction when looking to devise personalised modelling useful for analysing individual data sample, e.g. disease diagnosis, drug design, etc.

KNN is a simple classical transductive inference method that calculates the output for a new data vector based on the average output values of its K-nearest samples from the given data set. Some more sophisticated transductive reasoning methods TWNFI (Song & Kasabov, 2006) (see Appendix F) and TWRBF (Song & Kasabov, 2004) have been proposed for solving the problems requiring individual modelling analysis. These methods create a learning model based on the neighbourhood of new data vector, and then apply the trained model on the new data to calculate the output. However, this type of methods cannot select features and related parameters, such as what is the appropriate number of neighbors and how many features will be best fit for the classification problems. Also, there is no existing methodology to yield the information necessary for designing individual patients' treatment.

Inspired by the concept of genomic personalised medicine (Ginsburg & McCarthy, 2001; Shastry, 2006; Anderson et al., 2006), a personalised modelling based framework was introduced by Kasabov (2007b, 2007a) for data analysis and knowledge discovery. The concept of personalised medicine has been intensely researched in recent years (Kasabov, Hu, & Liang, 2009; Gurwitz, Lunshof, & Altman, 2006; Garrison & Austin, 2007). Pharmacogenomics research is currently conducted for the medical application of human genetic data for personalised drug development. The idea of personalised treatment is that an individually designed drug can significantly benefit by using a person's genetic information, and might not benefit other people having the same disease. Such approach brings the potential to improve drug effectiveness and reduce drug side-effects. Nevins et al. (2003) developed integrated clinico-genomic models for designing personalised medicine for breast cancer outcomes prediction. Their models used the information from the combination of gene

expression levels and clinical factors, which provided a more effective mechanism to characterise individual patients in terms of the performance of clinical outcomes prediction.

Here an outline of personalised modelling framework (PMF) is depicted in Figure 6.1.

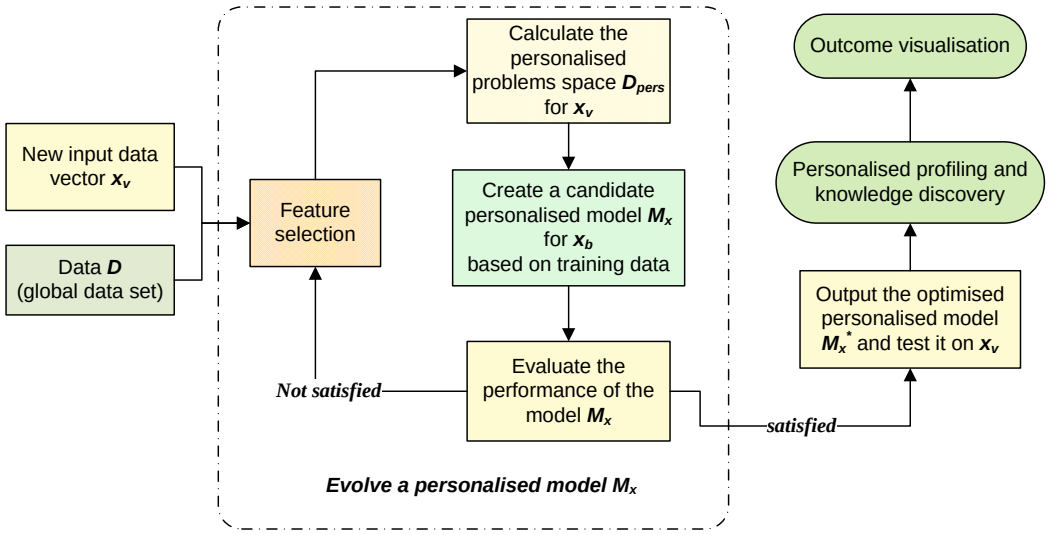


Figure 6.1: *A PMF for data analysis and knowledge discovery.*

This framework is initially designed for medical data analysis and knowledge discovery. However, PMF can be extended for various types of data analysis problems that require personalised modelling. PMF can be briefly described as follows:

1. Apply feature selection on the object data D (the global problem space) to identify which features are important to a new input vector x_v. The selected features are grouped into a candidate gene pool;

2. Select K_v nearest samples for x_v from D to form a local (personalised) problem space D_{pers};

3. Create a personalised model candidate M_x specifically for x_v, which includes a learning function (usually a classifier or a clustering function) denoted by f;

4. Evaluate the candidate feature subset $\mathcal{S}$ by a learning function f based on their performance within the personalised problem space D_{pers};

5. Optimising model M_x through an evolving approach until termination conditions are met. The output is the optimal or near-optimal solution to vector x_v. The solution includes an optimal personalised model M_x^* and a selected feature subset $\mathcal{S}^*$;

6. Use the model M_x^* to test the new vector x_v and calculate the outcome y_v;

7. Create a personalised profile for the input vector x_v, visualize the outcome with the selected important features $\mathcal{S}^*$, and provide an improvement scenario for data vector x_v for a given problem if it is possible.

6.2 A Methodology for Using the PMF to build a PMS

The core task of a PMS is to create a personalised model for each new input data sample using its unique information. Given a dataset D pertaining to a bioinformatics problem, $D = \{x_{ij}, y_i, i = 1, \cdots, n, j = 1, \cdots, m\}$, where x is a data sample, y is the responding outcome, n is the number of samples, m denotes the number of features (variables). The proposed method aims to optimise a model M_x suitable for analysing data, specific to every new input data vector x_v, e.g. to calculate y_v - the outcome of x_v. Data x_v contains a number of features that are related to the same scenario as the data samples in the global data D.

In order to obtain the optimal or near optimal personalised model M_x^* specifically for a new data sample x_v, the proposed method aims to find the solutions to the following objectives:

1. Determine how many and which features (variables) S are most suitable for building the model M_x^* that is able to successfully predict the outcome for the new data vector x_v;

2. Determine the appropriate number K_v for the neighbourhood of x_v to form a personalised problem space D_{pers};

3. Identify K_v samples from the global data set D which have the pattern most similar to the data x_v, and use these K_v samples to form the neighbourhood (a personalised problem space D_{pers});

4. Calculate the importance of selected features S within the personalised problem space D_{pers}), based on their contribution to the outcome prediction of the data vectors in D_{pers}. Compute a weight vector w_v for all selected features S;

5. Create the optimal personalised model M_x^* with the optimised parameters obtained in Steps 1∼4;

6. Validate the obtained model M_x^* by calculating the outcome y_v for the new data x_v;

7. Profile the new input data x_v within its neighbourhood D_{pers} using the most important features associated with a desired outcome;

8. If possible, provide the scenarios for improving the outcome for the new data vector x_v, which can be helpful for clinical use.

This is a method for determining a profile of a subject (new input vector x_v) using an optimal personalised model M_x^*, and for recommending the possible changes to the profile in relation to a scenario of interest in order to improve the outcome for x_v. The method comprises the following steps:

- Create a personalised profile for a new data vector x_v;
- Compare each important feature of input data vector x_v to the average value of important features of samples having the desired outcome;
- Determine which important features of input vector x_v can be altered in order to improve the outcome.

Principally, the decision of which variables should be changed will be based on the observation of the weight vector W_x of features (i.e. the contribution of the features to the classification). The term "*personalised profile*" used here refers to an input vector x_v and to its predicted outcome and related information, such as the size of its neighbourhood, its most important features specifically, etc.

Within the scope of PMS, the proposed method for building an optimal model M_x require the following functional modules:

- A module for selecting most relevant V_v features (variables) S^* and ranking their weighter w_x by importance for x_v;
- the module for the selection of a number K_v of neighbouring samples of x_v and for the selection of neighbouring samples D_{pers};
- A module for creating a prediction model M_x, defined by the a set of parameters P_v, such as K_v, V_v, D_{pers} which were derived in the previous modules;
- A module for calculating the final output y_v responding to the new data x_v
- A module for the creation of personalised profile and the design of scenarios for potential improvement.

6.3 A Simple Method for PM - An Incremental Search-based PMS (iPM)

The proposed method and system for PMS construction can be implemented in different ways. In this section, a simple approach for implementing a PMS is presented. This approach is based on incremental search and denoted as iPM method. The presented iPM has been applied on several benchmark datasets related to 3 types of cancer for disease classification.

The iPM method is developed for searching a combination of features and parameters to build optimal personalised model M_x^*:

1. Find an appropriate neighbourhood (the personalised problem space) for new input data sample x_v;
2. Generate a candidate personalised model M_x along with a set of features and relevant parameters S^*, K_v within the scope of the created personalised problem space in Step 1;

3. Evaluate the created model M_x to determine whether to remove or keep the candidate features depending on the evaluation performance;

4. Iterate the above process until all the features are evaluated or termination conditions are reached.

The optimal personalised model M_x^* is expected to be obtained when all features have been investigated or the termination criteria are reached (e.g. the best performance is obtained or all the features are evaluated).

The method of iPM searches new candidate features in the following way:

1. Update candidate gene set g_i by inserting a gene g' from the candidate gene pool g_ρ;

2. Evaluate the classification performance using the updated candidate genes.

3. ***if*** the performance is improved, ***then*** the gene g' will be kept in the candidate gene set for the next round evaluation.

4. ***if*** the performance is not improved, ***then*** append a gene that has the next highest ranking score from the rest of g_ρ and create a new candidate gene set.

   (a) evaluate the classification performance of newly updated gene set.

   (b) ***if*** the performance is improved, ***then*** keep this gene in the candidate set g_i, ***else*** discard it;

   (c) repeat Step 4, ***if*** the performance cannot be improved with gene g' during the last a times (a is a specified constant value), ***then*** discard gene g'.

5. Iterate the process until all genes in the gene pool are investigated.

6.3.1 The Illustration of the Proposed iPM on Three Gene Datasets

This experiment uses the proposed iPM on three benchmark gene expression datasets, namely colon cancer, DLBCL (lymphoma) and central nervous system cancer data.

Three classification models SVM, WKNN and WWKNN are investigated for a comparative study.

The new proposed iPM method is applied on three benchmark cancer gene expression datasets: Lymphoma data (Alizadeh et al., 2000), Colon cancer data (Alon et al., 1999) and Central Nervous System (CNS) cancer data (Pomeroy et al., 2002). These gene expression datasets produced by DNA microarray technology are publicly available and widely used for cancer classification studies. All the experiments presented in this chapter are conducted using Matlab 2008 on a personal computer with Intel Core Duo 2.66GHZ CPU and 2G RAM.

Three classification models are incorporated into the proposed iPM for cancer gene expression data analysis. In this experiment, the SVM classifier is based on a polynomial kernel function and is derived from the libSVM model (Chang & Lin, 2001). The parameters used in iPM are summarised in Table 6.1.

Table 6.1: *The parameter setup for iPM experiment*

Parameter	Definition	Value
K	the number of nearest neighbours (K) in WKNN, WWKNN	15
θ	the classification threshold	0.5
r_γ	the balanced ratio between two classes	3
ρ	the pre-defined number of genes to be selected by SNR filter	200

The number of nearest neighbours is set to 15, which is based on the findings from our previous experiments of gene expression data analysis. The number of genes to be selected by a SNR filter (200) is based on our previous experiments and suggestions from literature. The selection of too few genes may result in the loss of information, while too many genes will make the learning process very time consuming. The literature on microarray research has indicated that using a few dozens to a few hundreds genes is sufficient to discriminate between different patterns in most microarray experiments (Li & Yang, 2002). Hence, the number of genes to be used for constructing a candidate gene pool is 200.

6.3.2 Case Study 1: Colon Cancer Data Analysis

This colon cancer dataset (Alon et al., 1999) consists of 62 samples of colon epithelial cells from colon cancer patients, in which 40 samples are collected from tumors and labeled as "diseased (class 2)", and 22 "normal (class 1)" labeled samples are collected from healthy part of the colons from the same patients. Each sample is represented by $2,000$ genes selected out of total $6,500$ genes based on the confidence in measured expression levels.

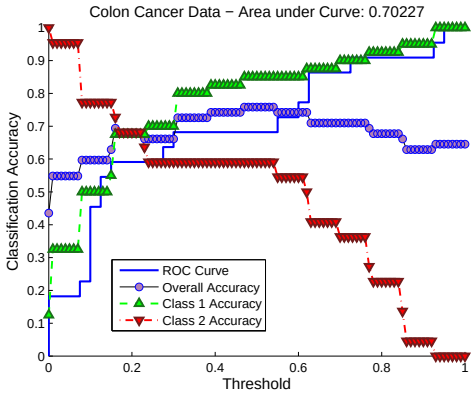
The experimental result of iPM on colon cancer data is shown in Table 6.2, for referencing, true positive (TP) , true negative (TN), false positive (FP) and false negative (FN) are also summarised in this table.

Table 6.2: *The classification results of iPM method for colon cancer data. The results are presented by the best LOOCV testing accuracy with TP, TN, FP and FN*

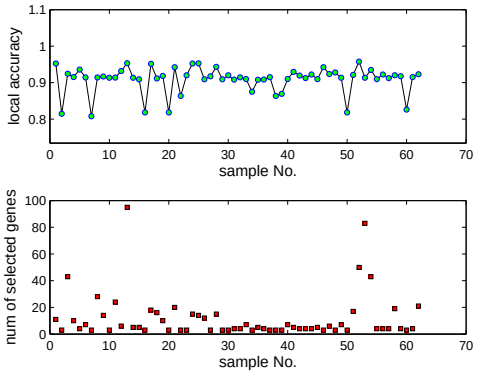
Classifier model	TP	TN	FP	FN	Classification Accuracy(%)
$WKNN$	13	34	6	9	**75.81**
$WWKNN$	9	35	5	13	70.97
SVM	9	34	6	13	69.35

Figure 6.2 illustrates the experimental results for the iPM method for colon cancer data, in which different classification algorithms are investigated and compared. In Figure 6.2 also shows that the local classification accuracy from training data is significantly higher than that from testing data. The *local accuracy* is defined as the average accuracy obtained in the training process within the personalised problem space. For example, suppose the personalised space for sample 5 ($D_{pers}(5)$) containing 13 samples, the local accuracy for this sample is the accuracy obtained from a classifier over these 13 samples during the training stage. In the case of WKNN classifier, Figure 6.2(a) and 6.2(b) show the local accuracy for most data samples in colon data is above 90%, which is significantly higher than the accuracy obtained from the LOOCV testing set (75.81%).

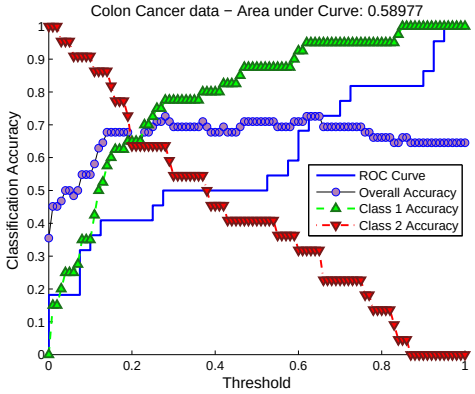
Similarly, most local accuracy obtained by WWKNN classifier from training stage is above 80%, which is clearly higher than the testing accuracy (70.97%). This inconsistent issue occurs in the experiment of SVM classifier on colon cancer data.



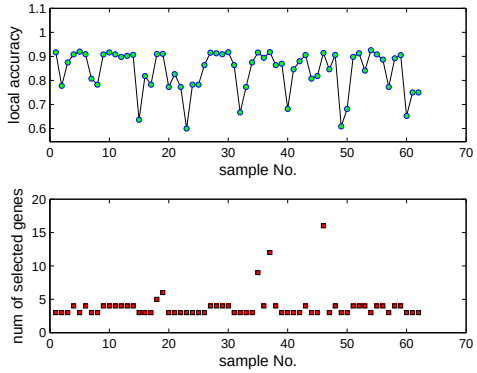
(a) WKNN classifier



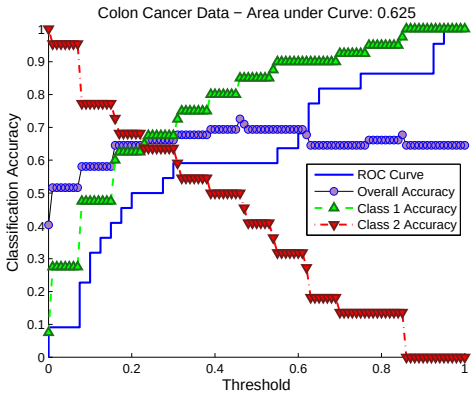
(b) WKNN classifier



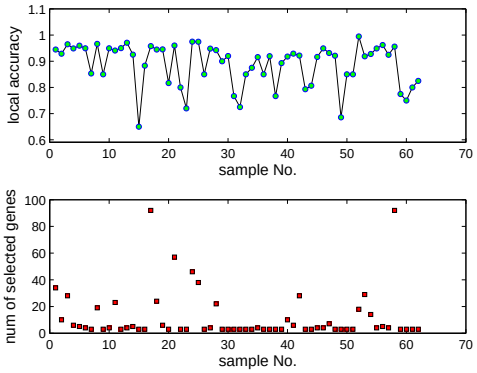
(c) WWKNN classifier



(d) WWKNN classifier



(e) SVM classifier



(f) SVM classifier

Figure 6.2: *The result of iPM on colon cancer data. Figure (a), (c) and (e) present the LOOCV accuracy using different classification threshold and ROC curve computed by the three classifiers through iPM method. Figure (b),(d),(f) plot the local accuracy obtained within the personalised problem space, and the number of selected genes for each testing sample.*

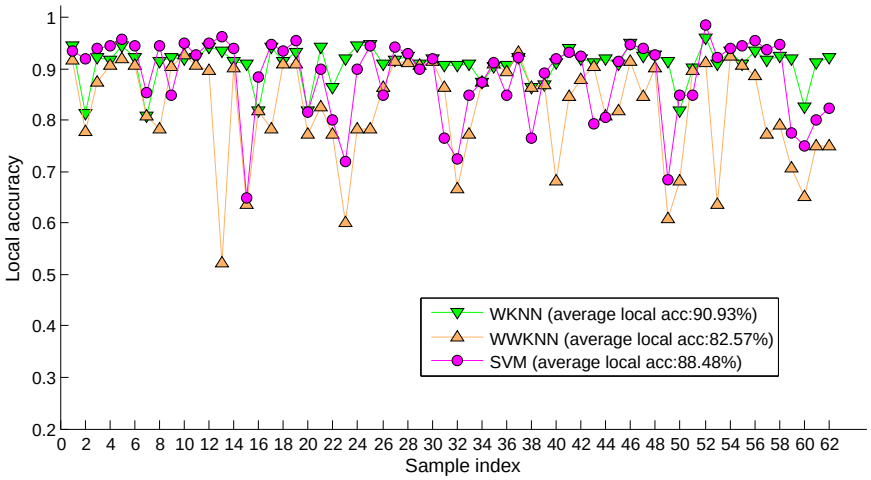


Figure 6.3: *A comparison of local accuracy from iPM method on colon cancer data using three classification models: WKNN, WWKNN and SVM*

With the personalised modelling based gene selection on colon cancer data, the number of genes selected for each testing sample is different. However, it is interesting to find that using three classifiers, including WKNN, WWKNN and SVM, the number of selected genes for each testing sample ranges from 5 to 20 (refer to Figure 6.2(b), 6.2(d) and 6.2(f)). (Note: in the case of classification accuracy measurement, x axis represents the classification threshold, y axis represents the classification accuracy; in the case of ROC curve, x axis denotes false positive rate (1-specificity), and y axis denotes true positive rate (sensitivity). The experiment results show, obviously, that several or several tens informative genes are able to give an optimum result, at least for this particular colon cancer gene expression dataset.

Figure 6.3 shows a comparison between the local accuracy obtained by iPM using the three different classifiers: WKNN, WWKNN and SVM. The results from the training stage are excellent, since the average local accuracy achieved by all three different classifiers is higher than 82%. However, the performance of iPM on testing colon cancer dataset is not very encouraging.

In Figure 6.2, the appropriate classification threshold for colon cancer classification is in the range from 0.3 to 0.5, which leads to the best cancer classification performance. The experiment also shows that each individual sample needs different number of informative genes for colon cancer disease distinction in order to achieve acceptable classification accuracy. The detailed testing report for each sample of colon cancer

data is shown in Appendix G.

6.3.3 Case Study 2: Lymphoma Data Analysis

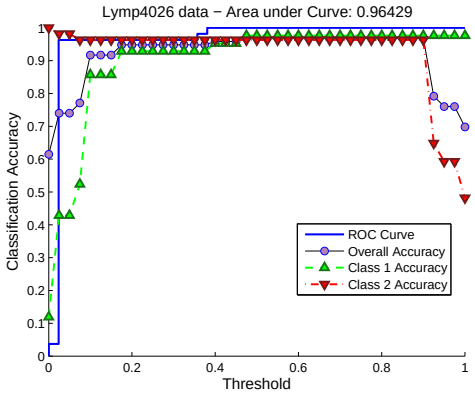
This Lymphoma dataset (Alizadeh et al., 2000) contains the expression levels of $4,026$ genes in 96 samples in lymphoma patients. Among them, 42 samples belong to Diffused large B cell lymphoma (DLBCL) group (class 1) while 54 are from other types (class 2). The objective of the study is to discriminate between DLBCL and other types of lymphoma.

Table 6.3: *The classification results of iPM method for lymphoma lymphoma data. The results are presented by the best LOOCV testing accuracy with TP, TN, FP and FN*

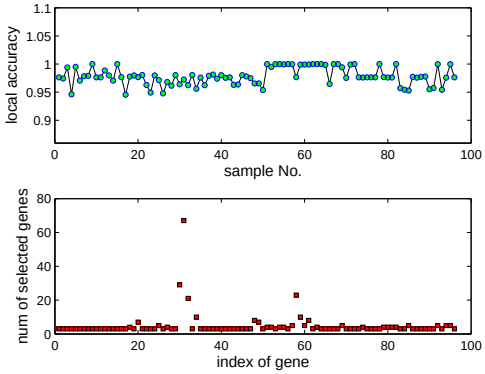
Classifier model	TP	TN	FP	FN	Classification Accuracy(%)
$WKNN$	52	41	1	2	**96.88**
$WWKNN$	52	39	3	2	94.79
SVM	52	41	1	2	**96.88**

The experimental results for the lymphoma dataset using iPM is presented in Figure 6.4. The LOOCV classification accuracy from three classifier models using iPM is summarized in Table 6.3. All three classifiers have achieved very good classification accuracy (around 95% accuracy). The accuracy for the testing and training set is consistently high, which shows the data having a good inherent consistency characteristic. Figure 6.5 shows a comparison between the local accuracy obtained from the three classifiers on lymphoma data.

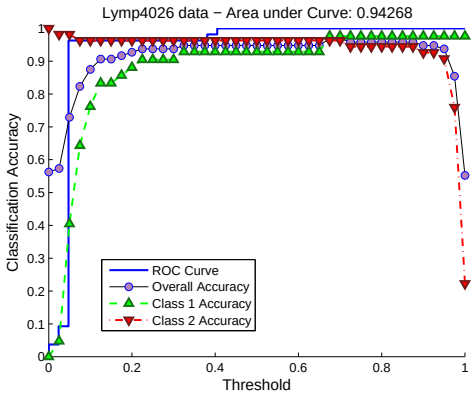
For this particular lymphoma data, WKNN and SVM slightly outperform WWKNN in terms of the classification accuracy for lymphoma distinction problem (96.88% vs. 94% accuracy). All three classification models yield satisfactory testing accuracy, mainly because of the contribution from the successful local training (i.e. all of these three classification models have very high local classification accuracy during the training process). During the training stage, all local classification accuracy is higher than 90%. In addition, in most cases only a small number of genes (mostly fewer than 10) are selected for each testing sample, and lead to a successful prediction outcome.



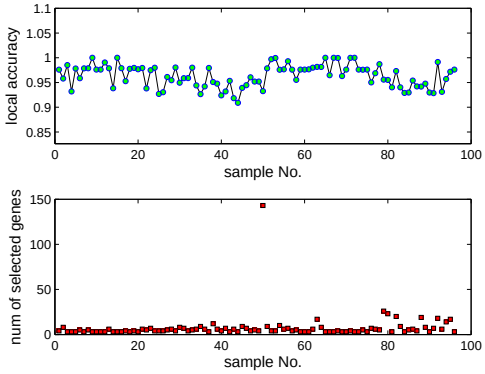
(a) WKNN classifier



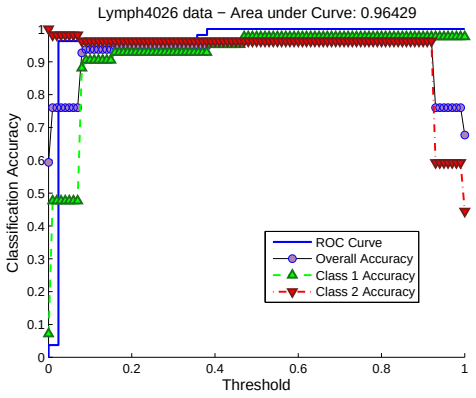
(b) WKNN classifier



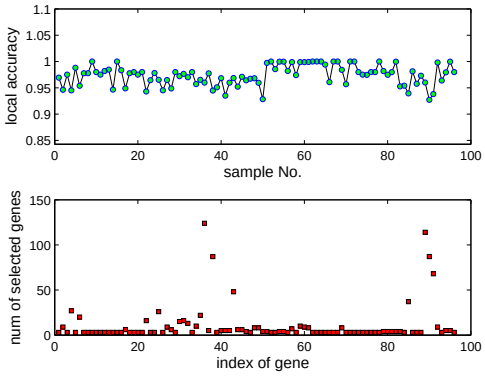
(c) WWKNN classifier



(d) WWKNN classifier



(e) SVM classifier



(f) SVM classifier

Figure 6.4: *The result of iPM on lymphoma data. Figure (a), (c) and (e) present the accuracy and ROC curve computed by the three classifiers through iPM method. Figure (b),(d),(f) plot the local accuracy obtained within the personalised problem space, and the number of selected genes for each testing sample.*

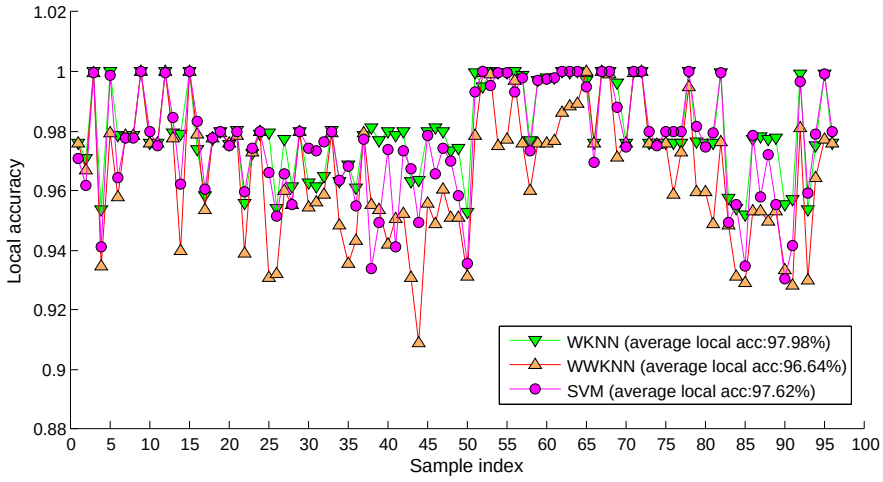


Figure 6.5: *A comparison of local accuracy from iPM method on lymphoma data using three classification models: WKNN, WWKNN and SVM*

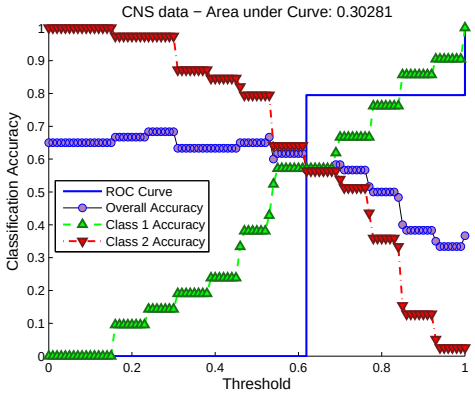
6.3.4 Case Study 3: CNS Data Analysis

CNS cancer data used in this thesis is the dataset C in Pomeroy's work (Pomeroy et al., 2002). It consists of 60 patient samples, in which 39 are medulloblastoma survivors (class 2) and 21 are treatment failures (class 1). The learning objective was to classify the patients who survived after the treatment and those who succumbed to CNS cancer. Each sample is represented by 7,129 probes from 6,817 human genes. Table 6.4 summarises the classification results of iPM on CNS cancer data. None of the classification models perform well on this data. WKNN classifier yields 66.67% accuracy, which is slightly better than the results obtained by WKNN and SVM classifiers (both of them provide 65% accuracy).

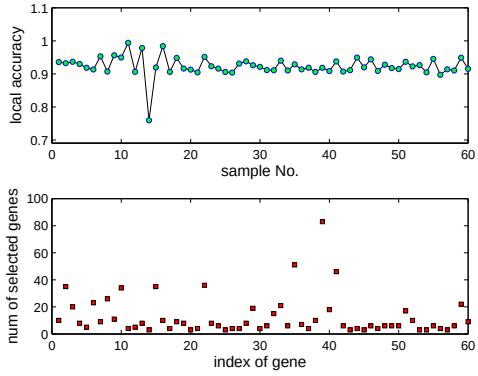
Table 6.4: *The classification results obtained using iPM for CNS cancer data*

Classifier model	TP	TN	FP	FN	Classification Accuracy(%)
$WKNN$	31	8	13	8	65.0
$WWKNN$	30	10	11	9	**66.67**
SVM	28	11	10	11	65.0

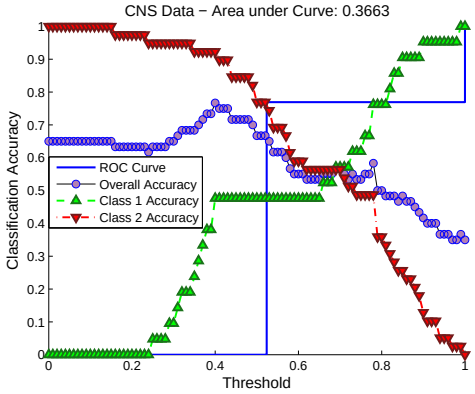
Figure 6.6 gives the ROC curves and the classification accuracy obtained by the three different classification models. Additionally, the relationship between testing accuracy and the local accuracy for CNS cancer data is also investigated and shown



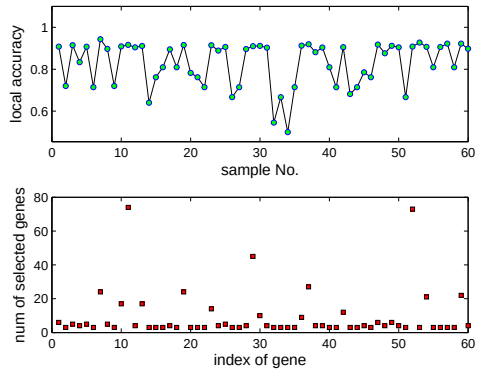
(a) WKNN classifier



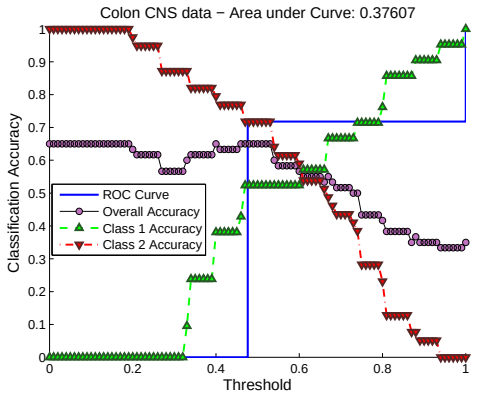
(b) WKNN classifier



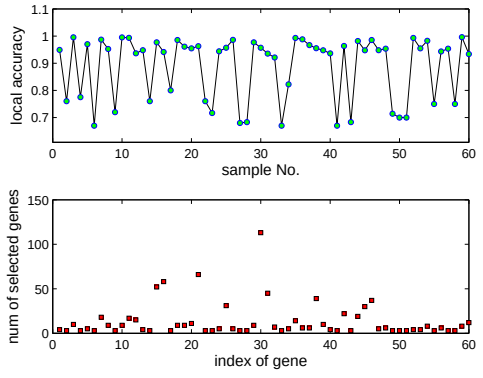
(c) WWKNN classifier



(d) WWKNN classifier



(e) SVM classifier



(f) SVM classifier

Figure 6.6: *The result of iPM on CNS data. Figure (a), (c) and (e) present the accuracy and ROC curve computed by the three classifiers through iPM method. Figure (b),(d),(f) plot the local accuracy obtained within the personalised problem space, and the number of selected genes for each testing sample.*

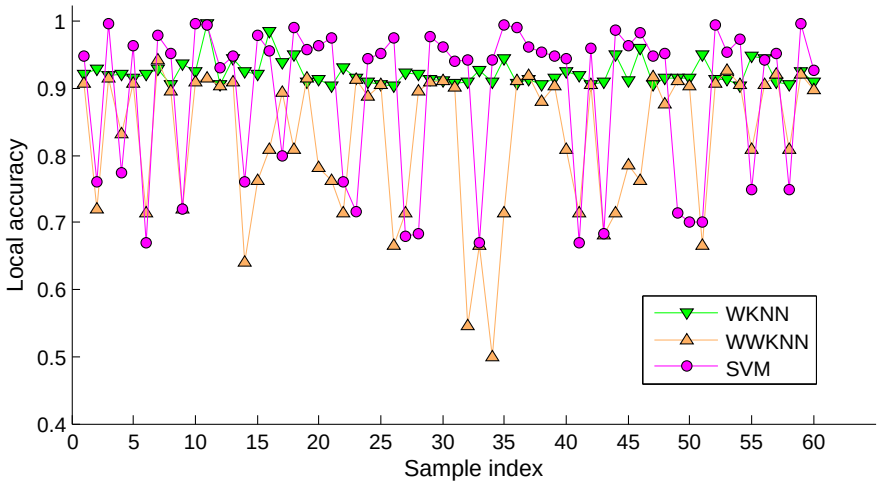


Figure 6.7: *A comparison of local accuracy from iPM method on CNS cancer data using three classification models: WKNN, WWKNN and SVM*

in Figure 6.6. For the sake of simplicity, the original ROC curve is shown in the figure, even thought its area is less than 0.5. The area under the curve calculated by each of the three classification models is less than 0.5. As we have mentioned earlier, the lowest test value can be reversed to the highest test value, so that the area under curve can be larger than 0.5. Figure 6.7 gives a comparison of local accuracy computed at the training stage by the three classifiers. The local accuracy varies significantly, which is probably the main reason that iPM does not perform well on this CNS data.

6.3.5 Discussion

The proposed iPM gives a simple approach to implement a personalised modelling system (PMS) for gene expression data analysis. It can be seen as a linear regression based approach, which mainly focuses on the statistical importance of each gene, though the quality of candidate genes is evaluated by a classifier model through an iterative learning process. However, these experiments do not show the strength of personalised modelling for gene expression data analysis, as the experimental results obtained from lymphoma, colon cancer and CNS data are not consistently good.

One interesting finding from iPM experiments is that the classification performance obtained from different classification models using iPM method is similar. This ex-

periment has investigated three classification models, namely WKNN, WWKNN and SVM for a comparison under similar experimental settings. All three classification models perform similarly on three gene expression datasets. The experiment has shown that the quality of selected genes and parameters tuning seems to be more critical to the success of analysis. Better optimised parameters (e.g. the size of neighbourhood) and more informative features (genes) contribute more than classification algorithms, in terms of improving classification performance.

This experimental study has demonstrated that the proposed iPM can extract some useful and important information from gene expression data analysis. The classification performance is not satisfying in some cases. This implementation of a PMS selects and evaluates features based on a univariate analysis in which the complex relationship among features is not sufficiently evaluated. Also, it must specify the relevant parameters of the personalised model at the very beginning of the experiment according to suggestions from literature or from experience. Moreover, there are no modules in iPM that can automatically optimise parameters. Such issues may significantly degrade the prediction performance of the personalised model M_x^* on some difficult gene expression datasets. Thus, in the next section I will introduce a new approach to implement PMS in a more effective and robust way.

6.4 Novel Methods and Algorithms for Personalised Modelling

The previous section has shown that one main difficulty in the PMS development lies in the evaluation of candidate genes during the training process. In the proposed iPM, the relationship among genes is measured to some extent, but it is not sufficiently evaluated. With iPM method, the candidate genes always include the genes with top statistical ranking scores. Therefore, whether other genes to be selected highly depends on they working together with these elite genes. However, it might be unfair because some genes do not have the chance to consolidate a candidate gene set, even though they can contribute to classification models in conjunction with other genes. For example, assume that gene#5 is ranked by a statistical model as one of the top genes. Gene#5 will be included in most cases and other genes have to work with it to form a new candidate gene set. If the performance from a candidate

gene (gene#7) with gene#5 is not good, the new inserted gene#7 will be excluded from candidate gene list even though it can be combined with other candidate genes (e.g. gene#20) to greatly benefit the given classification problem.

Such issue often results in an insufficiently trained personalised model, which produces an unsatisfactory prediction outcome. For the purpose to explore more combinations of candidate genes, a more sophisticated solution for constructing PMS is presented in the rest of this chapter, in which the search of candidate genes is driven by a model of evolutionary algorithm - genetic algorithm.

6.4.1 The Principle of PMS for Data Analysis and Knowledge Discovery

The proposed PMS creates a model specifically for every new input data sample. The method of PMS for gene expression data analysis is given as follows:

1. Use a statistical algorithm (e.g. SNR) to rank all the features in training data D, and remove the irrelevant features with very low ranking scores. The left features form to a pool of candidate features (g_ρ).
2. Create a personalised problem space (D_{pers}) specifically for the new data sample x_v through the calculation of an appropriate number of nearest neighbouring samples. The neighbourhood is calculated through an Euclidean distance based measurement.
3. Select a set of candidate features g_i from the pool g_ρ based on certain criteria, e.g. select several or several tens of top ranked features.
4. Create a candidate personalised model M_i consisting of candidate features g_i and related parameters (e.g. K_v - the number of neighbouring samples).
5. Evaluate the classification accuracy $P(g_i)$ using model M_i for each sample across the personalised space D_{pers}.
6. If the stoping criteria are **NOT** reached, update the candidate feature set g_i.
7. Iterate the process until the stoping criteria are met, output the current model as the optimal personalised model M_x^* for x_v.

8. Calculate the outcome of x_v using the optimal personalised model M_x^*.

Note: the updating of candidate features g_i in step 6 can be performed in different ways, which will be described in the following sections

In step 2, there is a concern that all the samples in the personalised problem space (D_{pers}) of x_v may fall into one class only. Under such circumstance, the constructed personalised model M_x is not sensitive to discriminate between diseased and normal samples, because the information either related to diseased pattern or normal pattern can be missing. Such issue is the *imbalanced class distribution problem* that we have discussed in Chapter 5.

To deal with this problem, a simple method is proposed to balance the sample distribution of both classes within the personalised problem space. A ratio r_γ is introduced in the construction of personalised problem space (D_{pers}) to ensure that the samples from both classes can be included. The ratio r_γ is defined in the following way:
Suppose a personalised problem space D_{pers} contains n_α samples from a majority class C_{max} and n_β from a minority class C_{min}, where $n_\alpha \leq n_\beta$. A ratio to balance the majority and minority class in the personalised problems space is calculated as:

$$n_\alpha = r_\gamma \cdot n_\beta, \quad | \; r_\gamma \in (0, 1] \tag{6.1}$$

where r_γ is a pre-specified constant value (e.g. 0.3).

If the sample distribution of D_{pers} does not satisfy Eq.6.1, D_{pers} will extend its space to include the next closet neighbour of x_v. Moreover, the ratio r_γ is not a fixed value. It is dependent to the optimal size of K_v. The neighbourhood will not extend if the number of nearest samples reaches to the maximum value.

There is a possibility that all the nearest samples in the personalised space for x_v are from one class only. In this case, the proposed PMS will produce the predicting outcome with 100% confidence, as the new sample's pattern is completely described by the samples from one class only.

An Unbiased Validation for PMS

To avoid the bias introduced by feature selection, Figure 6.8 illustrates an unbiased validation approach in the development of PMS. Within this approach, both gene selection and candidate models are only performed on the training dataset and no information from new data sample will be included during the training process.

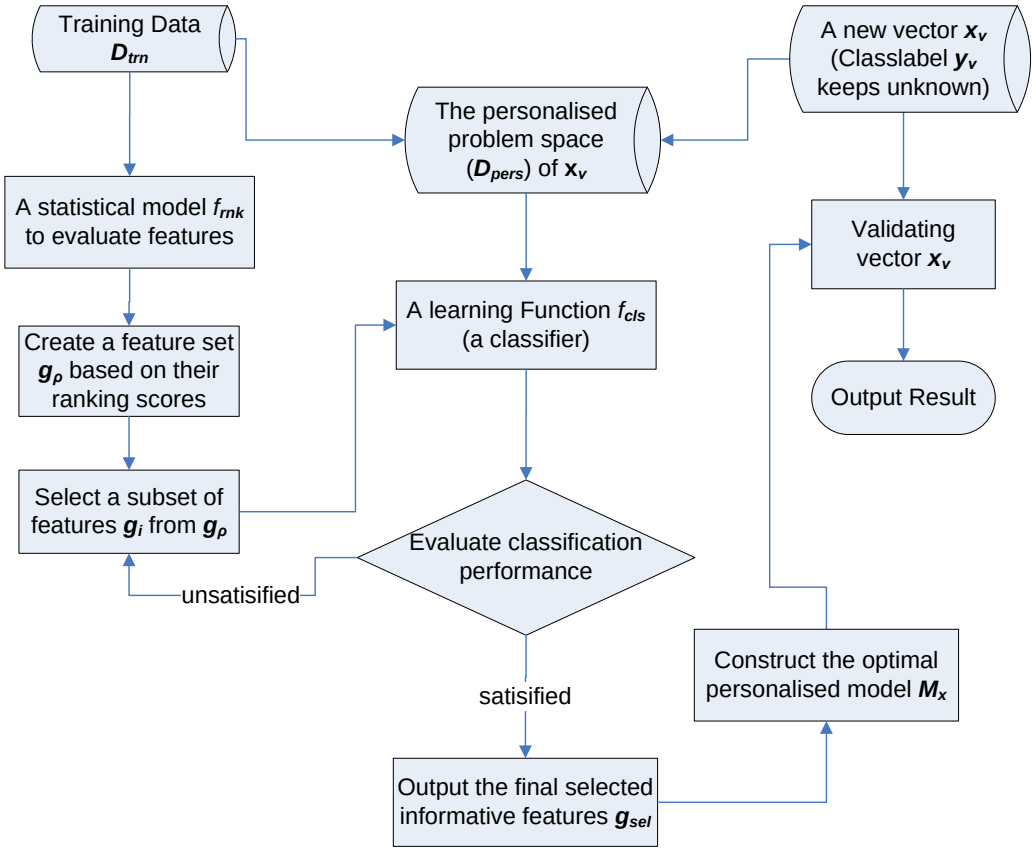


Figure 6.8: *An unbiased validation approach for PMS*

6.4.2 Evolutionary Algorithm based Approach for PMS

Evolutionary algorithm is a powerful algorithm based on generic population for solving optimisation problems. It is inspired by biological evolution, such as crossover, mutation, recombination, and selection to evolve the individuals (candidate solutions) based on the principle of "fitness to survival". Owing to its ability of driving

candidate solutions towards the target optimisation problem, evolutionary algorithm is able to explore the combination of features and related parameters, and principally able to converge to an optimal solution.

Being the most popular technique of evolutionary algorithm, GA can be applied to feature selection for model optimisation. The features can be encoded into different ways, such as binary encoding, real-value encoding, etc. Although GAs have been used in some previously developed methods for model optimisation, e.g. parameter and feature optimisation for local modelling in NeuCom (www.theneucom.com), the model and parameter optimisation for building global models (Sureka & Indukuri, 2008), GA and the other evolutionary optimisation techniques have never been used for the integrated optimisation of features, feature weights W_x, number of nearest neighbours K_v, models M_x and their parameters P_x related to personalised modelling.

The proposed general method using evolutionary algorithm based implementation for construction PMS works in the following way:

1. Select a number (K_v) of nearest neighbouring samples;
2. Iteratively select important features (V_x), and rank them through a weight vector (W_x) for the person in relation to a target problem;
3. Create a candidate personalised prognostic model M_x with the parameters (P_v) using the selected variables and nearest samples.
4. Evaluate the candidate model M_x according to its fitness computed by a learning function (a classifier);
5. Reselect features and optimise the parameters (V_x, W_x, K_v, P_x) together through an evolving way.
6. Iterate the selection and optimisation process until the termination conditions are reached.

The final optimal personalised model M_x^* may lead to the best or near best performance from the personalised prognosis.

The approach suggests a major novelty - a personalised profiling procedure in terms of defining variables that may need to be modified for the design of personal improvement scenarios afterwards, depending on the problem and the available resources.

With the optimal model M_x^*, the proposed PMS has discovered a compact set of features and relevant parameters which may bring the new insight to the given problem (complex human disease). This approach also allows for an adaptation, monitoring and improvement of the personalised model for a new input sample.

6.4.3 A Novel Gene Selection Method for Personalised Modelling

As explained in early chapters, feature selection is a fundamental step towards a successful development of PMS. In the context of biomedical data analysis, the selected features (genes) are of great importance for clinical decision support system, personalised drug design, etc. This section proposes a new feature selection method for identifying most important features for creating personalised models in PMS. For clarity, the new gene (feature) selection method is called personalised modelling based gene selection. Ideally, for the new sample x_v that is represented by a set of genes (G), the final selected set of informative genes $g_{sel}(x_v)$ and noise genes $g_{noise}(x_v)$ should satisfy the following criterion:

$$[g_{sel}(x_v), g_{noise}(x_v)] = \begin{cases} g_{sel}(x_v) \cap g_{noise}(x_v) = \Phi \\ g_{sel}(x_v) \cup g_{noise}(x_v) = G \end{cases} \tag{6.2}$$

It is impractical to evaluate the relationship of all genes in an exhaustive way when the number of genes is huge. Empirical studies have shown that most genes are redundant but only a small number of genes can benefit classification task. In literature, it is generally agreed that the good experimental results occur when several tens of genes are selected for a specific disease classification problem (Li & Yang, 2002). Hence, using univariate hypothesis tests, the proposed gene selection applies a filter method to eliminate most irreverent genes. Such method can be the classical statistical algorithms, e.g. t-test, SNR, etc.

Personalised modelling based gene selection is a hybrid approach that mainly consists of two steps:

1. Filter out the genes that are significantly irrelevant to the given scenario (e.g. disease distinction);

2. Use a wrapper method to discover informative genes from the rest of genes (a candidate gene pool).

In this thesis, personalised modelling based gene selection firstly uses SNR algorithm to rank all genes based on their univariate SNR scores, and then removes those genes having very low ranking scores. The left genes (usually several hundreds) form into a candidate gene pool to be further evaluated by a wrapper method in Step 2. SNR is used as a filter here because it is simple and fast, and outperforms another classical algorithm - T-test in terms of the classification accuracy in our experiments.

In the second step, the proposed gene selection uses a wrapper based approach to evaluate candidate genes, and employs a classifier as a learning function to evaluate the goodness of these genes within a personalised problem space. Principally, the learning function can be any classifier models. However, for a wrapper gene selection method, we need to take into account the computational cost introduced by the classifier. An appropriate classifier used in the personalised modelling method should be not only highly sensitive to the prediction results but efficient as well. Otherwise, the method may become impracticable if the classifier requires intensive computation. The pseudo code of personalised modelling based gene selection is given in Algorithm 1.

6.4.4 GA Search based PMS

This section presents an implementation for personalised modelling on gene expression data analysis using evolutionary algorithm search based approach. This approach incorporates the proposed personalised modelling based gene selection, which takes into account the interaction among genes for gene selection, and expects to have an improved classification performance and to extract more precise information and knowledge from microarray gene expression data.

To explore candidate genes, GA search based PMS takes into account the relationship among genes. The method for constructing an optimal model for a testing data vector x_v is briefly outlined in Algorithm 2.

Algorithm 2 gives a general solution using GA based search to construct a PMS for cancer gene expression data analysis. A cGA is used to replace the general GA search

Algorithm 1 Personalised Modelling based Gene Selection

Input: a new data vector x_v and a training dataset $D(n\text{-by-}m)$:

1: Normalized x_v and D
2: //Filter out the irrelevant genes
$G = f_{rnk}(D)$
3: create a candidate gene pool g_ρ from the gene set G obtained in **step 2**
4: //find the personalised problem space for x_v
$D_{pers} = f_{pers}(x_v, D), \qquad D_{pers} = \{x_i, y_i\}, i = 1, \cdots, q, q \leq n$
5: search a candidate gene set
$g_{sel} = f_{sel}(g_\rho, D_{pers})$
6: $p = f_{cls}(g_{sel}, D_{pers\_train}(x_v))$
7: **if** stopping criterion is reached **then**
8: output g_{sel}
break;
9: **else**
10: go to **Step 5** to reselect candidate genes
11: **end if**
12: //Evaluate the selected genes g_{sel} on the testing data vector x_v
$p(x_v) = f_{cls}(g_{sel}, x_v)$

where:
◇ f_{rnk}: a statistical function (e.g. SNR or T-test) for ranking all genes;
◇ ρ : a pre-specified value (usually several hundreds);
◇ f_{pers}: a function to search an appropriate personalised space for x_v;
◇ f_{sel} : a function for selecting candidate genes;
◇ p : classification performance;
◇ f_{cls} : a classification function;

part in **Step 3** in Algorithm 2. Algorithm 3 presents cGA based PMS (cGAPM) . The detailed description of cGA refers to section 3.1.7 in Chapter 3.

The main idea behind cGAPM method is that the candidate genes are selected based on a probability vector p. With the evolution of vector p driven by a cGA based algorithm, an optimal solution to construct a personalised model M_x is expected to achieve after a number of generations. Firstly, cGAPM randomly creates a probability vector p with l bits. Each bit is set to 0.5, which identifies that every bit has the equal probability to be 0 or 1. The chromosome (individual) encoding is illustrated in Figure 6.9.

Then, a probability generator function creates two individuals with the same length of bits to represent a set of candidate genes. Each bit's value is randomly created

Algorithm 2 GA search based PMS

Input: a new data vector x_v and a training dataset $D(n\text{-by-}m)$:

1: Use a statistical model (SNR) to filter out irrelevant genes.
2: Create a candidate gene pool of ρ genes from the genes selected in **Step 1**.
3: Initialize a population of μ individuals (chromosomes):
$P(gen) = f_{GA}(q, D_{pers})$
$P(gen)$ is a population created by a GA based function f_{GA}, each individual has q bits and each bit identifies a gene being selected (1) or not (0).
4: Select a population of candidate genes from $P(gen)$
$g(gen) = f_{sel}(P(gen))$
5: $p(gen) = f_{cls}(g(gen), D_{pers})$
p denotes the performance from a classifier f_{cls} using selected candidate genes $g(gen)$, D_{pers} is the personalised problem space of x_v.
6: **repeat**
7: gen ++;
8: $S(gen) = select(P_{gen-1}, \mu/2)$; //select $\mu/2$ pairs of fittest individuals.
9: $O(gen) = crossover(S(gen), \mu/2)$; //perform crossover
10: $O(gen) = mutate(S(gen)$; //perform mutation
11: $P(gen) = S(gen) + O(gen)$; //form a new generation
12: $g(gen) = select(P(gen))$
13: $p(gen) = f_{cls}(g(gen), D_{pers})$
14: **until** Stopping criterion is met
15: Output the optimal personalised model M_x with the final selected genes $g(gen)^*$

16: $p(x_v) = f_{val}(M_x, x_v)$

(ranges from 0 to 1), representing the probability whether the gene is to be selected or not. For example, if the value of bit 5 is 0.35 in Figure 6.9, the probability of this gene to be selected is 35%, i.e. this gene has a high probability (65%) to be unselected.

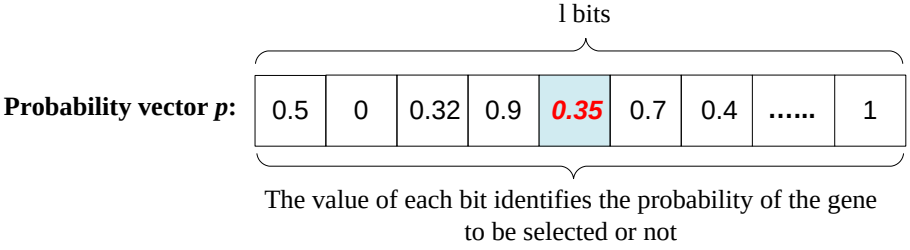


Figure 6.9: *The illustration of probability vector in cGAPM*

Algorithm 3 Compact GA (cGA) search based PM (cGAPM)

1: Use a statistical model (SNR) to filter out irrelevant genes.
2: Select ρ genes as a candidate gene pool from the genes left in **Step 1**.
3: Initialization: generate a probability vector p
$p(i) = 0.5; \quad i = 1, \ldots, l;$
4: Generate two individuals **a** and **b** based on the comparison with p:
$\mathbf{a}(gen) = \mathbf{generate}(p);$
$\mathbf{b}(gen) = \mathbf{generate}(p);$
5: Compete **a** and **b** based on their classification performance over D_{pers}:
$winner, loser = \mathbf{compete}(\mathbf{a}, \mathbf{b})$
6: Update the probability vector p towards the winner:
if $winner(i) \neq loser(i)$ then
if $winner(i) == 1$
then $p(i) = p(i) + \frac{1}{\mu}$;
else $p(i) = p(i) - \frac{1}{\mu}$;
7: Check wether any of the following terminating conditions are reached:
(1) the probability vector p has converged or,
(2) a maximum number of generations has been produced or,
(3) a highest classification performance is reached.
8: if **no** then go to **Step 2**;
9: if **yes**, then output the *optimal* personalised model M_x^* .
M_x contains the vector p that identifies which genes should be selected based on their performance from local training process, and the classifier model.
10: Validate the obtained model M_x on the testing data x_v:
$p(x_v) = f_{val}(M_x^*, x_v)$

After the first generation of two individuals are created, a classification model is applied on individual a and b, within the personalised problem space of new testing data x_v, respectively. According to the classification performance, there will be a winner between these two individuals. If the performance from individual a and b is same (i.e. no winner), cGAPM then randomly chooses one as the winner. The probability vector p is updated towards the winner to produce the next generation in the way as follows:

1. Based on the competition between individual a and b according to their contribution to the classification, cGAPM finds the winner and the loser.

2. Check whether every bit of winner and loser has the same value.

3. If they are same, no need to update this bit in the probability vector p.

4. Otherwise the i^{th} bit of probability vector p is updated by increasing $1/\mu$ if the i^{th} bit of winner is 1, or by decreasing $1/\mu$ if the i^{th} bit of winner is 0. Here, μ is a virtual population size, usually from several hundreds to several thousands depending on the length of individual.

5. If there is no winner from the competition, randomly select one individual as the winner and update the probability vector p using the same way described above.

6. Repeat the updating process, until the probability vector is converged (every bit's value is either 0 or 1), or the pre-specified stopping criterion is met, e.g. 100% classification accuracy or the maximum number of generations.

6.5 Conclusion

This chapter has introduced a PMF for data analysis and knowledge discovery. It has also presented novel methodologies and algorithms for developing PMSs. The presented methods for constructing a PMS have applications in information science, mathematical modelling, personalised medicine, profiling and prognostic systems for evaluating disease risks, using the information from a dataset in relation with the past outcomes for a given scenario.

The first method introduced for implementing a PMS is a simple method - iPM. It has been applied on three particular benchmark gene expression datasets. As an preliminary study, the experiment has shown that iPM approach is able to discover some useful information and knowledge from gene expression data. However, it does not perform effectively in some cases of difficult datasets for classification problems, mainly because it evaluates features based on univariate analysis and lacks optimisation of relevant parameters for building personalised models.

A more sophisticated methodology for implementing a GA search based PMS is proposed in this chapter. At the same time, a novel gene selection method - personalised modelling based gene selection method is developed for identifying most important genes (features) for each individual data sample, e.g. a patient's sample for cancer diagnosis and prognosis. All these algorithms and methods are genetic, and can be

used for other types of data analysis. The next chapter will apply these algorithms and methods on benchmark datasets for disease diagnosis and knowledge discovery.

For ease of reference, the presented PMS are primarily discussed in relation with bioinformatics research and its applications, such as disease diagnosis, disease risk evaluation, psychological profiling, etc. In the context of bioinformatics research, the features of the testing data may be any data from the collected samples, e.g. a person's tissue sample. All the collected samples from to a global dataset are considered to be related to a scenario of interest. Nevertheless, the presented method and system have shown the feasibility to be useful for personalised data modelling and profiling. The implementation is not limited to biomedical applications, but could be used in other data analysis areas, e.g. credit risk analysis in finance and economics.

CHAPTER 7

Personalised Modelling System for Cancer Diagnosis and Prognosis Based on Gene Expression Data

"A journey of a thousand miles begins with a single step."

- Confucius

Cancer diagnosis primarily relies on the histopathological appearances of the tumors, which has been proved unreliable and inaccurate in literature (Beart, 1995). Tumors sharing similar histopathological appearance can follow significantly different cancer courses and show different disease progressions and prognosis. The molecular heterogeneity of cancer has prevented inductive global models working efficiently on microarray gene expression data for cancer research. Contemporary cancer research demands the methodologies and systems which are able to create the useful and informative models specifically for assessing an individual cancer patient. Such circumstances motivate us to develop personalised modelling system (PMS) for cancer research using microarray gene expression data.

As an implementation of the personalised model M_x described in Figure 6.1 in Chapter 6, the proposed method here is to search for a solution to the following research problems:

1. Identify the informative features (genes) that will be used to construct personalised models for cancer classification.

2. Discover the information and knowledge from the analysis of gene expression data through personalised modelling based approaches. Such information and knowledge can be used for clinical decision system, such as risk evaluation, personalised profile visualisation, tailored personalised treatment design, etc.

The new method combines several functional modules, including a novel gene selection method, personalised space searching, outcome evaluation and personalised profile visualisation. I have applied this method to different benchmark microarray gene expression datasets, and presented the results through a comparative study in the rest of this chapter.

Many evaluation methods have been investigated for small-sample error estimation. Typically, a microarray experiment provides a dataset of small size, and as a result the most commonly used method for error estimation is *leave-one-out cross validation* (LOOCV). The LOOCV error rate estimator is often suggested in literature to be a straightforward technique for estimating generalization error in machine learning tasks and usually gives an almost unbiased performance estimation result (Breiman & Spector, 1992; Kohavi, 1995). Therefore, LOOCV classification error estimator is employed here for evaluating the performance of the proposed algorithms and models for personalised modelling.

7.1 Cancer Diagnosis and Prognosis with cGAPM using Gene Expression Data

Colon and CNS cancer gene expression datasets are used in the experiment of cGAPM for cancer classification. The validation in the experiment is followed by an unbiased validation schema illustrated in Figure 6.8, which ensures testing data

is independent to the training process. LOOCV is used for validating the quality of the optimised classifier with the final selected most important genes. Several widely used algorithms for classification problem are used to produce the gold standard for comparing the classification performance, namely MLR, MLP, SVM and ECF.

Table 7.1 summarizes the classification result for colon cancer diagnosis obtained by the proposed PMS with cGAPM algorithm. The result is reproducible and is carried out in an unbiased way. The results clearly show that the proposed cGAPM outperforms these widely used algorithms in terms of classification accuracy, if the unbiased validation approach is used. For colon cancer data analysis, Alon (1999) used 50 genes in his paper. Different number of features (20, 50 and 15) are used for global modeling algorithms in this comparison experiment.

Table 7.2 shows the classification performance of colon cancer data using a *biased* feature selection approach. Under this scenario, features are selected on the combination of training and testing data. It shows that using a biased feature selection method, statical methods can easily achieve better results than that from the models with unbiased feature selection. However, the good results cannot replaced when new coming data arrive.

Additionally, how many features should be selected for a specific data is a challenging problem for data analysis, as we don't know the outcome in advance for data predication in real world. Thus, it is arbitrary to pre-specify the number of features to be selected for data analysis.

Similarly, Table 7.3 and 7.4 give the comparison results of CNS data between cGAPM and other widely used methods in two different ways: biased and unbiased approach. The benchmark result reported in the original paper is included as well.

Again, it is easy to elucidate that the proposed cGAPM can produce better results in an unbiased way. Using a biased feature selection method, all the statistical algorithms can yield better results than the result reported in the original work.

The experiment results of colon and CNS cancer data are encouraging. The classification accuracy from colon and CNS cancer data using cGAPM method is noticeably improved compared to that from iPM method (refer to Chapter 6). The result from WKNN classifier of colcon cancer data is superior to the originally published result (refer to Table 7.1). The proposed cGAPM with WKNN classifier achieves the same

Table 7.1: *The comparison of classification results obtained by cGAPM and other widely used methods on Colon cancer gene expression data (benchmark result* refer to the result reported in the original paper). For all the models used in this experiment (except the reported results), the features are selected only based on training data. The feature selection used in original paper is on both training and testing data, which is biased. The number of selected features is based on the suggestion in literature and previous work.*

Data Set	Colon cancer data			
Method	Overall Acc(%)	Class 1/2 (%)	No. of selected Features	Validation Method
cGAPM	**87.10**	92.50 / 77.27	*automatically optimised*	LOOCV
MLR	83.87	95.00 / 63.64	20	LOOCV
MLR	72.58	75.00 / 68.18	50	LOOCV
MLR	80.65	95.00 / 54.55	15	LOOCV
MLP	80.65	87.50 / 68.18	20	LOOCV
MLP	80.65	87.50 / 68.18	50	LOOCV
MLP	75.81	80.00 / 68.18	15	LOOCV
SVM	85.48	87.50 / 81.82	20	LOOCV
SVM	85.48	87.50 / 81.82	50	LOOCV
SVM	85.48	90.00 / 77.27	15	LOOCV
ECF	82.26	87.50 / 72.73	20	LOOCV
ECF	85.48	87.50 / 81.82	50	LOOCV
ECF	79.03	87.50 / 63.64	15	LOOCV
Benchmark result*	87.0	N/A	20	holdout

Table 7.2: *The comparison of classification results obtained by different methods on Colon cancer gene expression data in a* biased *way. Features are selected based on the whole data (training + testing), which is the same approach used in the experiment in original work. The number of selected features is based on the suggestion in literature and previous work.*

Data Set	Colon cancer data			
Method (biased)	Overall Acc(%)	Class 1/2 (%)	No. of selected Features	Validation Method
SVM	88.71	90.00 / 86.36	50	LOOCV
SVM	88.71	90.00 / 86.36	20	LOOCV
ECF	87.10	90.00 / 81.82	50	LOOCV
ECF	83.87	90.00 / 72.73	20	LOOCV
Benchmark result*	87.0	N/A	20	holdout

Table 7.3: *The comparison of classification results obtained by cGAPM and other widely used methods on CNS cancer gene expression data (benchmark result* refer to the result reported in the original paper). For all the models used in this experiment (except the reported results), the features are selected only based on training data.*

Data Set	CNS data			
Method	Overall Acc(%)	Class 1/2 (%)	No. of selected Features	Validation Method
cGAPM	**78.33**	71.43 / 82.05	*automatically optimised*	LOOCV
MLR	58.33	52.38 / 61.54	100	LOOCV
MLR	56.67	42.86 / 64.10	50	LOOCV
MLR	48.33	42.86 / 51.28	20	LOOCV
MLP	65.00	23.81 / 87.18	100	LOOCV
MLP	75.00	47.62 / 89.75	50	LOOCV
MLP	45.00	28.57 / 53.85	20	LOOCV
SVM	71.67	57.14 / 79.49	100	LOOCV
SVM	73.33	57.14 / 82.05	50	LOOCV
SVM	55.00	38.10 / 64.10	20	LOOCV
ECF	73.33	42.86 / 89.74	100	LOOCV
ECF	76.67	52.83 / 89.74	50	LOOCV
ECF	55.00	47.62 / 58.97	20	LOOCV
Benchmark result*	78.33	N/A	20	holdout

Table 7.4: *The comparison of classification results obtained by widely used methods on CNS cancer gene expression data in a biased way. Features are selected based on the whole data (training + testing), which is the same approach used in the experiment in original work.*

Data Set	CNS data			
Method (biased)	Overall Acc(%)	Class 1/2 (%)	No. of selected Features	Validation Method
SVM	83.33	66.67 / 92.31	100	LOOCV
SVM	85.00	71.43 / 92.31	20	LOOCV
ECF	85.00	66.67 / 94.87	100	LOOCV
ECF	86.67	80.95 / 89.74	20	LOOCV
Benchmark result*	78.33	N/A	N/A	holdout

overall accuracy as originally published result. With WKNN classifier, cGAPM identifies a compact set of important genes that are frequently selected through LOOCV.

* For colon cancer classification, the 15 most frequently selected genes are: gene 66, 1423, 286, 897, 245, 267, 1771, 698, 187, 377, 571, 765, 415, 365, 780.

* For CNS classification, the top 10 frequently selected genes are: gene 6252, 5812, 1352, 2496, 2474, 2996, 6064, 4576, 844, 5871.

As mentioned in previous section, the focus of developing PMS method is to discover the useful information for each sample (a patient tissue sample), rather than simply to compare the classification accuracy from different algorithms. For this purpose, PMS is able to give a detailed profile for the new testing data sample. Here we give an example to demonstrate how PMS visualises the analysis result from a data sample.

Let us look at the sample 51 in colon data (sample 51 is randomly selected), cGAPM method selects *24* genes and the classifier successfully predicts that sample 51 belongs to diseased class. At the same time, cGAPM creates a personalised model specifically for colon sample 51, which comprises:

1. The personalised problem space (the neighbourhood) $D_{pers}(x_{51})$ contains 11 neighbours: sample *29, 31, 61, 57, 26, 54, 49, 6, 40, 19* and *32*;

2. A subset of informative genes: 24 genes are selected specifically for sample 51. Table 7.5 and 7.6 list the top 5 selected genes and their information. The full list of 24 genes are given in Appendix H;

3. A personalised model M_x^* is created and its local accuracy (83.82%) is evaluated on the 11 samples in $D_{pers}(x_{51})$;

4. A scenario for the improvement of sample 51 (a person) in terms of required changes in the gene expression values of each feature (gene), which is shown in Figure 7.1-b.

Table 7.5: *Top 3 genes selected for a colon cancer patient (sample 51)*

Gene Index	Gene EST Number	Gene Description (from GenBank)
377	Z50753	H.sapiens mRNA for GCAP-II/uroguanylin precursor
249	M63391	Human desmin gene, complete cds.
765	M76378	Human cysteine-rich protein (CRP) gene, exons 5 and 6.
513	M22382	Human mitochondrial matrix protein P1 (nuclear encoded) mRNA, complete cds.
286	H64489	yu67a12.s1 Weizmann Olfactory Epithelium Homo sapiens cDNA clone IMAGE:238846 3-, mRNA sequence.
...	...	...

Table 7.6: *An example: a scenario of the potential improvement for a colon cancer patient (sample 51)*

Index of Gene	Gene EST Number	Actual value	Desired average profile	Desired Improvement	Weighted importance
G377	*Z50753*	686.6330	233.8870	**-452.7460**	0.0659
G249	*M63391*	1765.1850	597.1193	**-1168.0657**	0.0625
G765	*M76378*	449.3950	260.3002	**-189.0948**	0.0555
G513	*M22382*	577.2560	1142.2057	**564.9497**	0.0533
G286	*H64489*	4474.7640	1225.8794	**-3248.8846**	0.0504
...	...	...	...	...	...

The weighted distance between the object sample and the average class profiles for each of the two classes is calculated by:

$$dst_w(x) = \sum_{i}^{l} \mid dst_{cls(i)} * \sigma_w \mid \tag{7.1}$$

where l is the number of selected features (genes), σ_w is the weighted importance of each gene (here is the SNR value), $dst_{cls(i)}$ is the distance between the testing sample's actual value and average profile of each of i class over each gene expression level, which is formulated by:

$$dst_{cls(i)} = avg(cls_{(i)}) - g_x(i), \quad i = 1, \cdots, l \tag{7.2}$$

where $avg(cls_{(i)})$ is the average profile of each of two classes, and $g_x(i)$ is the gene expression level value of gene i. The weighted distance calculated for sample 51 is as follows:

Weighted distance from class 1 profile: $dst_{cls(1)} = 470.2062$

Weighted distance from class 2 profile: $dst_{cls(2)} = \mathbf{301.9498}$

The above distance shows that sample 51 is closer to class 2 (diseased group). Also, the predicting output (1.72) for sample 51 is calculated by a WKNN classifier in the created personalised model M_x^*. Thus, sample 51 is correctly predicted as diseased (classification threshold for sample 51 is 0.4 that is determined based on the local accuracy during the training process).

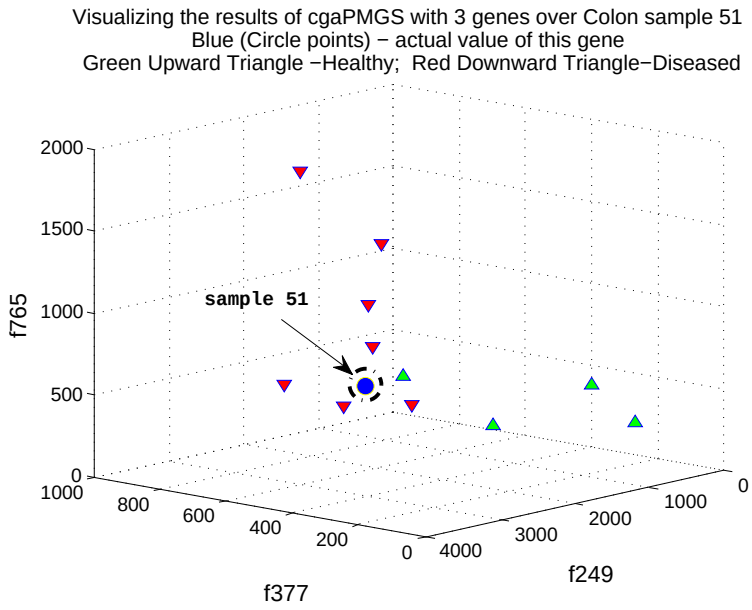
To help visualise the result, we have plotted 11 neighbours of sample 51 of colon data in a 3-D space of the top 3 genes in Figure 7.1-a. It is easy to elucidate that sample 51 is more likely to be in the diseased group, since most of its nearest neighbours belong to diseased group. Figure 7.1-b illustrates a scenario of the gene expression level improvement for a patient (here is sample 51 of colon cancer data), where x axis represents the gene index number and y axis represents the gene expression level value.

In order to recover from the disease, the patient should receive a personalised medical treatment tailored for him/her. Figure 7.1-b and Table 7.6 give an example for designing a personalised medical treatment for a colon cancer patient (data sample 51) using PM model, Table 7.6 gives an improvement scenario for a person (sample 51), which can be interpreted in the following way:
To improve the outcome from patient 51 towards a good outcome (survival), some genes need to change their expression levels through drug intervention or other means. Hence, gene 377 (EST: Z50753), 249 (EST: M63391) and 765 (EST: M76378) should be suppressed for a lower expression level, e.g. the expression level of gene 377 should be suppressed from 686.6330 to 233.8870. (Note: EST is the Expressed Sequence Tag of a gene, which is a unique index that is used retrieving genes from a NIH genetic sequence database *GenBank*).

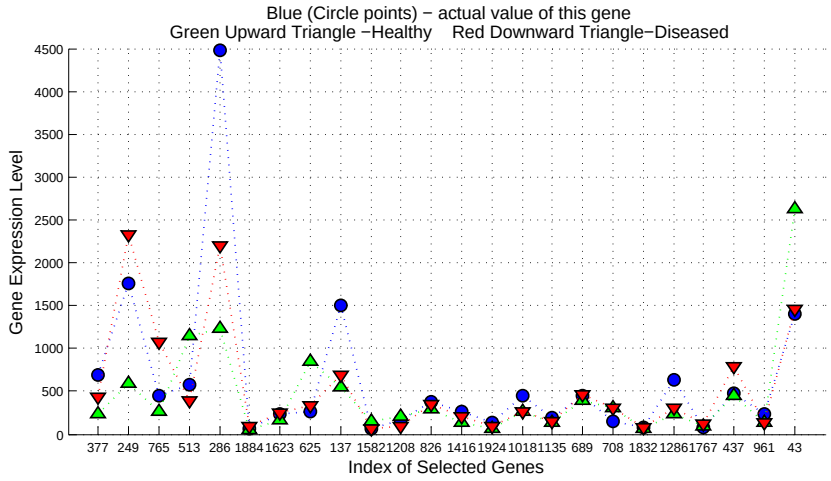
For CNS data experiment, similarly, a personalised model is created for a person (sample 31 is randomly selected), which includes:

1. The personalised problem space (the neighbourhood) $D_{pers}(x_{31})$ contains 21 neighbours: sample 48, 21, 20, 43, 26, 29, 41, 39, 8, 28, 45, 27, 30, 50, 7, 24, 13, 18, 54, 47 and 53;

2. A subset of informative genes: 23 genes are selected specifically for sample 31. The detailed list of these 23 genes are summarized in Appendix 3;

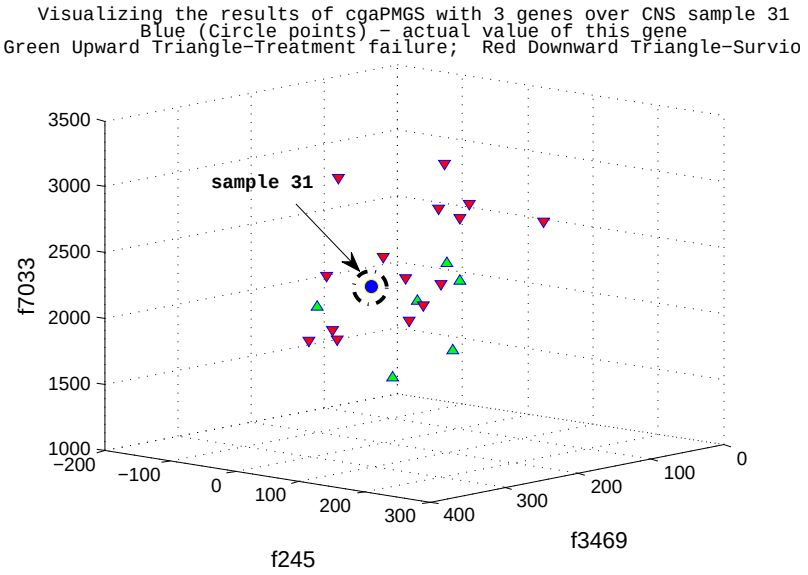


(a) A 3-D visualisation of the neighbourhood of colon sample 51 using 3 the most important genes(Gene 1772, 1325 and 1634)

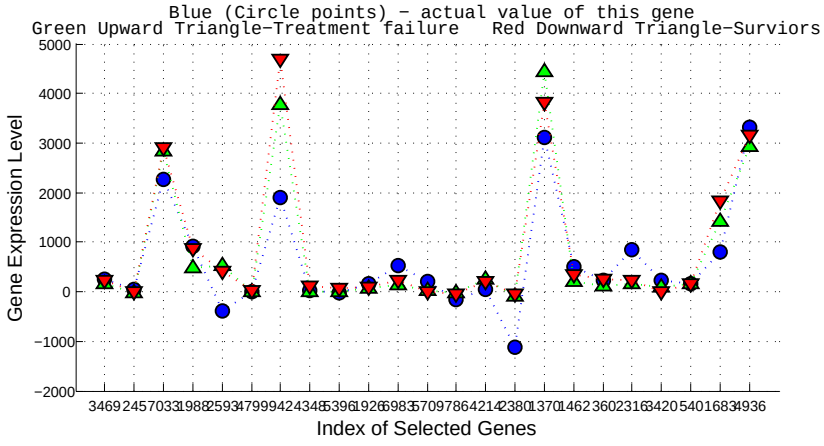


(b) A scenario of the gene expression level improvement for colon sample 51

Figure 7.1: *The profile for sample 51 of Colon cancer data*



(a) A 3-D visualisation of the neighbourhood of CNS sample 31 using 3 the most important genes(Gene 1772, 1325 and 1634)



(b) A scenario of the gene expression level improvement for CNS sample 31

Figure 7.2: *The profile for sample 31 of CNS cancer data*

3. A personalised model M_x^* is created and its local accuracy (95.84%) is evaluated on the 21 samples in $D_{pers}(x_{31})$;

4. A scenario for the improvement of sample 31 (a person) in terms of required changes in the gene expression values of each feature (gene), which is shown in Figure 7.1-b.

Figure 7.2 gives the visualisation of the experiment result over CNS sample 31.

The weighted distance calculated for sample 31 is as follows:

Weighted distance from class 1 profile: $dst_{cls(1)} = 410.9195$

Weighted distance from class 2 profile: $dst_{cls(2)} = \mathbf{405.5403}$

The weighted distance $dst_{cls(1)}$ and $dst_{cls(1)}$ is very close, which means the testing sample 31 is relatively difficult to classify in the personalised problem space. Although sample 31 is correctly predicted as diseased (class 2), a predicting risk (0.69) created by a Fuzzy KNN classifier represents the predicting result is not sufficiently confident (0.5 is the threshold for classification).

7.2 Conclusion

This chapter has presented a study to create a personalised modelling system (PMS) for cancer gene expression data analysis. The new developed personalised modelling based method offers an efficient way to construct a clinical decision support system for new coming patient samples. It has the significant potential for clinical practitioners to design tailored treatment for a patient.

The contribution of the proposed PMS is that it has introduced a new idea - selecting genes based on personalised modelling. PMS is able to discover the information from the given data and extracts a detailed profile specifically for a data sample based on the selected most informative features (genes or proteins). Such information can be used for further medical research, e.g. tailored disease treatment, personalised medicine design, drug response prediction, etc.

In addition, the research question - how to efficiently optimise the relevant parameters of personalised modelling in conjunction with feature selection, has not been solved. Some parameters in relation with personalised model construction, such as classification threshold θ and number (K) of the samples in the personalised problem space (the appropriate neighbourhood), are not sufficiently optimised. The method cGAPM does not take into account the relationship between candidate feature sets and the parameters, i.e. they are optimised separately. Such issue could be a main reason that prevents cGAPM being superior to other models in practice. This research question motivates us to develop a new method to select features and optimise related parameters simultaneously for personalised modelling.

CHAPTER 8

A Co-evolutionary Approach to Integrated Feature Selection, Neighbourhood Selection and Model Parameter Optimisation

"Imagination is more important than knowledge. Knowledge is limited. Imagination encircles the world."

- Albert Einstein

The classification of tissue samples for cancer patients is a main biomedical application in cancer research and of great importance in cancer diagnosis and potential drug discovery. However, the construction of an effective classifier involves gene selection and parameter optimisation, which poses a big challenge to bioinformatics research. This chapter presents an integrative (coevolutionary algorithm based) personalised modelling method (cEAP) for gene selection and parameter optimisation simultaneously in microarray data analysis. We apply cEAP method on four benchmark gene expression datasets to find the most important features and appropriate parameter combinations for personalised modelling.

8.1 Introduction and Motivation

In order to construct the personalised models for cancer diagnosis and prognosis using genomic data, it is critical to discover which genes (features) are most important for a specific individual patient, and find the best fit parameters for model construction. Much research effort has been put into selecting or scaling features to improve classification. A particularly popular approach is the use of evolutionary algorithms to optimise feature scaling. Another popular approach is to scale features by the mutual information of the training data with the training classes.

Another main difficulty for personalised modelling construction lies in the parameter optimisation. In the development of personalised models, the relevant parameters can be the decisive factors for: the creation of personalised problem space (the neighbourhood highly represents the pattern of new data sample), the determination of the threshold for classification and the suitable number of selected features for the specific new data. Moreover, these parameters need to be optimised along with feature selection, because their setting can be significantly influenced by different selected feature sets.

Evolutionary algorithms have been applied to a variety of research fields to search for the optimal solution in large and complex problem space. Evolutionary algorithms often have the advantage over many traditional search heuristic methods when search spaces are discontinuous, or highly constrained. However, in some cases conventional evolutionary algorithms may perform poorly. One such situation occurs when problems have very large search domains, interacting subspaces (Wiegand, 2003a). For example, this is often the case when we would like to evolve some functional elements along with their input data. The search space can be infinite in the extreme case. It is found in previous personalised modelling experiments that the optimal solution is hard to converge to use traditional evolutionary algorithms (refer to cGAPM method in Chapter 7).

To improve the performance of the personalised modelling for gene expression data analysis, the candidate solutions require different representations rather than one simple representation, i.e. the optimisation problem should be represented in different ways: the task of gene selection can be represented by binary bit flipping (either selected or not), while the solution to find the most appropriate parameters

for individual patient testing should be real-value encoded. Therefore, we need to find a better solution to the optimisation task of gene selection and parameter tuning simultaneously. Coevolutionary algorithms seem particularly desirable to solve this optimisation problem, since they are capable of dealing with a set of candidate solutions in parallel.

8.1.1 Coevolutionary Algorithm

Coevolutionary algorithms (CEAs) have attracted significant attentions as an enhancement and extension of conventional evolutionary algorithms (EAs) for solving complex computational problems. In the literature of evolutionary computation for optimisation problems, coevolutionary is defined as a change in the genetic composition of a species (or group of species) responding to a genetic change of another one (Coello, Lamont, & Veldhuizen, 2007; Potter & De Jong, 1994). A general claim of coevolutionary algorithms is an evolutionary algorithm in which the individuals from two or more populations are assigned fitness values based on their interactions with the individuals from the other populations (Wiegand, 2003b). An candidate solution is formed by a group of individuals in which every one is selected from each species.

CEAs are primarily distinguished from conventional EAs by the evaluation process in which an individual can only be evaluated by having its interaction with evolving individuals (interaction partners). These interaction partners come from the members of the same population or different populations depending on the search spaces (S. G. Ficici, 2004). In special cases, CEAs can be used for single-population evolution (Sims, 1994).

Conventional EAs are not always adequate for solving complex optimisation problems that are often in relation with problem decomposition. Consider a problem for optimising a function of m independent variables. A reasonable solution could decompose the problem into m subtasks, with each assigned to an optimisation for a single variable. In the case of personalised modelling, we do not know beforehand what is the appropriate number of the samples in the neighbourhood for a new testing data sample and which features are useful for classification. The greedy search is not a good solution for determining these factors. It seem that problem decompo-

sition consists of multiple optimisation tasks could be a more appropriate approach for solving this type of problems.

CEAs have been developed based on the premise that too few species in the problem nature may stagnate the evolution (Potter & De Jong, 2000). CEA initialises the species and evaluates its individuals in terms of the overall fitness of the given problem. It adds a new species to the problem nature if stagnation occurs. If a species can find a niche where it can benefit to the fitness evaluation, it will tend to exploit the problem nature. Within a CEA based model, species are evolved in their own populations, which can eliminate destructive cross-species mating that may make the offsprings not survive or be sterile (Smith, 1989).

Generally, a simple CEA starts with decomposing the problem space into multiple *subcomponents*. Each *subcomponent* is assigned to a *subpopulation* and then evolved by EAs. The evolution for each *subcomponent* is independent, except for the *fitness* evaluation. Since the candidate *individuals* from one *subpopulation* only represent a *subcomponent* of the problem space, the *fitness function* needs to have *collaborators* to recombine all *individuals* from different *subcomponents* for evaluation. Thus, based on the evaluated *fitness value*, a *best combined individuals* will be selected as a survivor. CEAs then proceed the selection towards the next generation and the process will be iterated until the terminating criteria are fulfilled, such as an optimal (or a near-optimal) solution is converged, or the maximum generation is reached. Coevolutionary algorithms have been implemented into a variety of artificial intelligent models for solving optimising problems, such as a neural network based coevolution model (Juille & Pollak, 1996; Potter & De Jong, 2000) and a simple GA based coevolution (S. Ficici & Pllack, 2000). They have been reported successful and efficient for finding optimal solutions to many benchmark evolutionary problems.

A basic coevolutionary model is illustrated in Figure 8.1. Although this model can be extended to multi-objective optimisation (known as *species* in literature), Figure 8.1 only demonstrates the problem with two species for simplicity. Each species (optimising task) is evolved within its own subcomponent space, .e.g., in Figure8.1 the evolution of species 1 is proceeded in its own population 1 through an application GA. The candidate individuals 1 and 2 from two species interact in a domain model and further combine into a whole individual for fitness evaluation. Based on the fitness value, the new generations will be created in both species 1 and 2 and the

process will be iterated until the terminating conditions are satisfied.

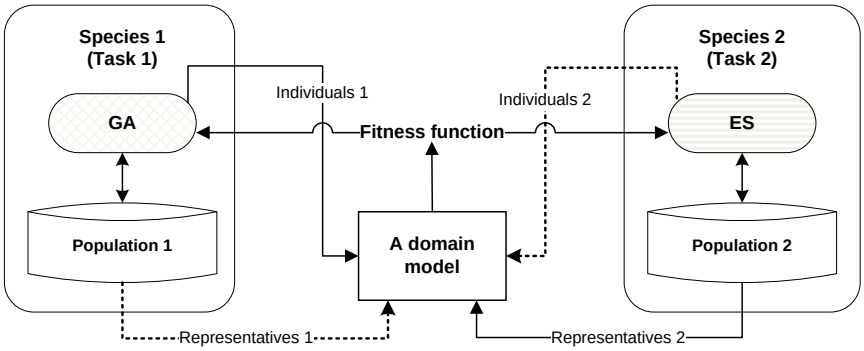


Figure 8.1: *The sample of a simple 2-species coevolutionary model. Task1 and task2 represent two subcomponent search space (*species*), respectively, the domain model can be a fitness function with existed domain knowledge. GA and ES are the evolutionary algorithms used for evolving objects in two subcomponent space, respectively*

8.1.2 Previous Work

There have been efforts using CEAs for solving complex computational problems. One of the earliest extensions to the conventional EA model for solving the optimisation in multi-components is the *classifier system* proposed by J. Holland (1986). The classifier system is a rule based system that evolves a population of stimulus-response rules through a GA. All individual rules in the population work together to consolidate a complete solution to a target problem. An algorithm called *bucket brigade* assigns the credits to the rules in a model to handle the interactions between population members. Such dynamical complexity of the model results in the problem decomposition and the preservation of diversity. Hillis (1991) has presented a method of coevolving sorting networks in which each individual of a population represented a potential sorting network. The sorting network is given a fitness score based on its contribution to an opponent data set working with the other population.

Potter and De Jong (1994) opened a door for cooperative CEAs research by introducing a general framework for cooperative CEA models. They applied the framework to static function optimisation and extened to neural network learning (Potter & De Jong, 2000). In their model, each population contains individuals representing

a component of a solution. The evolution of these populations occurred almost independently, in which interaction was performed to obtain fitness scores. Such a process could be:
(1) static, if the divisions for the separate components is decided beforehand and never altered, or:
(2) dynamically, if the populations of components may be added or removed as the learning approaches (Wiegand, 2003a).

There has been very few implementations of CEAs in bioinformatics research for solving complex optimisation problems so far. I propose a coevolutionary algorithm based personalised modelling (cEAP) for solving the challenge that involves gene selection and parameter optimisation.

8.2 Methodology

The prime goal of this chapter is to develop a new algorithm for gene selection and parameter optimisation which can be incorporated into personalised modelling systems.

8.2.1 The Proposed cEAP Algorithm

Given a general objective optimisation problem $f(x)$ to minimize (or maximize), $f(x)$ is subject to two constraints $g_i(x)$ and $h_j(x)$. A candidate solution is to minimize the objective function $f(x)$ where x represents a n-dimensional decision (or optimisation) variable vector $X = \{x_i \mid i = 1, \cdots, n\}$ from the sample space Ω. The two constraints describe the dependence between decision variables and parameters involved in the problem, and must be satisfied in order to optimise $f(x)$. The constraints $g_i(x)$ and $h_j(x)$ are denoted as inequalities and equalities respectively and mathematically formulated as:

$$g_i(x) \leq 0 \mid i = 1, \ldots, n \tag{8.1}$$

$$h_j(x) = 0 \mid j = 1, \ldots, p \tag{8.2}$$

The number of degrees of freedom is calculated by $n-p$. Note the number of equality constraints must be smaller than the number of decision variables (i.e. $p < n$). The *overconstrained* issue, occurs when $p \geq n$, because there is no degrees of freedom left for optimising objective function.

The method is to find the optimal solution to an objective function. Given an objective function $f(x)$: for $x \in \Omega, \Omega \neq \emptyset$, a global minimum of the objective problem $f(x)$ can be mathematically defined as $f^* \triangleq f(x^*) > -\infty$, ***only if***

$$\forall x \in \Omega : \quad f(x^*) \leq f(x) \tag{8.3}$$

where x^* denotes the minimum solution, Ω is the sample universe of x.

I hereby propose cEAP algorithm for selecting genes and optimising the parameters of learning functions (a classifier threshold θ and the number of neighbours k_v) simultaneously. The basic idea underlying cEAP algorithm is to coevolve the search in multiple search spaces (here is for gene selection and parameter optimisation). I employ a compact genetic algorithm(cGA) as an evolutionary model to search the subcomponent of gene selection, and use evolutionary strategy for parameter optimisation.

Regarding personalised modelling for gene expression data analysis, the whole optimisation problem space can be decomposed into three subcomponents as follows:

1. Subcomponent $\Omega_{(1)}$ for gene selection that is encoded into a binary bit string, in which each bit denotes wether this gene is selected (1) or not (0);
2. Subcomponent $\Omega_{(2)}$ for finding the appropriate number of samples K in the personalised problem space, which is real-value encoded;
3. Subcomponent $\Omega_{(3)}$ for determining the classification threshold θ to best fit individual patient sample, which is real-value encoded.

The decomposed problem space consisting of three subcomponents for gene selection and parameter optimisation is shown in Figure 8.2

The objective of this study is to build personalised models for data analysis and knowledge discovery, which are able to minimise the prediction accuracy of disease

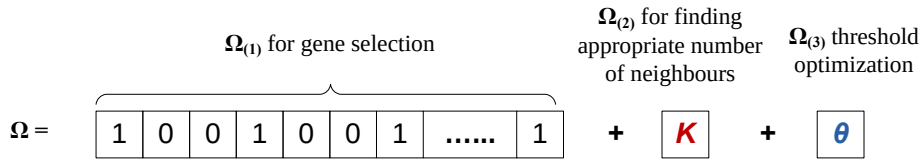


Figure 8.2: *The combined individual consisting of 3 subindividuals from subcomponent* $\Omega_{(1)}$, $\Omega_{(2)}$ *and* $\Omega_{(3)}$, *respectively.*

distinction and create a personalised profile for individual patient. Given a gene expression data $D = \{X, Y\} \mid X = x_{ij}, Y = y_i, i = 1 \ldots n, j = 1 \ldots m$, the objective is therefore defined to optimise a classifier that involves the selected genes and related parameters:

$$f(s^*) \leq f(s) \tag{8.4}$$

where f is a classification function, and s denotes an independent variables set. As s can be represented by the data vector X, Y with selected genes and related parameters, Eq.8.4 is rewritten as follows:

$$f(X, Y, \zeta_l^*) \leq f(X, Y, \zeta_l), \quad |\zeta \in \Omega, l = \{1, 2, 3\}. \tag{8.5}$$

where ζ_l denotes the candidate solution from l different subcomponents. The final solution is obtained when Eq.8.4 is fulfilled, i.e. ζ_l^* is taken as the desired solution to the problem of gene selection and parameter optimisation when the classification error is less or equal to the value at any other conditions.

The proposed cEAP method employs a compact genetic algorithm (cGA) based model for gene selection, and incorporates an evolutionary strategy to search the solution in the subcomponent of parameters optimisation. To construct a personalised model for a given dataset D pertaining to the task of cancer diagnosis and prognosis, cEAP algorithm starts with the creation of the populations of three subcomponents: gene selection in $\Omega_{(1)}$, number of samples (K) in $\Omega_{(2)}$ and the disease classification threshold (θ) in $\Omega_{(3)}$.

The population in gene selection subcomponent is generated based on a probability vector p with l bits $(l \leq n)$. Each bit in p is initialized to 0.5, representing the equal probability of this bit(gene) being selected or not. Within the subcomponent $\Omega_{(1)}$, cGA randomly creates two vectors a and b, and compares them with the probability

vector p in order to generate two bit string individuals G_a and G_b. The bit string individual is created based on the comparison result, e.g. if the value of bit i in a is larger than that of bit i in p, bit i in G_a is set to 1, otherwise 0.

Simultaneously, in the subcomponent $\Omega_{(2)}$, a probability function (e.g. a gaussian distribution function) creates a pair of individuals K_a and K_b randomly based on certain domain knowledge. Another probability function creates individuals θ_a and θ_b in the same way in subcomponent $\Omega_{(3)}$, respectively. Then, subindividuals G_a, K_a and θ_a recombines into a whole individual α that will be evaluated by a fitness function F. Similarly, another combination of subindividuals G_b, K_b and θ_b consolidates a candidate individual β.

The proposed cEAP algorithm lets individuals α and β compete to produce new generations. The evolution in gene selection subcomponent is through updating the property vector p based on the competition result. The updating scheme for p is to check each bit's value of the winner and the loser as follows:
if they are same, then there is no need to update the i^{th} bit value in vector p,
otherwise it is updated by $1/\mu$ probability of increase or decrease.
where μ is the population size.
Hence, the new generation created by the updated probability vector p will be more fitted to the fitness function F.

The basic selection scheme in cEAP for creating a new generation is:
Firstly, cEAP selects the winner from the competition of individuals α and β according to their fitness values. Then cEAP updates the probability vector p based on the comparison between the winner and loser in the gene selection subcomponent $\Omega_{(1)}$. cEAP uses the similar strategy of cGA for updating vector p:
check whether $winner(i) = loser(i), i \in [1, l]$,
if they are same, then there is no need to update the i^{th} bit of vector p, otherwise updating $p(i)$ in the following way:

if $winner(i) = 1$
  then $p(i) = p(i) + \frac{1}{N}$
else
  $p(i) = p(i) - \frac{1}{N}$
endif

where N is the population of size (a pre-defined constant value, usually several tens

or hundreds). After the updating, the probability of the alleles being either 1 or 0 in the gene selection subcomponent will increase $1/N$ in the next generation. For example, suppose individual α is the winner, if the value of $bit3$ in winner individual α is 1, then the value of $bit3$ (e.g. 0.5) in probability vector p will increase $1/N$ $(0.5 + 1/N)$. Hence, the value of $bit3$ in new offsprings will have more chance to be 1 than their parents.

At the same time, evolutionary strategy is applied to evolve the new generation in the other subcomponents - K and θ optimisation. A probability generating function is adopted to create a new pair of subindividuals for K and θ using the result from the competition between α and β: if the winner's K and θ are larger than the loser's, then their offsprings should have a higher probability to be larger than their parental pair in the loser. The existing domain knowledge can be utilised for parameters initialization, e.g., the most common value for classification threshold θ is 0.5, and parameter K can be initialized by a ratio - n/ω (ω is a weight value and n is the sample size of given data).

Once all the subcomponents have their new generations, cEAP will continue the coevolution and iterate the process until the terminating condition is reached. For clarity, Algorithm 4 gives the pseudo code of cEAP.

8.3 Cancer Gene Expression Data Classification

This case study presents a comparison experiment on four microarray cancer gene expression datasets with proposed cEAP method, SVM method and a consistency based method. SVM is generally considered as a reliable and efficient statistical method for classification. The SVM classifier used in this experiment is derived from the libSVM toolbox (Chang & Lin, 2001) developed by Chang and his colleagues in National Taiwan University. The consistency based method is our previously published model using consistency based gene selection algorithm (CAGSC) (Pang, Havukkala, Hu, & Kasabov, 2008). This method is developed based on a conventional GA, which is capable of achieving consistently good classification performance on gene expression datasets (Hu, 2008).

Algorithm 4 cEAP algorithm

1: initialize the subindividuals in the subcomponent for gene selection:
generate a probability vector p with l bits, $p_i = 0.5$, where $i \in 1, \cdots, l$,
2: generate two subindividuals from the vector p, respectively:
$(G_a, G_b) = generate(p)$;
3: generate a pair of subindividuals K_a, K_b by a probability function f_p;
4: generate a pair of subindividuals: θ_a and θ_b using a probability function f'_p;
5: recombine the above subindividuals from three subcomponents into two individuals:
$\alpha = G_a + K_a + \theta_a$;
$\beta = G_b + K_b + \theta_b$;
6: evaluate individuals α and β by a fitness function F, respectively;
7: compete individual α and β:
$winner, loser = compete(\alpha, \beta)$
8: create new populations in three subcomponents:
(i) use cGA to create the new generation for gene selection subcomponent
if $G_a(i) \neq G_b(i)$
if $winner(i) = 1$ ***then*** $p_i = p_i + \frac{1}{\mu}$
else $p_i = p_i - \frac{1}{\mu}$
(ii) use ES to create the new generation for K and θ in the other subcomponents:
Keep the winner of K and θ to form the offsprings K'_a and θ'_a; the other offsprings K'_b and θ'_b are generated through a mutation performed by probability functions f_p and f'_p.
9: check wether the termination criteria are reached:
if **yes**, ***then*** the winner individual represents the final solution ζ^*, including the selected genes G^* and optimised parameters K^* and θ^*
otherwise iterate the process from step 2.

8.3.1 Data

Four benchmark cancer gene (protein) expression datasets ares used in this study: colon cancer data (Alon et al., 1999), Leukaemia data (Golub et al., 1999), Lung cancer data (Gordon et al., 2002) and Ovarian cancer data (Petricoin et al., 2002)

8.3.2 Experiment Setup

The parameter setting is summarised as follows: the initial value of θ is 0.5 that is the most commonly used threshold for binary classification problem, and $K = n/\omega$,

where n is the sample size of the given data, and $\omega \approx 2.5$. The suggested initial value is based on our previous experimental results for personalised modelling.

Leave-one-out cross validation (LOOCV) is a widely used technique for performance evaluation on small sample size data and gives an almost unbiased validation result. The sample size in a typical microarray dataset is small, and as a result we take LOOCV classification error estimation as a straightforward approach to evaluate the performance of cEAP method for personalised modelling. For the given data(n-by-m), all samples are divided n times, where in each time all samples except one are used for training and the withheld sample (known as the left out sample) is used for testing.

8.3.3 Experiment Results

The experimental results using cEAP method over four benchmark gene datasets are applausable in terms of LOOCV classification accuracy. A comparison of classification performance from cEAP, SVM and CAGSC are summarised in Table 8.1, along with the results reported in the original study of these datasets. Figure 8.3, 8.4, 8.5 and 8.6 show the LOOCV classification results of cEAP on leukaemia, colon cancer, lung cancer and ovarian cancer data, respectively. For reference, Table J.1 summarises the results obtained by cEAP on colon cancer data through LOOCV classification in Appendix J.

This proposed method allows for the creation of an optimal personalised diagnostic and prognostic model for a new patient, which includes the prediction of outcome or risk evaluation. The method can also assist to design a tailored personal improvement scenario.

Here, I used two examples to demonstrate the profiling ability of proposed PMS. I randomly select one sample from colon cancer data and leukaemia data, respectively. They are sample#57 from colon cancer data and sample#65 from leukaemia data.

In the case of colon sample#57, cEAP selects **11** out of 2,000 genes that are most informative for colon cancer classification. Along with these selected genes, two parameters - classification threshold θ and the number of neighbouring samples (K) are optimised specifically for *sample#57*. Figure 8.7 presents a profile for colon sam-

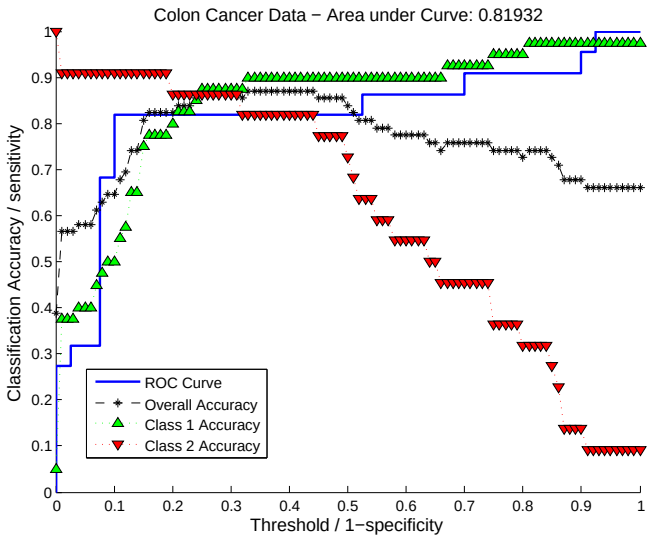


Figure 8.3: *The LOOCV classification accuracy of cEAP on colon cancer data, where in the case of classification accuracy measurement, x axis represents the classification threshold and y axis is the classification accuracy; in the case of ROC curve, x axis represents false positive rate (1-specificity), while y axis is true positive rate (sensitivity)*

Table 8.1: *The classification accuracy of different methods on all datasets. The classification accuracy of cEAP is presented by overall accuracy and class 1/class 2 accuracy*

Dataset	**cEAP**[%]	CAGSC[%]	SVM[%]	original publication[%]
Colon	87.10 (90.00/81.82)	82.26	87	87(Alon et al., 1999)
Leukaemia	100 (100/100)	95.84	93.75	85(Golub et al., 1999)
Lung	98.90 (93.55/100)	91.28	95.30	90(Gordon et al., 2002)
Ovarian	99.60 (100/99.38)	98.38	92.49	97(Petricoin et al., 2002)

ple#57, in which Fig.8.7.(a) shows the personalised modelling space (a neighbourhood with an appropriate size) of sample#57 using top 3 selected genes (gene 249, 377, 267). The neighbourhood contains 24 samples who are most close to sample#57 in terms of similarity measurement. In Fig. 8.7.(a), the personalised modelling space clearly shows that sample#57 is surrounded by the samples from diseased class (the red downward triangle points) much more than the samples from healthy class (the green upward triangle points). Thus, sample#57 is more likely to be a diseased sample based on the above observation. This assumption is afterwards proofed by

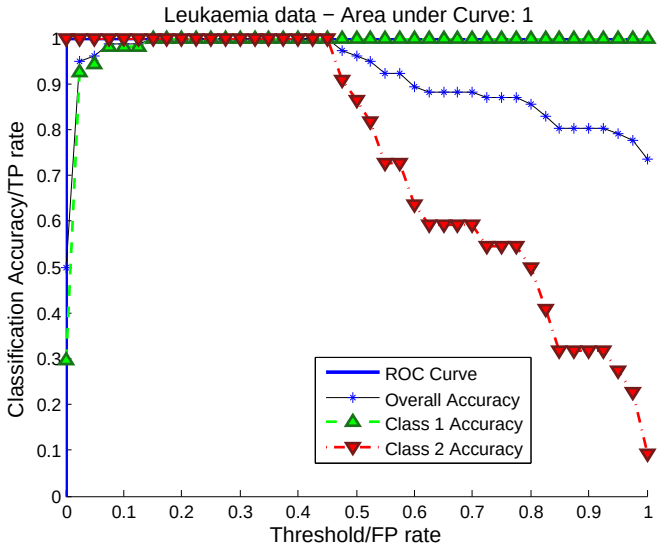


Figure 8.4: *The LOOCV classification accuracy of cEAP on leukaemia data, where in the case of classification accuracy measurement, x axis represents the classification threshold and y axis is the classification accuracy; in the case of ROC curve, x axis represents false positive rate (1-specificity), while y axis is true positive rate (sensitivity)*

the prediction result obtained using cEAP method.

A personalised model is created by cEAP method for classifying colon sample#57 as follows:

- K = 24 neighbours of sample#57;
- neighbouring samples in the personalised space of sample#57: $D_{pers}(57)$ = 51, 31, 28, 55, 8, 32, 49, 14, 47, 61, 12, 29, 54, 22, 27, 30, 59, 6, 15, 1, 38, 26, 36, 41
- The optimised classification threshold θ for sample#57 is 0.55;
- 11 genes are selected as the informative genes for sample#57 and weighted through SNR for the personalised space D_{pers}:

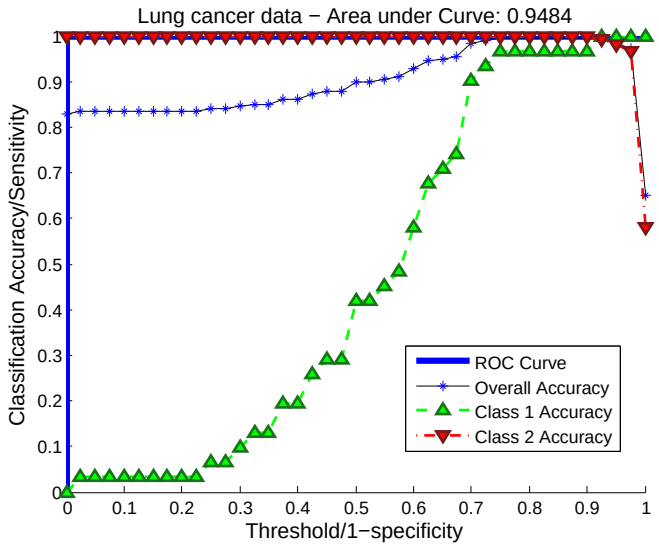


Figure 8.5: *The LOOCV classification accuracy of cEAP on lung cancer data, where in the case of classification accuracy measurement, x axis represents the classification threshold and y axis is the classification accuracy; in the case of ROC curve, x axis represents false positive rate (1-specificity), while y axis is true positive rate (sensitivity)*

Gene Index	*Weighted SNR value*	*Gene Index*	*Weighted SNR value*
*G*249	0.1241	*G*1982	0.0854
*G*377	0.1218	*G*1582	0.0797
*G*267	0.0970	*G*662	0.0745
*G*419	0.0942	*G*1870	0.0735
*G*1674	0.0914	*G*43	0.0681
*G*548	0.0903		

Table 8.2 lists these 11 genes with ***Genbank*** accession number and their biological descriptions.

- The best local accuracy calculated by a WKNN classifier in $D_{pers}(57)$ over the 24 nearest neighbouring samples is ***82.58%***.

- The predicted outcome for sample#57 is 1.65, so that it is classified as a diseases sample (the threshold is 0.55). Moreover, the outcome shows the certainty (risk probability) to determine which class this sample belongs. In this case, the interval between the predicted outcome and threshold is small (0.65-0.55=0.1), which shows an average certainty of the predicted outcome.

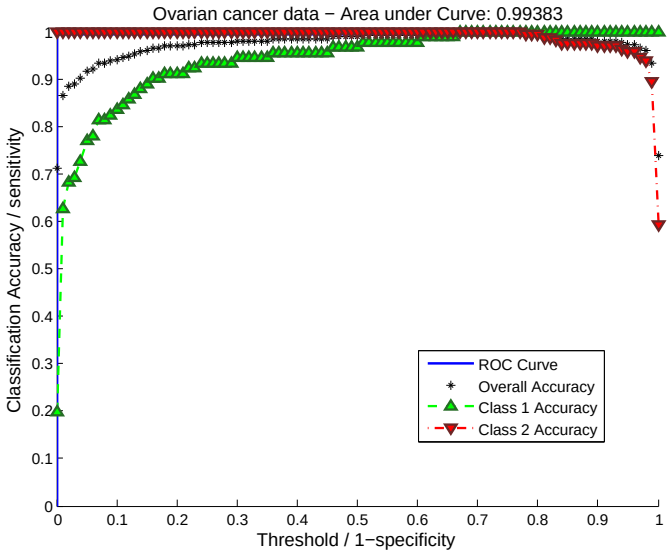


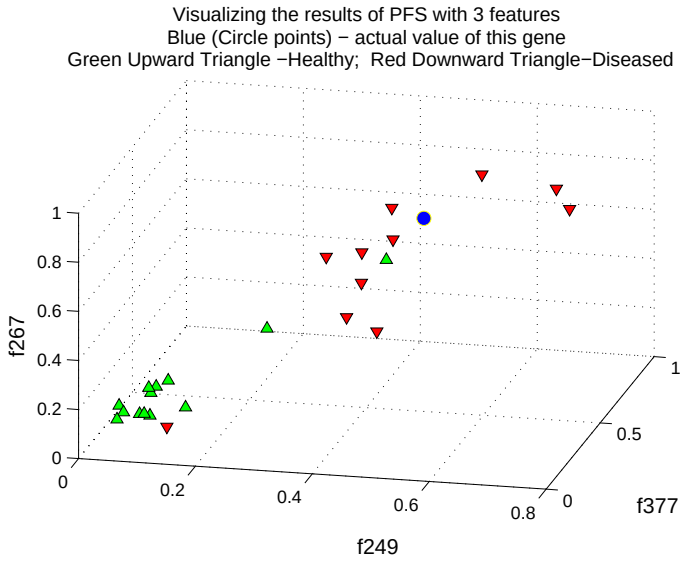
Figure 8.6: *The LOOCV classification accuracy of cEAP on ovarian cancer data, where in the case of classification accuracy measurement, x axis represents the classification threshold and y axis is the classification accuracy; in the case of ROC curve, x axis represents false positive rate (1-specificity), while y axis is true positive rate (sensitivity)*

- A profile of sample#57 is designed and shown in Table 8.3.

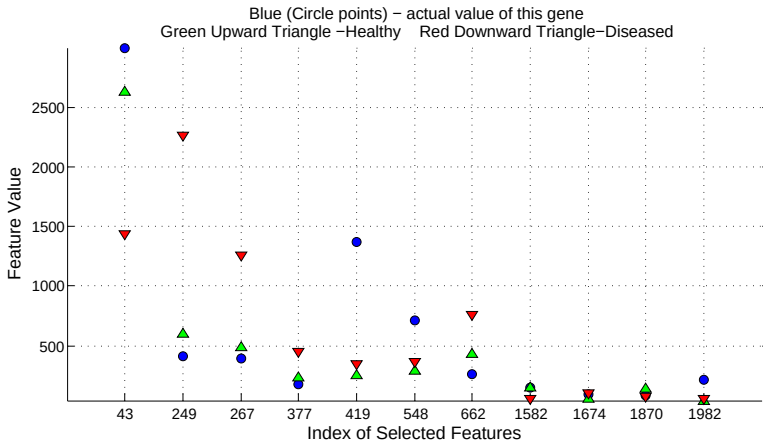
Table 8.2: *The 11 selected genes for colon sample#57*

Index of Gene	GenBank Accession Number	Description of the Gene (from GenBank)
G249	M63391	Homo sapiens desmin gene, complete cds
G377	Z50753	H.sapiens mRNA for GCAP-II/uroguanylin precursor
G267	M76378	Human cysteine-rich protein (CRP) gene, exons 5 and 6
G419	R44418	NUCLEAR PROTEIN (Epstein-barr virus)
G1674	T67077	SODIUM/POTASSIUM-TRANSPORTING ATPASE GAMMA CHAIN (Ovis aries) cds
G548	T40645	Human Wiskott-Aldrich syndrome (WAS) mRNA, complete cds.
G1982	T89666	INTERLEUKIN-6 RECEPTOR BETA CHAIN PRECURSOR (Homo sapiens)
G1582	X63629	H.sapiens mRNA for p cadherin.
G662	X68277	H.sapiens CL 100 mRNA for protein tyrosine phosphatase
G1870	H55916	PEPTIDYL-PROLYL CIS-TRANS ISOMERASE, MITOCHONDRIAL PRECURSOR(HUMAN)
G43	T57619	40S RIBOSOMAL PROTEIN S6 (Nicotiana tabacum)

Note: the detailed experimental result of cEAP on for colon cancer sample#57 is included in the Appendix K.



(a) The neighbourhood of the sample#57 of colon data



(b) A scenario of potential genome improvement for sample#57

Figure 8.7: *The personalised profile of sample#57 from colon cancer data*

In addition, cEAP has created a scenario of potential genome improvement for sample#57, which is illustrated in Table 8.3. In Table 8.3, the actual value represents the actual gene expression level of a gene from sample#57. Desired average profile is the average gene expression level from healthy samples group and desired improvement value identifies the change of the gene expression level that this patient (sample#57) should follow in order to recover from the disease. For example, the distance between M63391 gene expression level of sample#57 and the average class profile for class 1 (normal class) and class 2 (diseased class) is:

Table 8.3: *An example: a scenario of the potential improvement for colon sample#57*

Index of Gene	GenBank Accession Number	Actual value	Desired average profile	Desired Improvement	Weighted importance
G249	**M63391**	411.6240	597.1193	185.4953	0.1241
G377	**Z50753**	179.9090	233.8870	53.9780	0.1218
G267	**M76378**	397.7460	490.9205	93.1746	0.0970
G419	**R44418**	1370.3900	249.8221	-1120.5679	0.0942
G1674	**T67077**	98.2440	56.9415	-41.3025	0.0914
G548	**T40645**	717.0060	288.2512	-428.7548	0.0903
G1982	**T89666**	215.9140	43.2651	-172.6489	0.0854
G1582	**X63629**	151.1990	154.7945	3.5955	0.0797
G662	**X68277**	262.8410	428.0565	165.2155	0.0745
G1870	**H55916**	90.0480	142.6591	52.6111	0.0735
G43	**T57619**	2997.3980	2623.7725	-373.6255	0.0681

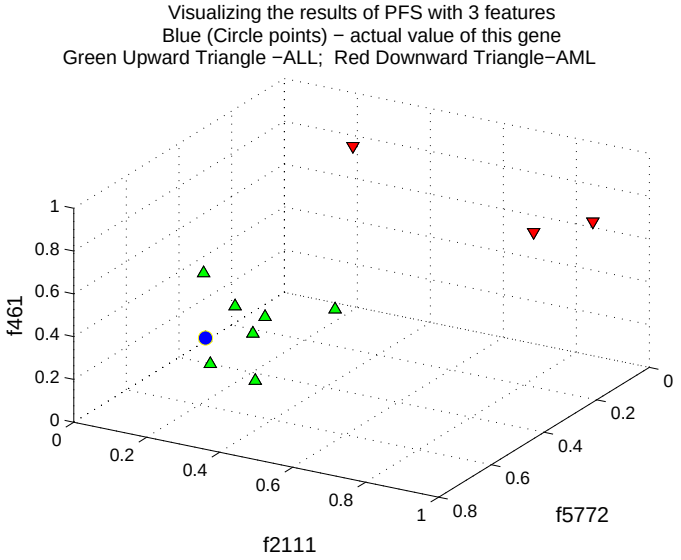
Table 8.4: *The 16 selected genes for leukaemia sample#65*

Index of Gene	GenBank Accession Number	Description of the Gene (from GenBank)
G5772	U22376	C-myb gene extracted from Human (c-myb) gene, complete primary cds, and five complete alternatively spliced cds
G2111	M62762	ATP6C Vacuolar H+ ATPase proton channel subunit
G461	D49950	Homo sapiens mRNA for interferon-gamma inducing factor(IGIF),complete cds
G2354	M92287	Homo sapiens cyclin D3 (CCND3) mRNA, complete cds
G2759	U12471	Homo sapiens thrombospondin gene, partial cds, alternatively spliced
G6974	M28170	Human cell surface protein CD19 (CD19) gene, complete cds
G2242	M80254	PEPTIDYL-PROLYL CIS-TRANS ISOMERASE, MITOCHONDRIAL PRECURSOR
G2546	S82470	BB1=malignant cell expression-enhanced gene/tumor progression-enhanced gene [human, UM-UC-9 bladder carcinoma cell line, mRNA, 1897 nt]
G3056	U32944	Human cytoplasmic dynein light chain 1 (hdlc1) mRNA, complete cds
G1829	M22960	Human protective protein mRNA, complete cds
G4951	Y07604	H.sapiens mRNA for nucleoside-diphosphate kinase
G6225	M84371	Human CD19 gene, complete cds
G1144	J05243	Human nonerythroid alpha-spectrin (SPTAN1) mRNA, complete cds
G5348	M61853	Human cytochrome P4502C18 (CYP2C18) mRNA, clone 6b
G6990	U21689	SAT Spermidine/spermine N1-acetyltransferase
G6847	M13485	Human metallothionein I-B gene, exon 3

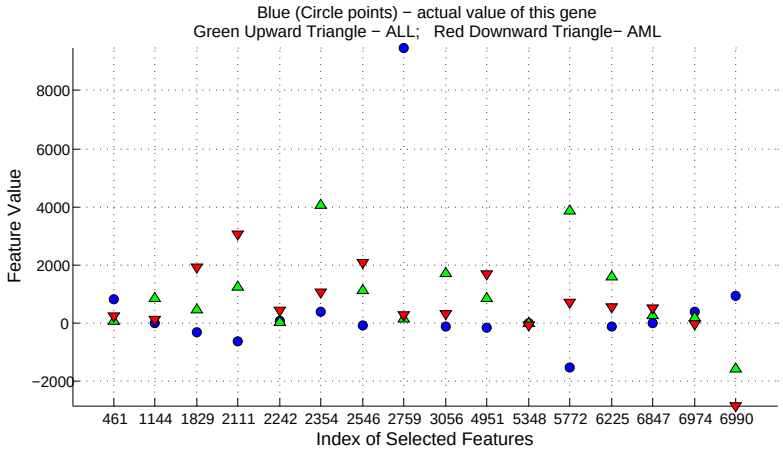
185.4953 (for class 1)

1851.8648 (for class 2)

i.e. if patient (sample#57) wants to be recovered from colon cancer, a potential solution can be given for increasing his/her M63391 gene expression level from 411.6240 to 597.1193. Table 8.3 also summarizes the importance of each selected genes in terms of the contribution to disease prediction. Larger the importance value, more



(a) The neighbourhood of sample#65 of Leukaemia data



(b) A scenario of two types of leukaemia in terms of genome difference for sample#65

Figure 8.8: *The personalised profile of sample#65 from leukaemia data*

informative the gene. The information concluded by this improvement scenario can be used for designing personalised treatment for cancer patient.

It is interesting to find that gene *249 (M63391)* and *377 (Z50753)* are selected as top 2 genes by cEAP and cGAPM (ref Chapter 7). It may conclude that these two genes are highly contributive to colon cancer diagnosis.

Similarly, this study presents the experimental result from Leukaemia data using

cEAP method. Table 8.4 summarizes the selected **16** informative genes with their GenBank accession number and biological descriptions.

The experimental findings above we have discussed for colon sample#57 is mainly from the computational prospective. Since an important objective of this study is to identify some potential marker genes for cancer classification, I have compared the selected genes by cEAP with those reported in Golub's famous work (Golub et al., 1999). Golub and his colleagues selected 50 genes (see fig.3 in their paper (Golub et al., 1999)) for building classification model. Among Golub's 50 top genes, four genes (gene U22376, M62762, M92287, U32944) are also selected out by cEAP method. Gene U22376 is consistently identified as the most informative one for disease classification by both methods. This gene can be considered as a biomarker genes for distinguishing leukaemia types.

8.4 Gene Marker Discovery

The proposed cEAP has been so far applied on four genomic datasets for cancer classification. The prediction accuracy has been improved compared with previously published benchmark results. In order to find a smaller number of genes, as global markers that can be applied to the whole population of the given problem, all genes selected for every sample in the dataset are ranked based on their likelihood to be used for all samples. The top l genes (most frequently used for every individual models) are selected as a set of potential markers for cancer diagnosis across the whole population.

The approach used here for selecting potential marker genes is as follows:

1. Calculate the frequency of the features selected by cEAP on the given data (refer to section 8.3.3);
2. Use the most frequently selected l features as the marker genes (G_{mk}), which is a global selection based on PM;
3. Apply LOOCV on the data with the marker genes (G_{mk}) for classification;
4. Use different number of neighbours (K) for evaluating the performance of cancer classification.

In this experiment, colon cancer gene expression data is used for demonstrating the above approach. Based on the result obtained by cEAP in previous section, the frequency of genes selected for each sample in colon cancer data has been computed. As Alon reported in their study that 20 genes selected by t-test could lead to good result (Alon et al., 1999), I selected the same number of genes according to the selecting frequency obtained using cEAP. Table 8.5 lists these 20 selected genes. Figure 8.9 shows the frequency of the 20 genes selected by cEAP across the global problem space - colon cancer data.

Table 8.5: *The 20 most frequently selected genes (potential marker genes) for colon cancer gene data*

Index of Gene	GenBank Accession Number	Description of the Gene (from GenBank)
G377	Z50753	H.sapiens mRNA for GCAP-II/uroguanylin precursor
G1058	M80815	H.sapiens a-L-fucosidase gene, exon 7 and 8, and complete cds.
G1423	J02854	Myosin regulatory light chain 2, smooth muscle ISOFORM (HUMAN)
G66	T71025	Human (HUMAN)
G493	R87126	Myosin heavy chain, nonuscle (Gallus gallus)
G1042	R36977	P03001 Transcription factor IIIA
G1772	H08393	COLLAGEN ALPHA 2(XI) CHAIN (Homo sapiens)
G765	M76378	Human cysteine-rich protein (CRP) gene, exons 5 and 6.
G399	U30825	Human splicing factor SRp30c mRNA, complete cds.
G1325	T47377	S-100P PROTEIN (HUMAN).
G1870	H55916	PEPTIDYL-PROLYL CIS-TRANS ISOMERASE, MITOCHONDRIAL PRECURSOR (HUMAN)
G245	M76378	Human cysteine-rich protein (CRP) gene, exons 5 and 6.
G286	H64489	Leukocyte Antigen CD37 (Homo sapiens)
G419	R44418	Nuclear protein (Epstein-barr virus)
G1060	U09564	Human serine kinase mRNA, complete cds.
G187	T51023	Heat shock protein HSP 90-BETA (HUMAN)
G1924	H64807	Placental folate transporter (Homo sapiens)
G391	D31885	Human mRNA (KIAA0069) for ORF (novel proetin), partial cds.
G1582	X63629	H.sapiens mRNA for p cadherin.
G548	T40645	Human Wiskott-Aldrich syndrome (WAS) mRNA, complete cds.

The objective of this experiment is to investigate whether utilising these 20 potential marker genes can lead to improved colon cancer classification accuracy. Thus, four classification models are used for comparison, including WKNN, MLR, SVM and transductive neuro fuzzy inference system with weighted data normalisation for personalised modelling (TWNFI) (Song & Kasabov, 2006). Personalised MLR and SVM are used as the golden standard in this comparison experiment.

TWNFI is a dynamic neuro-fuzzy inference system in which a local model is created

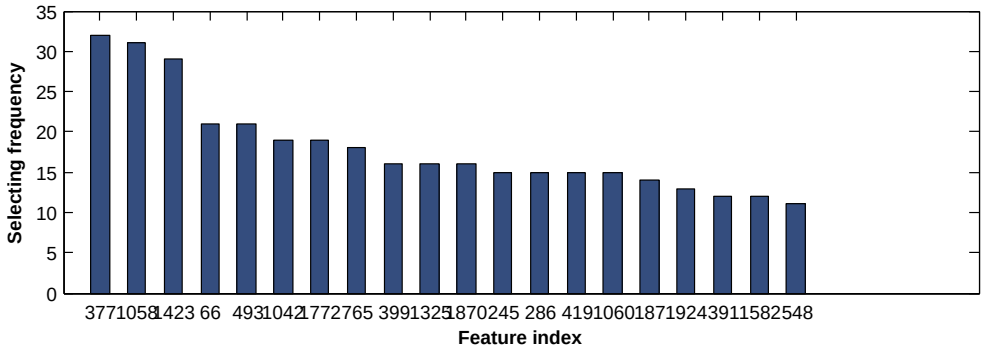


Figure 8.9: *The 20 most frequently selected genes by cEAP across colon cancer data, where x axis represents the index of genes in the data, y axis is the selected frequency of a gene.*

for analysing each new data vector x_v. TWNFI introduces a local generalisation approach, in which the Zadeh-Mamdani type fuzzy inference engine (Zadeh, 1988) is applied. The local generalisation creates a model in a sub-space (local area) of the whole problem space. This created model performs generalisation in this specific local area. In the TWNFI model, Gaussian fuzzy membership functions are used in each fuzzy rule for both antecedent and consequent parts. A steepest descent (back-propagation) learning algorithm is applied for optimising the parameters of the fuzzy membership functions (Song & Kasabov, 2006).

TWNFI usually performs a better local generalisation over new data. Comparing with weighted distance nearest neighbour algorithms, TWNFI creates an individual model for each data vector and takes into account the location of the new input vector in the space. In this sense, TWNFI is an adaptive model in which the input-output pairs of data vectors can be added to the dataset continuously and available for transductive inference of local models. The detailed learning algorithm of TWNFI is described in Appendix F.

These PM based algorithms are applied on colon cancer data with 20 potential maker genes for cancer classification. In this experiment, MLR and SVM are implemented for personalised modelling. They are called personalised MLR and SVM, because they create a unique neighbourhood for each testing sample. All the algorithms are validated based on LOOCV across the whole dataset. Since the main objective is to

validate the importance of 20 selected genes for cancer classification from a global viewpoint, each testing sample has a fixed neighbourhood size. However, each sample has its own neighbourhood (different neighbours). The experiment also evaluates the results obtained using different number of neighbours. Figure 8.10 shows the results obtained using four algorithms with different size of neighbourhood. Table 8.6 summarises the classification results obtained using four personalised algorithms using 20 potential marker genes selected by cEAP.

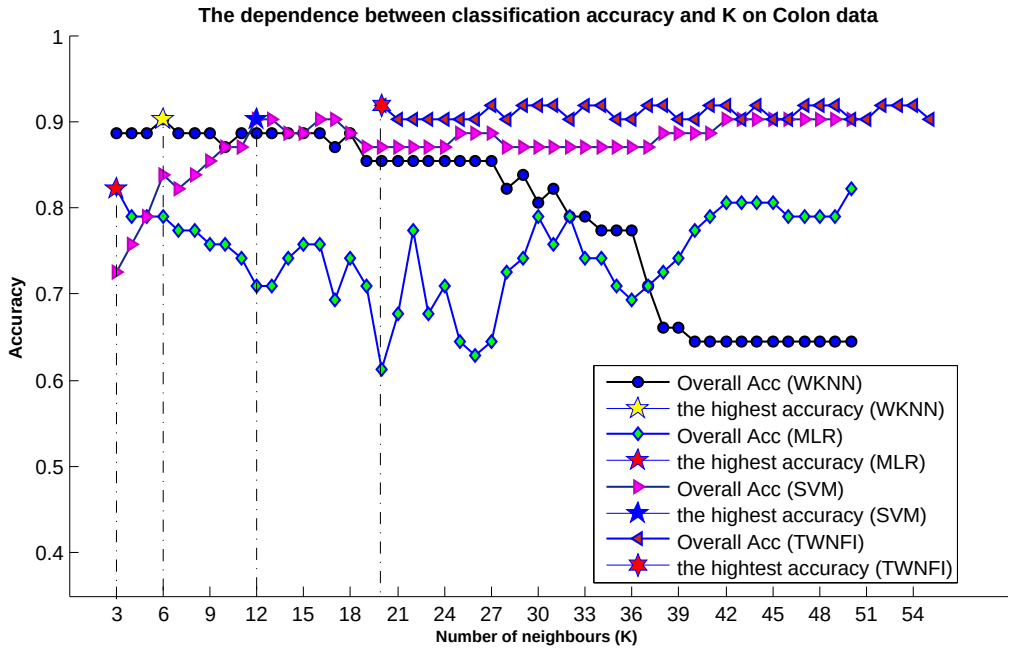
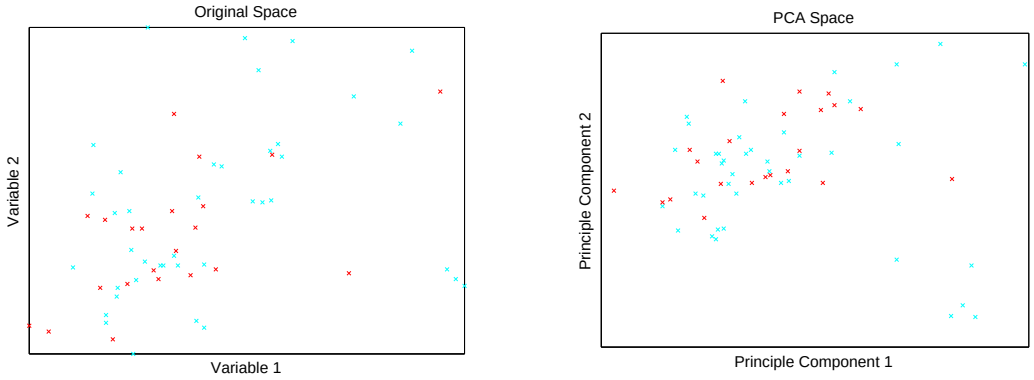


Figure 8.10: *The comparison of classification results obtained by 4 classification algorithms employed for PM, using 20 potential maker genes, where x axis represents the size of neighbourhood, y axis is the classification accuracy, k is the number of nearest neighbours.*

Figure 8.11 and 8.12 give the visualisation for the colon cancer data in different condidtions. The former figure demonstrates the data with all features (genes) in the original space and in a PCA space, while the latter shows the data with 20 marker genes in the original space and in a PCA space. Different colours of the data points represent the different classes. Using PM selected marker genes, the samples are clearly separated in the PCA space. It is clear that personalised modelling is able to identify important features, which can lead to better classification performance.

Table 8.6: *The best classification accuracy obtained by four algorithms on colon cancer data with 20 potential maker genes. Overall - overall accuracy; Class 1 - class 1 accuracy; Class 2 - class 2 accuracy;*

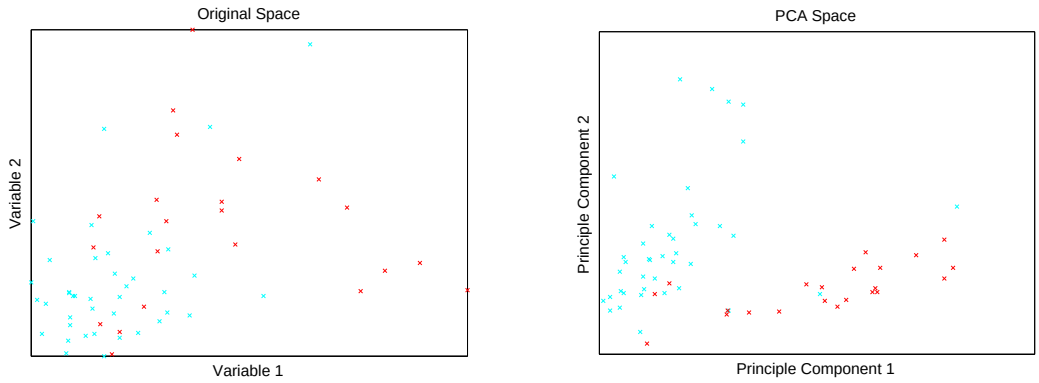
Classifier	**Overal**[%]	Class 1[%]	Class 2[%]	Neighbourhood size
MLR (Personalised)	82.3	90.0	68.2	3
SVM (Personalised)	90.3	95.0	81.8	12
WKNN (Personalised)	90.3	95.0	81.8	6
TWNFI (Personalised)	**91.9**	95.0	**85.4**	20
Original publication (Alon et al., 1999)	87.1	-	-	-



(a) The original space; (b) The PCA space

Figure 8.11: *The visualisation of colon cancer data with all genes, whereas in (a), all samples are plotted by first two variables (genes) in the original space, while in (b), all samples are plotted by two PCA variables in a PCA space.*

The experiment results illustrate that the 20 potential marker genes selected by personalised modelling system (cEAP) can lead to improved classification accuracy. These potential marker genes might be very helpful for diagnosing colon cancer through a global way, which shows the potential for drug and treatment design. Also, this experiment depicts that personalised modelling based algorithms are able to produce improved results for colon cancer classification with the globally selected features. Personalised SVM and WKNN have yielded the same classification accuracy. As a more sophisticated PM classifier, TWNFI has produced the best result (91.9% accuracy) in this experiment, which significantly improves the classification



(a) The original space; (b) The PCA space

Figure 8.12: *The visualisation of colon cancer data with 20 selected marker genes, whereas in (a), all samples are plotted by first two variables (genes) in the original space, while in (b), all samples are plotted by two PCA variables in a PCA space.*

accuracy comparing with the benchmark result (87.1%) reported in Alon's work (Alon et al., 1999).

8.5 Conclusion

In this study, we have presented a new integrative method (cEAP) using the concept of coevolutionary algorithm for gene selection and parameter optimisation for gene expression data analysis. Along with the proposed PMS, I have applied cEAP method on four benchmark cancer gene and protein expression datasets and compared the experimental results obtained by cEAP with other reported results in literature. Compared with the other three methods in Table 8.1, cEAP consistently produces better classification performance. More importantly, cEAP creates the personalised models, including selected genes and optimal disease classification parameters specifically for the observed patient sample, which are helpful to construct the clinical decision support systems for cancer diagnosis and prognosis.

To validate cEAP method from biology perspective, I have compared the selected genes by cEAP method with the biomarker genes reported in Golub's work(Golub et al., 1999). To distinguish between acute myeloid leukemia (AML) and acute

lymphoblastic leukemia (ALL), there are 16 genes most commonly selected by cEAP for each sample. These selected genes have shown the agreement with reported biomarker genes: 4 of the 16 genes - U22376, M62762,M92287, U32944 are identified informative in both cEAP and Golub's method. The difference can be accounted by the fact that I have used personalised modelling for testing each patient sample while Golub and his colleagues apply a global modelling approach for gene selection.

Another interesting finding is that gene U22376 is consistently identified as the most informative for disease classification by both methods. Additionally, this study also concludes that the selected genes for each sample in the same cancer data are not identical, i.e., the importance of genes for each cancer patient could be varied significantly, even though the genes are known to discriminate between diseased and normal samples.

In the case of colon cancer data analysis, the top 3 selected informative genes for colon sample#57 by cEAP are also marked as top genes by cGAPM. Thus, we may conclude that these three genes are more likely to be the cancer genes for diagnosing colon cancer.

The experimental results have shown that cEAP can be a good solution to complex optimisation problems, which allows to build a personalised model for different types of applications. Applications may involve a variety of modelling systems in the areas of medicine, ecology, business intelligence, finance, nutrigenomics, etc.

In the discussion section, a comparison experiment is given to demonstrate the effectiveness of selected potential marker genes for colon cancer diagnosis. The experiment results have shown that PM based classifiers can effectively work with these globally selected genes (based on their selecting frequency) for cancer classification. Such type of genes (potential maker genes) can be very useful for drug and treatment design.

The limitation here is that the optimal personalised model is not created from a global optimisation. To find an optimal solution with GA, each personalised model should be created from a global optimisation, i.e. a final created personalised model should be carried out some runs to ensure the optimal solution is not randomly reached. However, due to time and resource limitations, global optimisation for each model seems impractical for our experiment. GA based optimisation algorithms

are generally thought as the least efficient optimisation algorithms (Bhattacharyya et al., 2009; Solomatine, 1998), even though they may achieve the best solution for the target problems. In the proposed cEAP method, the personalised model is built through generations in one run. Instead of applying global optimisation, the frequency of each feature selected as informative ones has been summarized for further investigation.

Moreover, there is another open question that needs to be answered in personalised modelling: *whether different parameters of the learning function (e.g. a classifier) will significantly affect the performance of created personalised model?* The next chapter will discuss this problem and present a case study for a real world problem - Crohn's disease risk prediction using SNPs data.

CHAPTER 9

A Personalised Modelling Method and System for Disease Risk Evaluation Based on SNPs Data

"We can only see a short distance ahead, but we can see plenty there that needs to be done."

- Alan Turing

This chapter presents a case study for Crohn's disease classification using the proposed personalised modelling system - cEAP. The main goal of this case study is to design a preliminary experiment for the research project of predicting Crohn's disease using single nucleotide polymorphisms (SNPs) data. The classification problem investigated here, is a real world problem, which makes the use of SNPs data for predicting Crohn's disease risk. It is expected to elicit more information and knowledge through the analysis over selected features (SNPs in this case study). Also, I will focus on investigating the feasibility whether personalised modelling (PM) can work properly on real world biomedical data. This study will demonstrate how PM method improves the prediction outcome using different approaches, starting from the approach of using simple parameter optimisation to the approach of employing cEAP method for parameter optimisation and feature selection.

9.1 Background and Motivation

Being able to accurately predict an individual's disease risk or drug response and using such information for personalised treatment is a major goal of clinical medicine in the 21st century (Jorgensen, 2008). For many common conditions a patient's health outcome is influenced by the complex interplay of genetic, clinical and environmental factors (Nevins et al., 2003). With the advancement of microarray technologies collecting personalised genetic data on a genome-wide (or genomic) scale has become quicker and cheaper (McCarthy & Hirschhorn, 2008; Hindorff et al., 2009). Such personalised genomic data may include: DNA sequence data (e.g. Single Nucleotide Polymorphisms (SNPs), gene and protein expression data. Many world-wide projects have already collected and published a vast amount of such personalised data. For example, Genome-wide Association Scan (GWAS) projects have so far been published for over 100 human traits and diseases and many have made data available for thousands of people (`http://www.genome.gov/gwastudies`).

The datasets available in UK WTCCC data bank (`http://www.wtccc.org.uk`) will be used in this study, which includes multivariate personalised data of DNA SNPs, genomic, clinical, environmental and nutritional variables. If this case study is successful, this approach will be used for the development of a prognostic system to accurately predict clinical outcomes and appropriate treatment of CD patients in New Zealand and will be further applied for other diseases.

9.1.1 Crohn's Disease

Crohn's disease (CD) is a chronic and debilitating autoimmune disorder of the gastrointestinal tract. It is a major subtype of inflammatory bowel disease (IBD) which is diagnosed endoscopically and characterized by recurring episodes of abdominal pain, diarrhoea and weight loss. The aetiology of CD is complex and unclear but is generally thought to involve abnormal immune response to intestinal microorganisms in genetically predisposed individuals (Sartor, 1997). As a consequence of ongoing inflammatory "flares", a large number of CD patients will develop strictures and fistulae during the course of disease which can seriously impact the quality of life and often requires surgery (Vermeire, Van Assche, & Rutgeerts, 2007).

The incidence of CD is increasing dramatically in industrialized countries worldwide, including New Zealand (Loftus, 2004; Eason, Lee, & Tasman-Jones, 1982; Gearry & Day, 2008). As part of the "Canterbury IBD Project", Gearry and his colleagues (2006) conducted a comprehensive population-based survey of IBD in the Canterbury region and showed that rates of CD were amongst the highest reported worldwide - incidence: 17/100000 and prevalence: 155/100000. The age of diagnosis of CD in this cohort peaked at around 30 years. This study especially, indicates that CD is a mounting public health problem in New Zealand and requires research attention aimed at reducing personal and societal burden.

Unfortunately, there is currently no completely effective clinical strategy for treating Crohn's disease. Pharmacological treatment usually involves the trail of anti-inflammatory drugs (e.g. corticosteroids), immunomodulators (e.g. suppressants like Azathioprine), and biological (e.g. anti-tumor necrosis factor agents like Infliximab). Current treatment paradigms used in the clinic are the so-called "step-up" and "top-down" approaches. Step-up refers to the more classical approach that uses progressively intense treatment as disease severity increases, usually starting with lighter anti-inflammatory drugs. The top-down approach refers to early, more aggressive treatment with biological and immunosuppressants to prevent disease complications, for the purpose to improve the quality of life (Hommes et al., 2005; Baert, Caprilli, & Angelucci, 2007). The top-down approach can be highly effective but can increase risk of serious adverse reactions causing infection or cancer (Bongartz et al., 2006).

Whether or not a patient should be given step-up or top-down treatment for IBD is a controversial topic in clinical gastroenterology. The main issue is that it is difficult to accurately predict which of the two approaches will provide the most favorable outcome for an individual patient. It is increasingly believed that patients at high risk of developing CD complications will benefit more from top-down therapy. The inheritance risk probability of Crohn disease is unclear, because a variety of genetic and environmental factors are reported to be involved in literature. For example, people who smoke have are a higher risk to develop Crohn's disease than nonsmokers. Therefore, using accurate predictive tools to identify high-risk patients and personalised treatment is a major goal for clinicians.

9.1.2 SNPs Data for Crohn's Disease Risk Evaluation

The SNPs data used for Crohn's disease (CD) prediction is accessible from a UK's public data bank - Wellcome Trust Case Control Consortium (WTCCC) . The raw SNPs data is originally used in genome-wide association (GWA) studies of 14,000 cases of 7 major diseases and a shared set of 3,000 controls (WTCCC, 2007). An Affymetrix GeneChip mapping array set is used to record approximately 500,000 SNPs. However, the data size is extremely huge (more than 10GB) and in a unique format (*ped* file), which makes it difficult to be analysed by traditional computational models on PC. Therefore, the raw SNPs data needs to be preprocessed in an effectively way for further analysis.

Data Preprocessing

Unlike gene expression data is represented by continuous numerical value, SNPs data is described by categorical value which brings a challenge to conventional computational models for finding hidden patterns from the data. There have been some attempts to analyse categorical SNPs data in the literature. For example, Park (2007) and his colleagues employed a nearest shrunken centroid method to build a SNPs database - SNP@Ethnos. In their work, the categorical value of genotypes were coded by numerical values directly, and then the data were analysed by the NSCM of the R package *pamr*. Interestingly, same as gene selection playing an important role in gene expression data analysis, it has been found that only a small number of SNPs (known as relevant) have the genotype patterns highly in association with the object group of individuals (Liu, Li, Cheung, Sham, & Ng, 2009). Therefore, the enormous irrelevant SNPs should be excluded before the SNPs data is further exploited for modelling construction.

In this case study, the SNPs data was partially preprocessed and provided by Rod Lea and his research team at Environmental Science & Research (ESR) institute. Lea and his research team developed a Multi-factor Data Reduction (MDR) approach to identify the most important SNPs for predicting Crohn's disease (CD) risk. With their MDR method, the whole dataset was separated into 3 subsets: dataset A and B were used as the training sets, and dataset C was used as the testing data for validating the selected important features (SNPs) for CD predic-

tion. They used the whole genome association analysis tool set - PLINK (`http://pngu.mgh.harvard.edu/~purcell/plink/index.shtml`) for data propressing and analysis. Their MDR method finally selected 42 SNPs as the biomarkers for CD prediction based on the evaluation over two training datasets A and B. The training accuracy obtained by their MDR method was approximately 72% while the testing accuracy on validation dataset C was about 65%.

Based on Lea's work, I have used their preprocessed SNPs data in which each sample is represented by 42 SNPs (selected by their MDR method) in conjunction with 2 clinical factors (age and gender). All the samples for Crohn's disease prediction are randomly grouped into 3 subsets:

1. Set A contains 1049 samples in which 561 samples are diseased and 488 are controlled.
2. Set B contains 1045 samples in which 560 samples are Crohn's disease cases, while 485 are controlled.
3. Set C is an independent dataset that contains 106 samples (57 diseased cases vs. 49 controlled).

where the values for each SNP are relative risk values.

The proportion of missing values across the whole given SNPs data is 7.89%. I have replaced them by the major value of each feature (SNP). In this SNPs data, most features' values vary from 0 to 3, except the value of feature *Age* ranging from 1 to 10 (category value, the actual age is the product of the age multiplied by 10). Most SNPs have only 2 or 3 unique values, e.g. feature 3 (SNP $X2065477\_A$) has two risk values 0.92 and 2.14. To create a personalised problem space for each testing sample, a PMS uses Euclidean distance to measure the similarity between the samples across the feature space. However, the value of feature *Age* is out of the range that most feature values fall in, which significantly affects the distance measurement. Hence, I have normalised the feature *Age* into the range between 0 and 1.

9.2 Method

This study has approached the experiment into the following 7 steps:

1. Apply a global SVM model on training data (A+B), train the model and optimise the related parameters. Validate the trained global SVM model on testing subset D_x; This is an experiment for acquiring gold standard to compare the results from the proposed PM.

2. Use all features (42 SNPs + 2 clinical factors), and optimise the parameter K_v within the personalised problem space (neighbourhood) for each sample from testing subset D_x; Compute the classification accuracy using the model with optimised parameter K_v;

3. Optimise the parameters for each testing sample of D_x by using all features. Such parameters include: (1) K_v for the personalised problem space and, (2) c and γ for the kernel function of SVM model;

4. Optimise all related parameters, including K_v, c and γ. Also, select features (S^*) for each testing sample of D_x. Then, used the optimised PM model (with selected features (S^*) and optimised parameters (K_v, c and γ) to classifiy the testing dataset D_x;

5. Validation - Use the optimised PM model obtained in Step 4 to do the classification on the independent testing set C.

6. Evaluate the reliability of personalised modelling - The above Step 5 is repeated on a random sample from data C 20 times. The outcome is used for investigating the frequency of features selected in the 20 runs, and the local accuracies as well.

7. Create a globally optimised personalised model and profile for one sample, according to the finding and knowledge discovered in Step 6. Re-test the model for the given sample.

The experiment starts with the creation of a testing set that contains 10 randomly selected samples from dataset C. For the purpose to provide a fair comparison, these

10 random samples will be used all through the experiment in this case study, and are denoted as D_x:

$Sample:$ 392 408 269 458 120 857 1011 791 834 572

Five samples are from controlled group (class -1) while the other five samples are from diseased group (class 1). Each sample is represented by 44 features (42 SNPs plus 2 clinical factors).

The same SVM algorithm is used in this case study as the classifier for a fair comparison. The SVM model is derived from the well-known LibSVM package (Chang & Lin, 2001).

9.3 Experiment

In this case study, all the experiments are carried out on a PC with Matlab environment.

9.3.1 Step 1 - Global SVM Modelling

This section presents the experiment of global SVM modelling on SNPs data for Crohn's disease (CD) risk prediction. In order to find appropriate parameters for SVM, such as γ and c for the kernel function of SVM model, 5-fold cross-validation is employed for training datasets A and B. Then the trained SVM model is applied on the testing set C to perform the CD risk prediction. In this experiment, there is no feature selection, and we use all 44 features that are reported important for CD prediction in Lea's experiment.

Table 9.1 gives the experiment result of global SVM model on SNPs data for CD classification. The parameters for SVM kernel function are c (the cost) and γ that are optimised through 5-fold cross validation during the training stage. The overall accuracy for CD classification here is 0.70, which is not satisfactory for a test on 10 randomly selected samples. Moreover, there is no further information and knowledge that we can discover from this global SVM modelling experiment for designing

Table 9.1: *The experiment result of a global SVM model on the D_x of the SNPs data for CD classification, where class 1 accuracy is the classification accuracy of controlled samples (class label -1), while class 2 is the classification accuracy of diseased samples (class label 1).*

Sample ID:	392	408	269	458	120	857	1011	791	834	572
Actual	-1	-1	-1	-1	-1	1	1	1	1	1
Predicted	-1	*1*	-1	-1	*1*	1	*-1*	1	1	1
Parameters for SVM:				-c	200	-g	0.01			
Overall Accuracy:				**70%**						
Class 1 Accuracy:				60%		Class 2 Accuracy			80%	

medical treatment. In the next section, we will investigate the size of personalised problem space for CD risk evaluation using the proposed method - cEAP.

9.3.2 Step 2 - Personalised Modelling (Optimise K_v)

This experiment uses the same 10 random samples that are used in the global SVM modelling experiment. The learning function for CD prediction is still the LibSVM classifier. We implement PMS in a very simple way in which there is no feature selection. This approach only searches the optimal number of samples (K_v) for each sample x_v from the subset D_x of SNPs data. It evaluates different number of neighbouring samples (K_v) according to the classification performance of SVM.

The experiment result of this implementation of personalised modelling on D_x is illustrated in Table 9.2

Table 9.2: *The experiment result of a personalised modelling on the D_x of the SNPs data for CD classification (only optimise K_v), where local acc is the local accuracy that is defined as the accuracy of each given sample calculated on the its personalised problem space D_{pers}.*

Sample ID:	392	408	269	458	120	857	1011	791	834	572
Actual	-1	-1	-1	-1	-1	1	1	1	1	1
Predicted	-1	*1*	-1	*1*	-1	1	1	*-1*	1	1
Local Acc	0.75	*0.68*	0.63	*0.67*	0.78	0.77	0.75	*0.60*	0.79	0.61
K_v	51	38	33	34	19	32	38	39	43	19
Parameters for SVM:				-c	200	-g	0.01			
Overall Accuracy:				**70%**						
Class 1 Accuracy:				60%		Class 2 Accuracy			80%	

In this case, the personalised modelling method has optimised one parameter K_v and give the local accuracy for each testing sample. Although the performance of personalised modelling based method for classify CD samples is not improved in terms of accuracy, the result from personalised modelling brings us some information that may reveal the reason why it is not effective in this case. One possible reason is that the low training accuracy results in the misclassification. It is easy to elucidate from experiment results that in general, most samples with high local accuracy are successfully classified, except sample 572 that has a very low local accuracy (0.61). For example, sample 408 belongs to the controlled class, but is misclassified into diseased group. Its local accuracy across the personalised problem space (38 nearest neighbouring samples) is quite low - 68%. Similarly, the local accuracies of sample 458 and 791 are 0.67, and 0.60, which are not satisfactory in terms of classification performance, so that both of them are misclassified.

This experiment raises some open questions that need to be solved:

- ◇ How can we improve the local accuracy for a testing sample?
- ◇ Whether local accuracy will significantly affect the classification performance for testing new coming samples?

The next section will investigate these problems through another approach of personalised modelling.

9.3.3 Step 3 - Personalised Modelling (Optimise K_v and the Parameters of Learning Function)

In order to improve the local accuracy for the new testing sample, a new approach is proposed in this section for SNPs data analysis. Three parameters are optimised for building more efficient personalised, including the number of samples (K_v) and the parameters for SVM classifier (c and γ). The optimisation is evolved by a evolution strategy based algorithm, which is described in the method of cEAP in Chapter 8.

In this experiment, only two samples are misclassified: sample 408 and 458. The classification accuracy is slightly improved, but the local accuracy of each testing

Table 9.3: *The experiment result of a personalised modelling on the D_x of the SNPs data for CD classification (optimise K_v, c and γ), where c and γ are two parameters for SVM classifier*

Sample ID:	392	408	269	458	120	857	1011	791	834	572
Actual	-1	-1	-1	-1	-1	1	1	1	1	1
Predicted	-1	*1*	-1	*1*	-1	1	1	1	1	1
Local Acc	0.76	0.62	0.75	0.69	0.78	0.76	0.71	0.52	0.81	0.76
K_v	44	32	33	34	19	26	19	22	38	31
c(SVM)	233	236	233	244	387	232	244	235	352	371
γ(SVM)	0.0037	0.0042	0.0036	0.0056	0.0295	0.0034	0.0056	0.0040	0.0235	0.0269
Overall Accuracy:				**80%**						
Class 1 Accuracy:				60%		Class 2 Accuracy			100%	

samples is still unsatisfactory. The local accuracy of sample 408 and 458 is 0.62 and 0.69 calculated based on the personalised space of 32 samples and 34 samples, respectively. Such low local accuracy results in the misclassification, even though the parameters of SVM classifier is optimised for each testing sample. The experiment result is reported in Table 9.3.

Although this approach has optimised all the parameters relevant to the personalised modelling for SNP data analysis, the low local accuracy prevents the created personalised model working well on new testing data. Also, the question raised in last section that whether local accuracy will affect the performance of personalised modelling is not well answered here and still keeps uncertain. Additionally, this section has raised another open question that need to solved in this study:

◇ what other issues need to be considered in terms of improving the performance of personalised modelling for SNPs data analysis?

9.3.4 Step 4 - Personalised Modelling (Integrated Feature Selection, Neighbourhood Optimisation K_v and Parameter of Learning Function Optimisation)

It is shown in last section that the approach of personalised modelling has slightly improved the classification performance for CD prediction through optimizing relevant parameters K_v, c and γ. However, the experiment has not sufficiently proved the strength of personalised modelling over global modelling for a classification problem using SNPs data. Therefore, this section aims to answer the questions raised in

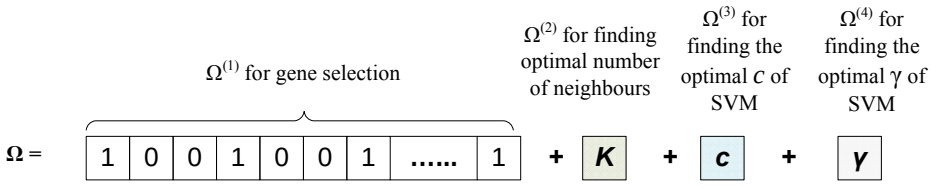


Figure 9.1: *The combined chromosome consists of 4 subcomponents* $\Omega_{(1)}$, $\Omega_{(2)}$, $\Omega_{(3)}$ *and* $\Omega_{(4)}$, *respectively.*

last section and give the solution.

In this case study, I have so far proposed three approaches to develop personalised modelling systems for CD risk evaluation using SNPs data. However, using only 10 randomly selected samples, the experiment has not achieved applausable results in terms of classification performance, even though a set of parameters in relation with the construction of personalised models have been optimised.

As discussed in early chapters, feature selection is a critical part in the construction of personalised models. The above three experiments are carried out based on the assumption that all 44 features are important to CD risk prediction. With personalised modelling, features are of different importance to different testing samples regarding to a biological problem. Therefore, feature selection should be applied on the SNPs data for CD classification, along with the parameter optimisation.

In this experiment, cEAP method is employed for feature selection and optimise parameters simultaneously. The problem space is decomposed into four subcomponents, which are shown in Figure 9.1. The subindividual of gene selection is binary-string encoded, while the rest subdividuals for parameter optimisation (K_v, c, γ) are real value encoded. The detailed description of cEAP method refers to Chapter 8.

Table 9.4 reports the classification result that lists all the features optimised by cEAP method. Also, the selected genes for each testing sample of subset D_x are summarised as follows:

Sample 392: 13 selected features;

Feature List: 1, 7, 11, 15, 19, 20, 21, 24, 25, 26, 37, 38, 40;

Sample 408: 17 selected features;

Feature List: 1, 3, 4, 6, 11, 15, 19, 21, 23, 28, 31, 32, 33, 37, 38, 39, 42;

Table 9.4: *The experimental results of a personalised modelling on the D_x of the SNPs data for CD classification (include feature selection and parameter optimisation for K_v, c and γ), where Num of features shows how many features are selected for testing a specific sample from D_x*

Sample ID:	392	408	269	458	120	857	1011	791	834	572
Actual	-1	-1	-1	-1	-1	1	1	1	1	1
Predicted	-1	*1*	-1	-1	-1	1	1	1	1	1
Local Acc	0.84	0.77	0.76	0.79	0.73	0.80	0.75	0.83	0.82	0.80
K_v	53	50	33	31	27	48	18	50	38	44
c(SVM)	312	345	335	308	257	300	299	349	291	293
γ(SVM)	0.0183	0.0265	0.0240	0.0173	0.0051	0.0153	0.0152	0.0274	0.0134	0.0138
Num of Features	13	17	22	23	18	21	17	22	29	19
Overall Accuracy:				**90%**						
Class 1 Accuracy:				80%		Class 2 Accuracy			100%	

Sample 269: 22 selected features;

Feature List: 1, 3, 4, 6, 7, 8, 10, 13, 15, 16, 17, 18, 20, 23, 28, 29, 31, 35, 37, 39, 42, 44;

Sample 458: 23 selected features;

Feature List: 1, 2, 5, 6, 8, 9, 10, 16, 18, 20, 21, 24, 26, 27, 28, 30, 35, 36, 38, 40, 41, 42, 44;

Sample 120: 18 selected features;

Feature List: 1, 3, 6, 7, 9, 13, 15, 16, 17, 19, 20, 23, 27, 29, 30, 37, 39, 44;

Sample 857: 21 selected features;

Feature List: 1, 2, 3, 4, 5, 6, 7, 11, 17, 21, 24, 26, 28, 31, 32, 33, 38, 39, 40, 43, 44;

Sample 1011: 17 selected features;

Feature List: 1, 5, 6, 7, 9, 10, 13, 15, 16, 20, 27, 29, 37, 38, 40, 41, 44;

Sample 791: 22 selected features;

Feature List: 1, 2, 3, 5, 6, 7, 8, 9, 12, 13, 14, 15, 17, 18, 19, 20, 22, 23, 26, 38, 39, 42;

Sample 834: 29 selected features;

Feature List: 1, 2, 5, 6, 8, 9, 12, 14, 15, 16, 17, 19, 20, 22, 23, 26, 27, 28, 30, 31, 33, 34, 35, 36, 37, 39, 41, 42, 44;

Sample 572: 19 selected features;

Feature List: 1, 3, 5, 7, 8, 10, 16, 18, 19, 20, 21, 23, 26, 29, 36, 38, 41, 42, 44;

It is obvious that this approach for personalised modelling has improved the performance in terms of classification accuracy. Only one controlled case (sample 408) is misclassified as diseased. This approach has achieved high local accuracy achieved across all testing samples (all of them are higher than 0.73), which could be the main reason that leads to the better performance of personalised modelling for CD prediction. This could be the main reason why the classification performance is

significantly better than that from global modelling and the insufficiently learned personalised model presented in the above experiments. It seems that a well designed personalised model is a competitive method for biomedical data analysis. At the same time, the experiment has clearly demonstrated the importance of feature selection and parameter optimisation in personalised modelling for a real world data analysis problem. Additionally, the selected features (SNPs) are of great importance for each individual patient sample for medical applications, such as personalised clinal treatment, personalised drug design and drug response. Global modelling approaches are not able to offer such information for building clinical decision systems.

Although I have so far demonstrated the superior classification performance of personalised modelling based method over global modelling on a real world SNPs dataset, the number of samples used in the above experiments is very small which is mainly for principle proofing. The good classification accuracy (90%) achieved in Step 4 using personalised modelling method might be created by chance, because of the limited number of testing samples. The next experiment will test more samples for the validation of the proposed methods for personalised modelling.

9.3.5 Step 5 - Validation

In this experiment, dataset C is used for validating the personalised model created in Step 4. The experiment consists of two modelling techniques for SNPs data analysis: (1) global SVM modelling; (2) personalised modelling (cEAP). The personalised modelling based method creates a better classification accuracy than the global model (73% vs. 70%), and provides a unique model for each testing sample. The classification accuracy of global SVM modelling on the testing data C 70% (class 1: 63%, class 2: 75%). The parameters for SVM model are: c=200, $\gamma = 0.01$. The method of personalised modelling (combining feature selection and all parameters) outperforms global SVM on this data. It yields 73% classification accuracy (class 1: 76%, class 2: 70%). The detailed experimental results is in Appendix L.

It is clear that using PM can extract some useful information and knowledge from the experiment over this SNPs testing dataset:

1. The average number of selected features is around 17;

2. The average size of personalised problem space (neighbourhood) is 70;

3. There are five most important features for predicting Crohn's disease. One is a clinical factor - *Age*. The others are 4 SNPs: X10210302_C, X17045918_C, X2960920_A and X7970701_G.

The discovered information and knowledge are of great importance to create a profile for each patient sample, and can be helpful for tailored treatment design and drug response and unknown types of disease diagnosis.

9.3.6 Step 6 - Reproducibility Evaluation

The main goal of the experiment in this section is to evaluate the reproducibility of personalised modelling based method proposed in Step 4. We are interested in whether the proposed personalised modelling based method is capable of producing highly consistent outcome for one sample? More specifically, this experiment is aiming to answer the questions:

1. What is the performance of proposed personalised modelling based method using global optimisation?

2. What is the variance of the local accuracy calculated from the global optimisation?

3. What is the frequency of each features to be selected during this experiment in 20 runs?

4. How many features should be selected for a successful prediction in general?

A sample (#392) is randomly selected and evaluated through 20 runs. The detailed experiment results are in Appendix M. Personalised modelling creates an applausable prediction accuracy: the prediction for sample 392 is always correct through all 20 runs. The average local accuracy for this sample through 20 runs is 82.45%. In addition, the personalised modelling method seems to work effectively on sample 392, as the computed local accuracy through 20 runs is very stable - the highest one is 83% and the lowest is 81%.

Figure 9.2 illustrates the selecting frequency of each feature for testing sample 392 during 20 runs. Here *Age* is again the most important feature for CD prediction, as it has been always selected during 20 runs. The next top 5 selected features are:

Feature Id	SNP Id	*Selecting frequency*(/20*times*)
20	X4252400_T	19
24	X2155777_T	18
12	X7683921_A	14
9	X2270308_T	13
23	X10883359_G	13

It seems that SNP X4252400_T and X2155777_T are two decisive factors for predicting CD risk specifically for sample 392.

Figure 9.3 summarizes the number of selected features in each run. It is easy to elicit that using approximately $12 \sim 16$ SNPs plus the feature of Age could lead to the successful prediction for sample 329. This finding is in agreement with the previous outcome in the experiment in Step 5.

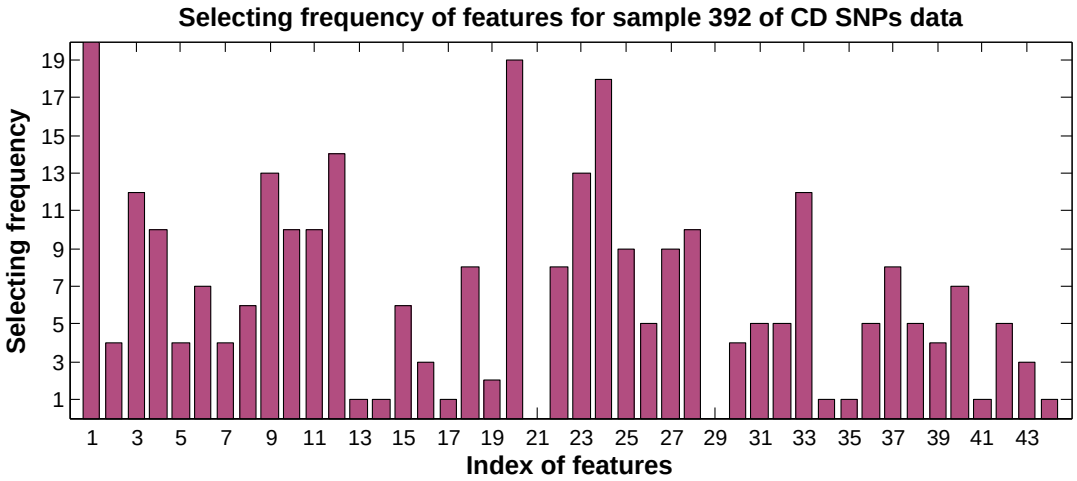


Figure 9.2: *The frequency of each feature to be selected from 20 runs for sample 392 of SNPs data for CD risk evaluation*

Personalised modelling based method works consistently well on a sample for CD risk prediction. The prediction outcome is reliable and the local accuracy is reproducible. The training procedure within the personalised problem space is stable through a number of runs (such process can be thought as a global optimisation). However, the selected SNPs is dependent on the parameter combination, such as the parameters

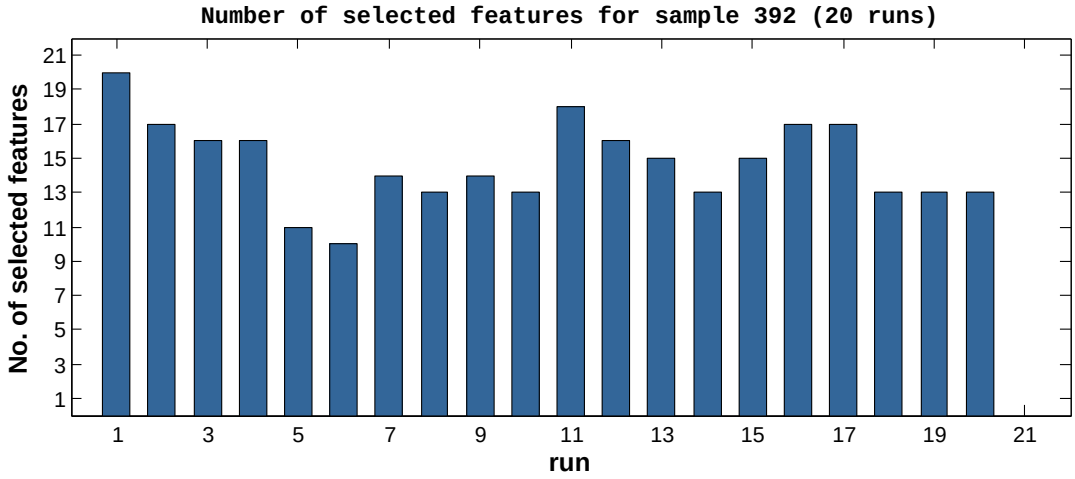


Figure 9.3: *The number of selected features for sample 392 in each of the 20 runs of the PM procedure*

of learning function. For personalised medical treatment design, this study suggests that the proposed PM method should run several times over the testing sample, to find the most informative features (SNPs) through different runs, i.e. the most commonly selected features in different testing runs.

9.3.7 Step 7 - Personalised Profiling

The goal of this step is to use the information and knowledge discovered from above steps to create a globally optimised profile for a new testing sample. As described in Step 6, personalised modelling method employs evolutionary computation to optimise the parameters and select features, which may create different personalised models for a new testing sample from different runs. The method for this experiment is as follows:

1. Take one random sample (e.g. here is sample 392) as the testing sample;
2. select a set of features S^* based on their selecting frequency during 20 runs in the experiment of Step 6;
3. Create a model on the training data A and B using the selected features, and compute the local accuracy;

4. Test the training model over sample 392, and provide a profile for the sample.

The assumption made for selecting features is that more frequently the selected features through a global optimisation process, more important the features for the given task. In this experiment, a threshold of selecting frequency is introduced to determine whether the feature is selected or not, i.e. whether the feature is selected more than half of all runs (10 out of 20 runs). There are 8 features selected more than 10 times during 20 runs shown in Figure 9.2. Thus, these 8 features are selected for building a personalised model:
1 (Age), 20 (X4252400_T), 24 (X2155777_T), 12 (X7683921_A), 9 (X2270308_T), 23 (X10883359_G), 3 (X2065477_A), 33 (X17221417_G).

The local training accuracy obtained on the training data (data A and B) is 100% accurate. The parameters used in the final optimal personalised model for sample 392 are suggested as:
the appropriate neighbourhood size is 51, and the parameters for SVM classifier are $c = 235, \gamma = 0.0284$.
With this global optimised personalised model, sample 392 is successfully classified.

From this experiment, it is easy to conclude that feature selection is more likely to be the most decisive factor for CD risk prediction. With a subset of most frequently selected features, a personalised modelling system is able to provide an accurate prediction for a new testing sample. Moreover, this experiment reveals that the local accuracy within the personalised problem space plays an important role for disease risk prediction. The good local accuracy is more likely to lead to a satisfactory testing performance. Therefore, how to improve the local accuracy within the personalised problem space under different situations will be an interesting research question in future study.

9.4 Discussion and Conclusion

Unlike synthetic problem simply focused on theoretical approval, the real world problem usually brings big challenges coming from different aspects, such as data collection, data preprocess (e.g. missing value and data scaling), etc. This chapter has addressed and discussed these issues in this chapter.

This chapter has presented a comparison experiment in which I have used global SVM modelling and different approaches of personalised modelling for Crohn's disease (CD) risk evaluation. To build a personalised model for each testing sample, I have approached the personalised modelling in four steps, from simple optimisation (only optimise one parameter: K_v) to the method of optimising a set of parameters and selecting features simultaneously. It shows that the approach only based on parameter optimisation may not be able to find an optimal personalised model for a particular data point, even though it may perform slightly better than global modelling approaches. To build an effective personalised models, a PMS should comprise the function modules for optimising relevant parameters optimisation and feature selection.

The main contribution of this case study is that it has theoretical proved the feasibility that personalised modelling is able to produce improved classification performance for real world biomedical data analysis. It has also demonstrated the strength of personalised modelling over global modelling for the classification over this specific SNPs data. Personalised modelling approach allows each individual patient to have a detailed unique profile, which is very useful for personalised clinical decision system.

This chapter also raises some open research problems that need to be investigated in my future study:

- How to find a suitable approach to visualize the profile in SNPs data analysis? SNPs data are generally category data, which brings a big challenge to visualise the profile in a PMS. It is not appropriate to employ the visualisation schema used for gene expression data analysis, because the change between different category values does not reveal any useful information for clinical decision making system. Hence, In order to effectively visualise the results from SNPs data analysis, it is critical to have in-depth biological understanding of SNPs data.

- How to balance the computational complexity and disease prediction accuracy? Personalised modelling usually needs intensive computation due to the creation of personalised model for each individual testing sample. GA based searching scheme brings more computational complexity, though it often comes with better performance.

- How to provide a more efficient way to measure the similarity of samples to create the personalised problem space (an appropriate neighbourhood)? In this study, Euclidean distance is used for calculating the neighbourhood. However, for SNPs data with categorical values, Euclidean distance may not the best option for similarity measurement.

These questions will be discussed in the next chapter as part of the future research.

CHAPTER 10

Conclusion and Future Study

"Our imagination is the only limit to what we can hope to have in the future."

- Charles F. Kettering

This research has presented a novel conceptual personalised modelling framework (PMF) for data analysis and knowledge discovery. To the best of my knowledge, this study is the first comprehensive study of personalised modelling (PM) from the point of view of computational intelligence. It is a feasibility analysis of PM for genomic data analysis and for possible clinical applications. Five novel methods have been developed during this course of study: (1) personalised modelling based gene selection, (2) increment search based approach for personalised modelling (iPM), (3) genetic algorithm search based approach for personalised modelling (gaPM), (4) compact GA search based personalised modelling (cGAPM), and (5) co-evolutionary algorithm based method for gene selection and parameter optimisation in personalised modelling (cEAP). These PM methods and systems have been applied on different benchmark gene expression datasets, a proteomic dataset and a SNPs dataset for disease classification. This research is not the end, but just a beginning to explore the filed of personalised modelling for knowledge discovery.

10.1 Summary of the Thesis

Every research endeavor starts with the objectives that guide the direction of the research. The ultimate objective of this research is to develop novel information methods and systems for PM and specifically for genomic data analysis and biomedical applications. In brief, this thesis has presented the following **main contributions** for personalised modelling study:

1. Analysed the problems related to PM and proposed potential solutions;
2. Developed five novel algorithms and methods for PM, including personalised feature selection and personalised profiling;
3. Developed two PMSs, specifically for different gene expression data analysis;
4. Developed one PMS for SNPs data analysis;
5. Gave the research direction for the future study.

The proposed personalised modelling system is the platform and system that integrates novel machine learning and modelling techniques for the specific research problems:

- ✓ feature selection;
- ✓ classification;
- ✓ disease outcome prediction;
- ✓ adaptation to new data;
- ✓ knowledge discovery and model validation;
- ✓ data sample profiling and results visualisation.

As an important part in PMS, Chapter 7 has proposed a novel feature (gene) selection method. It is a hybrid method comprising two feature selection techniques: filter and wrapper selection. In brief, PMGS applies filter on the objective data to measure

features' importance based on the calculated statistical scores and remove a large number of irrelevant features that have very low ranking scores. Wrapper selection works together with a learning function (a classifier) to evaluate the rest features through an evolving way.

This thesis has presented a critical analysis of problems related to PM. Such issues and challenges include: feature selection, imbalanced data structure, data sampling, the optimisation of relevant parameters, error measuring methods, inconsistency problem, profiling, etc. To solve these problems, this research has proposed a variety of algorithms and models in the development of personalised modelling. The proposed methods and systems for personalised modelling are evolving through incremental addition of new data to adaptive learning.

This study has investigated a variety classification models during the development of PMS. Such algorithms and models include KNN, WKNN, WWKNN, SVM, ECF, MLR, Naive Bayes classifier, TWNFI, etc. One interesting finding is that the experimental results have shown that classification models are important, but not the decisive factor for PMS construction. Feature selection and the quality of personalised problem space are two more critical factors that directly affect the classification/prediction performance of personalised modelling methods. The experimental outcomes have shown that a simple classifier works efficiently and is able to create satisfactory results in many cases, such as KNN, WKNN, and SVM. Some sophisticated algorithms for classification may yield good results in some difficult cases, but introduces huge computational burden.

This study has presented two approaches for implementing PMS: incremental search based approach (iPM) and GA search based approach. These two approaches are used for solving PM problems under different situations. Incremental search based approach works fast on large datasets and is able to produce good results in some cases of the classification on simple data. However, its performance is usually not as competitive as that from other more sophisticated methods, such as gaPM, because iPM only evaluates features individually and neglects their complex interactions.

GA search based personalised modelling system usually yields improved results than the from iPM, as it takes into account the relationship between features during feature selection. However, the proposed GA search based PM raises a problem: how to optimize the relevant parameters in conjunction with feature selection for building

a PMS? The experimental results of GA based PMS show that it does not outperform other modelling techniques in some cases of difficult prediction tasks. It does not take into account the relationship between candidate feature sets and the parameters, i.e. feature selection and relevant parameters are evaluated separately so that they may not be sufficiently optimised. To solve this problem, this thesis has proposed a novel method - an integrative (coevolutionary algorithm) based personalised modelling method (cEAP) for gene selection and parameter optimisation simultaneously.

In Chapter 8, cEAP method has been applied on four dataset - colon cancer data, leukaemia cancer data (Golub et al., 1999), lung cancer data (Gordon et al., 2002) and ovarian cancer data (Petricoin et al., 2002). cEAP consistently outperforms other methods for cancer classification, and discovers more useful information, including selected informative genes and optimal disease classification parameters specifically for the observed patient sample, which are helpful to construct the clinical decision support systems for cancer diagnosis and prognosis. For biological reference, some of experimental findings are proofed in the literature, e.g. the selected genes of leukaemia data by cEAP are reported as biomarkers in other published papers. Chapter 9 has theoretically proofed the strength of cEAP method that is superior to other global modelling techniques on a challenging real-world problem - using SNPs data for crohn's disease risk prediction.

In summary, personalised modelling offers a novel and integrated methodology that comprises different computational techniques for data analysis and knowledge discovery. Compared with the results obtained by other published methods, the new algorithms and methods based on PM have produced improved outcomes in terms of prediction accuracy and discovered more useful knowledge, because they take into account the location of new input sample in a subspace. The subspace (personalised space) excludes noise data samples and provides more precise information for analysing new input data sample.

PM is an adaptive and evolving technique, in which new data sample can be continuously added to the training dataset and subsequently contribute the learning process of personalised modelling. More importantly, the technique of personalised modelling offers a new tool to give a profile for each new individual data sample. Such characteristic makes personalised modelling based methods are promising for medical decision support systems, especially for complex human disease diagnosis

and prognosis, such as cancer and brain disease.

However, as a PMS creates a unique (personalised) model for each testing data sample, it requires more computational power and performance time than traditional global modelling methods, especially to train the models on large data sets. The proposed methods have shown the great potential for solving the problems that require individual testing. This study is the first step in this research direction and needs more in-depth understanding in bioinformatics for validating the experimental findings and knowledge discovery.

10.2 Directions of Future Research

This section presents some promising future direction for the development of the methods and systems in personalised modelling. However, the problems in bioinformatics are in principle very challenging and difficult due to the inconsistency in data and the lack of efficient methods. Although this study has proposed new algorithms and methods for personalised modelling in data analysis and biomedical problems, there are limitations and open research problems need to be investigated and solved in future research.

10.2.1 How to Deal with Variability in Data and Achieve Consistent Results

In this study, evolutionary computation has been applied in the proposed methods and algorithms for PM, the near optimal results can be different. This may affect the determination of choosing the markers (important features) for medical applications. Some partial solutions are proposed in the thesis (refer. Chap 9), e.g. applying multiple runs to ensure the consistent outcomes.

To verify the experimental results presented in this study, some of them have been discussed with the experts in related research fields. Some new findings will be reported in academic papers and will be applied to new biomedical applications, such as the new coming project of functional outcomes prediction using stroke data.

10.2.2 Similarity Measurement

In order to find a personalised problem space (an appropriate neighbourhood) for a new input data sample, there must be an effective model to measure the the similarity of the objective samples. In the proposed PMSs, the similarity measurement is computed by a Euclidean distance based method. Euclidean distance is a straightforward geometric distance that simply calculates the difference in each dimension (feature). It is widely used in data mining and pattern recognition tasks that involve calculating abstract "distances" between data points. However, Euclidean distance measurement has a main limitation: it is strongly sensitive to the scales of the objective variables (features). Personalised modelling problems are involved at dealing with different data which may have the variables with very different scales, such as age, gender, weight, blood pressure, etc. Using simple Euclidean distance might not be an appropriate solution to measure the similarity of this type of data. Moreover, Euclidean distance does not taken into account the correlation among variables.

As mentioned in early chapters, building an appropriate neighbourhood (personalised problem space) is a critical step in the personalised modelling for knowledge discovery. Different types of data need suitable methods for similarity measurement. In this sense, how to design an appropriate method to calculate the 'distance' between variables in different types of data will be one of the future research directions.

10.2.3 Optimisation Strategies

In this thesis, evolutionary computation has been used as the technique to evolve the candidate solutions of personalised models. Genetic algorithm and evolutionary strategy are two major algorithms incorporated in the optimizers for feature selection and parameter optimisation. However, GA based algorithm is often criticized by its high computational cost, which results in the difficulty of testing large dataset (e.g. CD's SNPs data).

Population-based incremental learning (PBIL) (Baluja, 1994) is able to produce a satisfactory performance with less computational cost in many cases (Rastegar & Hariri, 2006). It might be a good option to incorporate it as the optimising module into PMS to improve the computational efficiency.

10.2.4 Spiking Neural Network Models for Personalised Modelling

Spiking Neural Network (SNN) is a biologically plausible model of a spiking neuron that includes a dynamic network of genome items, such as genes and proteins. The interactions of genes in neurons affect the whole network that leads to the change of a gene expression function (Kasabov, Benuskova, & Wysoski, 2005). Recently, there are some attempts to apply SNN on benchmark datasets for classification problems (Belatreche, Maguire, & McGinnity, 2007; Kasabov et al., 2005; Ponulak & Kasiński, 2010). It seems that SNN could be potentially a powerful tool to be employed in the PMS for more complex problems of patter recognition and knowledge discovery.

10.2.5 Biomedical Applications Using PMS

The personal data, such as gene expression data, SNPs data and clinical data are collected and accumulated massively these days. Such circumstance makes the data more accessible for analysis. However, it is always a big challenge to convert the data to precious knowledge that can benefit scientific community. The methods and system for PM developed in this research are expected to be explored more datasets and applied new biomedical applications.

The potential project using personalised modeling is to develop knowledge engineering and knowledge discovery methods and systems to enable personalised prediction of outcomes after brain injury (BI). Reliable prediction of BI risk and outcomes for the individual is likely to enable personalised rehabilitation, management and prevention. New knowledge and better understanding of environmental, clinical and genetic interplays are expected to be achieved and directed towards practical use.

References

Alizadeh, A. A., Eisen, M., Davis, R. E., Ma, C., Lossos, I. S., Rosenwald, A. et al. (2000). Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, *403*(6769), 503-11. xxix, 14, 64, 104, 108

Allison, D., Cui, X., Page, G. P. & Sabripour, M. (2006). Microarray data analysis: from disarray to consolidation and consensus. *Nature Reviews Genetics*, *7*(1), 55-65. 21, 85

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D. et al. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences of the United States of America*, *96*, 6745-50. xxix, 47, 51, 104, 105, 127, 147, 149, 157, 159, 160

Alter, O., Brown, P. O. & Botstein, D. (2000). Singular value decomposition for genome-wide expression data processing and modeling. *PNAS*, *97*(18), 10101-6. 71

Ambroise, C. & McLachlan, G. J. (2002). Selection bias in gene extraction on the basis of microarray gene-expression data. *PNAS*, *99*(10), 6562-6566. 64, 86

Anderson, J. (2000). *Cognitive psychology and its implications*. New York: Worth Publishers. 2

Anderson, J., Hansen, L. L., Mooren, F. C., Post, M., Hug, H., Zuse, A. et al. (2006). Methods and biomarkers for the diagnosis and prognosis of cancer and other diseases: Towards personalized medicine. *Drug Resistance Updates*, *9*(4-5), 198-210. 4, 98

Asyali, M. H., Colak, D., Demirkaya, O. & Inan, M. S. (2006). Gene expression profile classification: A review. *Current Bioinformatics*, *1*, 55-73. 64, 88

Babovic, V. (1996). *Hydroinformatics: Emergence, evolution, intelligence*. Taylor & Francis. 26

Baert, F., Caprilli, R. & Angelucci, E. (2007). Medical therapy for crohn's disease: top-down or step-up? *Dig Dis.*, *25*(3), 260-6. 166

Baggerly, K. A., Morris, J. S., Edmonson, S. R. & Coombes, K. R. (2005). Signal in noise: Evaluating reported reproducibility of serum proteomic tests for ovarian cancer. *J Natl Cancer Inst 2005*, *97*(4), 307-309. 20

Baker, J. E. (1985). Adaptive selection methods for genetic algorithms. In *1st*

international conference on genetic algorithms (p. 101-111). 29

Baldi, P. & Hatfield, G. W. (2002). *Dna microarrays and gene expressions.* Cambridge, UK: Cambridge University Press. 3, 15, 16

Baluja, S. (1994). *Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning* (Tech. Rep.). Carnegie Mellon University. 188

Bartek, J. & Lukas, J. (2001). Are all cancer genes equal. *Nature*, *411*, 1001-1002. 18

Beart, R. W. (1995). Pouchitis: A clarification. *Gastroenterology*, *109*(3), 1022 - 1023. 125

Beckers, G. J. & Conrath, U. (2006). Microarray data analysis made easy. *Trends in Plant Science*, *11*(7), 322 - 323. 17

Belatreche, A., Maguire, L. P. & McGinnity, T. M. (2007). Advances in design and application of spiking neural networks. *Soft Comput.*, *11*(3), 239-248. 189

Ben-Dor, A., Bruhn, L., Frideman, N., Schummer, M. & Yakhini, Z. (2000). Tissue classification with gene expression profiles. In *Annual conference on research in computational molecular biology: Proceedings of the fourth annual international conference on computational molecular biology* (p. 54-64). Tokyo, Japan: ACM Press. 15

Ben-Dor, A., Friedman, N. & Yakhini, Z. (2001). Class discovery in gene expression data. *RECOMB*, 31-38. 71

Benson, D. A., Ilene, K.-M., Lipman, D. J., Ostell, J., Rapp, B. A. & Wheeler, D. L. (2002). Genbank. *Nucleic Acids Res.*, *30*(1), 17-20. 17

Beyer, H.-G. & Schwefel, H.-P. (2002). Evolution strategies: A comprehensive introduction. *Natural Computing*, *1*, 3-52. 31, 32

Bezdek, J. C. (1982). *Pattern recognition with fuzzy objective function algorithms.* Norwell, MA, USA: Kluwer Academic Publishers. 51

Bhattacharyya, I., Bandypopadhyay, A. K., Gupta, B., Chattopadhyay, A., Chattopadhyay, R. & Yasumoto, K. (2009). Vector ga: a novel enhancement of genetic algorithms for efficient multi-variable or multi-dimensional search. *ACM SIGSOFT Software Engineering Notes*, *34*(6), 1-5. 162

Bishop, C. M. (1995). *Neural networks for pattern recognition.* Oxford University Press. 44

Bongartz, T., Sutton, A. J., Sweeting, M. J., Buchan, I., Matteson, E. L. & Montori, V. (2006). Anti-tnf antibody therapy in rheumatoid arthritis and the risk of

serious infections and malignancies: Systematic review and meta-analysis of rare harmful effects in randomized controlled trials. *JAMA*, *295*(19), 2275-2285. 166

Boser, B. E., Guyon, I. M. & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the 5th annual acm workshop on computational learning theory* (p. 144-152). ACM Press. 41

Bosnic, Z., Kononenko, I., Robnik-Sikonja, M. & Kukar, M. (2003). Evaluation of prediction reliability in regression using the transduction principle. In *Eurocon 2003. computer as a tool. the ieee region 8* (Vol. 2, p. 99-103 vol.2). 45

Boyd, L. K., Mao, X. & Lu, Y.-J. (2009). Use of snps in cancer predisposition analysis, diagnosis and prognosis: tools and prospects. *Expert Opinion on Medical Diagnostics*, *3*(3), 313-326. 22

Bozic, I., Zhang, G. & Brusic, V. (2005). Predictive vaccinology: Optimisation of predictions using support vector machine classifiers. In *Ideal* (p. 375-381). 39

Braga-Neto, U., Hashimoto, R., Dougherty, E. R., Nguyen, D. V. & Carroll, R. J. (2004). Is cross-validation better than resubstitution for ranking genes? *Bioinformatics*, *20*(2), 253-258. 85, 88

Bratko, I., Mozetic, I. & Lavac, N. (1989). *Kardio: A study in deep and qualitative knowledge for expert systems*. The MIT Press. 2

Breiman, L. & Spector, P. (1992). Submodel selection and evaluation in regression: The x-random case60. *International Statistical Review*, *60*, 291-319. 86, 87, 126

Breiman, L., Stone, C., Friedman, J. & Olshen, R. (1984). *Classification and regression trees*. Wadsworth International Group. 2

Bremermann, H. J. (1958). *The evolution of intelligence. the nervous system as a model of its environment.* (Tech. Rep. No. Technical Report No. 1). University of Washington. (Contract No. 477(17)) 26

Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, *2*, 121-167. 39, 41, 42

Cancer facts & figures 2008 (Tech. Rep.). (2008). American Association for Cancer Research. 18

Carlson, B. (2008). Snps - a shortcut to personalized medicine: Medical applications are where the market's growth is expected. *Genetic Engineering & Biotechnology News*. 22

Chang, C.-C. & Lin, C.-J. (2001). LIBSVM: a library for support vector machines

[Computer software manual]. 104, 146, 170

Cho, S.-B. & Won, H.-H. (2003). Machine learning in dna microarray analysis for cancer classification. In *Conferences in research and practice in information technology: Proceedings of the first asia-pacific bioinformatics conference on bioinformatics 2003* (Vol. 19, p. 189-198). Adelaide, Australia: Australian Computer Society. 64

Chuang, H.-Y., Liu, H., Brown, S., McMunn-Coffran, C., Kao, C.-Y. & Hsu, D. F. (2004). Identifying significant genes from microarray data. bibe 2004: 358-365. In *Bibe 2004. proceedings. fourth ieee symposium* (p. 358 - 365). 64

Coello, C. A. C., Lamont, G. B. & Veldhuizen, D. A. V. (2007). *Evolutionary algorithms for solving multi-objective problems.* Springer. 139

Collins, F. S. & Barker, A. D. (2008). Mapping the cancer genome. *Scientific American*, *18*(3), 22-29. 19

Cooper, G., Aliferis, C., Ambrosino, R., Aronis, J., Buchanan, B., Caruana, R. et al. (1997). An evaluation of machine learning methods for predicting pneumonia mortality. *Artificial Intelligence in Medicine*, *9*, 107-138. 2

Darwin, C. & Wallace, A. R. (1858). On the tendency of species to form varieties; and on the perpetuation of varieties and species by natural means of selection. *Journal of the Proceedings of the Linnean Society of London*, *Zoology 3*, 46-50. 25

De Jong, K. A. (1975). *An analysis of the behavior of a class of genetic adaptive systems.* Phd, University of Michigan. 29

Denis, G. V. (2008). Imatinib mesylate gleevec and the emergence of chemotherapeuticss drug-resistant mutations. In H. L. Kaufman, S. Wadler & K. Antman (Eds.), *Molecular targeting in oncology.* Totowa, NJ: Humana Press. 19

D'haeseleer, P., Liang, S. & Somogyi, R. (2000). Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics*, *16*(8), 707-26. 48

DiChristina, M. (2008). In this issue. *Scientific American*, *18*(3), 1-2. 18

Ding, C. & Peng, H. (2003). Minimum redundancy feature selection for gene expression data. In *Proc. ieee computer society bioinformatics conference (csb 2003)* (p. 523-529). Stanford, CA. 67, 71

Draghici, S., Khatri, P., Eklund, A. & Szallasi, Z. (2006). Reliability and reproducibility issues in dna microarray measurements. *Trends Genet*, *22*(2), 101-9. 15

Draghici, S., Kulaeva, O., Hoff, B., Petrov, A., Shams, S. & Tainsky, M. A. (2003). Noise sampling method: an anova approach allowing robust selection of differentially regulated genes measured by dna microarrays. *Bioinformatics*, *19*(11), 1348-1359. 67, 71

Dudoit, S., Fridlyand, J. & Speed, T. P. (2000, June). *Comparison of discrimination methods for the classification of tumors using gene expression data* (Tech. Rep.). UC Berkeley. 65

Dudoit, S., Yang, Y., Callow, M. J. & Speed, T. (2002). Statistical methods for identifying differentially expressed genes in replicated cdna microarray experiments. *Stat. Sinica*, *12*, 111-139. 70

Duncan, B. & Olson, A. (1996). Applications of evolutionary programming for the prediction of protein-protein interactions. In L. F. V, P. Angeline & T. Baeck (Eds.), *Evolutionary programming* (p. 411-417). Cambridge: MIT Press. 33

Eason, R. J., Lee, S. P. & Tasman-Jones, C. (1982). Inflammatory bowel disease in auckland, new zealand. *Aust N Z J Med*, *12*(2), 125-31. 166

Efron, B. (1979). Bootstrap methods: another look at the jacknife. *Annals of Statistics*, *7*(1), 1-26. 86, 87

Efron, B. (1983). Estimating the error rate of a prediction rule: Improvement on cross-validation. *J. of the American Statistical Association*, *78*, 316-331. 87

Efron, B., Tibshirani, R., Storey, J. & Tusher, V. (2001). Empirical bayes analysis of a microarray experiment. *Journal of the American Statistical Association*, *96*, 1151-1160. 67

Ehrenreich, A. (2006). Dna microarray technology for the microbiologist: an overview. *Appl Microbiol Biotechnol*, *73*(2), 255-273. 15

Eklund, A. & Szallasi, Z. (2008). Correction of technical bias in clinical microarray data improves concordance with known biological information. *Genome Biology*, *9*(2), R26. 20, 21

Fawcett, T. (2004). *Roc graphs: Notes and practical consideration for researchers* (Technical report No. HPL2003-4). HP Laboratories. 90

Ficici, S. & Pllack, J. (2000). A game-theoretic approach to the simple coevolutionary algorithm. In *the sixth parallel problem solving from nature* (p. 467-476). Springer-Verlag. 140

Ficici, S. G. (2004). *Solution concepts in coevolutionary algorithms.* Unpublished doctoral dissertation, Brandeis University. 139

Fogel, D., Fogel, L. & Porto, V. (1990). Evolving neural networks. *Biological*

Cybernetics, *63*, 487-493. 33

Fogel, D., Wasson, E., Boughton, E. & Porto, V. (1997). A step toward computer-assisted mammography using evolutionary programming and neural networks. *Cancer Letters*, *119*, 93-97. 33

Fogel, D., Wasson, E., Boughton, E. & Porto, V. (1998). Evolving artificial neural networks for screening features from mammograms. *Artificial Intelligence in Medicine*, *14*, 317-326. 33

Fogel, L. (1962). Autonomous automata. *Industrial Research*, *4*, 14-19. 32

Fogel, L., Owens, A. J. & Walsh, M. J. (1966). *Artificial intelligence through simulated evolution.* New York: John Wiley. 26

Forrest, S. & Mayer-Kress, G. (1991). Genetic algorithms, nonlinear dynamical systems, and global stability models. In L. Davis (Ed.), *The handbook of genetic algorithms.* New York: NY: Van Nostrand Reinhold. 30

Freund, Y. & Schapire, R. E. (1998). Large margin classification using the perceptron algorithm. In *Machine learning* (pp. 277–296). 81

Friedberg, R. M. (1958). A learning machine, part i. *IBM Journal of Research and Development*, *2*, 2-13. 26

Friedberg, R. M., Dunham, B. & North, J. H. (1959). A learning machine: Part ii. *IBM Journal of Research and Development*, *3*, 282-287. 26

Friedman, G. J. (1959). Digital simulation of an evolutionary process. *General Systems Yearbook*, *4*, 171-184. 26

Furey, T., Cristianini, N., Duffy, N., Bednarski, D., Schummer, M. & Haussler, D. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, *16*(10), 906-914. 48, 67

Furey, T. S., Cristianini, N., Duffy, N., W, D. & Haussler, D. (2000). *Support vector machine classification and validation of cancer tissue samples using microarray expression data.* 81

Galvin, J. & Ginsberg, S. (2004). Expression profiling and pharmacotherapeutic development in the central nervous system. *Alzheimer Dis. Assoc. Disord.*, *18*, 264-69. 17

Galvin, J., Powlishta, K., Wilkins, K., McKeel, D. J., Xiong, C., Grant, E. et al. (2005). Predictors of preclinical alzheimer disease and dementia: a clinicopathologic study. *Arch Neurol*, *62*(5), 758-65. 17

Garrison, L. P. & Austin, M. J. F. (2007). The economics of personalized medicine :

A model of incentives for value creation and capture. *Drug information journal*, *41*(4), 501-509. 98

Gearry, R. B. & Day, A. S. (2008). Inflammatory bowel disease in new zealand children - a growing problem. *N Z Med J.*, *121*(1283), 5-8. 166

Gearry, R. B., Richardson, A., Frampton, C. M., Collett, J. A., Burt, M. J., Chapman, B. et al. (2006). High incidence of crohn's disease in canterbury, new zealand: results of an epidemiologic study. *Inflamm Bowel Dis.*, *12*(10), 936-43. 166

Gehlhaar, D. & Fogel, D. (1996). Tuning evolutionary programming for conformationally flexible molecular docking. In L. F. V, P. Angeline & T. Baeck (Eds.), *Evolutionary programming* (p. 419-429). Cambridge, MA: MIT Press. 33

Gerstein, M. B., Bruce, C., Rozowsky, J. S., Zheng, D., Du, J., Korbel, J. O. et al. (2007). What is a gene, post-ENCODE? History and updated definition. *Genome Research*, *17*(6), 669-681. 12

Gibbs, W. W. (2003). Untangling the roots of cancer. *Scientific American*, *289*(1), 56-65. 18

Ginsburg, G. S. & McCarthy, J. J. (2001). Personalized medicine: revolutionizing drug discovery and patient care. *Trends in Biotechnology*, *19*(2), 491-96. 98

Glymour, C., Madigan, D., Preigbon, D. & Smyth, P. (1996). Statistical inference and data mining. *Communication of the ACM*, *39*(11), 35-41. 64

Goldberg, D. (1989). *Geneticalgorithm in search, optimization and machine learning.* MA: Kluwer Academic. 28

Goldman, L., Cook, E., Brand, D., Lee, T., Rouan, G., Weisberg, M. et al. (1988). A computer protocol to predict myocardial infarction in emergency department patients with chest pain. *N Engl J Med*, *318*(13), 797-803. 2

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mersirov, J. P. et al. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, *286*, 531-537. xxix, 13, 65, 78, 147, 149, 155, 156, 161, 186

Gordon, G. J., Jensen, R., Hsiao, L.-L., Hsiaox, S. & JE, B. (2002). Translation of microarray data into clinically relevant cancer diagnostic tests using gege expression ratios in lung cancer and mesothelioma. *Cancer Research*, *62*, 4963-67. xxx, 147, 149, 186

Graepel, T., Burger, M. & Obermayer, K. (1998). Self-organizing maps: Generalizations and new optimization techniques. *Neurocomputing*, *21*, 173–190.

51

Gurwitz, D., Lunshof, J. E. & Altman, R. B. (2006). A call for the creation of personalized medicine databases. *Nature Reviews Drug Discovery*, *5*, 23-26. 98

Guyon, I. & Elisseeff, A. (2006). An introduction to feature extraction. In I. Guyon, S. Gunn, M. Nikravesh & L. A. Zadeh (Eds.), *Feature extraction: Fondations and applications* (Vol. 207, p. 1-25). Heidelberg: Springer-Verlag. 68, 75

Guyon, I., Weston, J., Barnhill, S. & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, *46*(1), 389-422. 74

Hall, D. (2007, 12-Dec-2009). *A single nucleotide polymorphism is a change of a nucleotide at a single base-pair location on dna.* `http://en.wikipedia.org/wiki/File:Dna-SNP.svg`. x, 21

Hamamoto, Y., Uchimura, S. & Tomita, S. (1996). On the behavior of artificial neural network classifiers in high-dimensional spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *18*(5), 571-574. 65

Harik, G. R., Lobo, F. G. & Goldberg, D. E. (1999). The compact genetic algorithm. *IEEE Transactions on Evolutionary Computation*, *3*(4), 287-297. 33

Hastie, T., Tibshirani, R., M.B., E., Alizadeh, A., R., L., L., S. et al. (2000). 'gene shaving' as a method for identifying distinct sets of genes with similar expression patterns. *Genome Biology*, *1*(2), 1-21. 71

Henkes, M., Kuip, H. v. der & Aulitzky, W. E. (2008). Therapeutic options for chronic myeloid leukemia: focus on imatinib glivec, gleevec. *Therapeutics and Clinical Risk Management*, *4*(1), 163-187. 19

Herdy, M. (1992). Reproductive isolation as strategy parameter in hierarchically organized evolution strategies. In R. M?nner & B. Manderick (Eds.), *Parallel problem solving from nature* (Vol. 2, p. 207-217). Amsterdam: Elsevier. 32

Hillis, W. D. (1991). Co-evolving parasites improve simulated evolution as an optimization procedure. *Physica D: Nonlinear Phenomena*, *42*(1-3), 228-234. 141

Hindorff, L. A., Sethupathy, P., Junkins, H. A., Ramos, E. M., Mehta, J. P., Collins, F. S. et al. (2009). Potential etiologic and functional implications of genome-wide association loci for human diseases and traits. *Proc Natl Acad Sci USA*, *106*(23), 9362-9367. 165

Holland, J. (1975). *Adaptation in natural and artificial systems.* The University of Michigan Press. 27

Holland, J. (Ed.). (1986). *Escaping brittleness: The possibilities of general-purpose learning algorithms applied to parallel rule-based systems* (Vol. 2). Los Altos, CA: Morgan Kaufmann. 29, 141

Holland, J. H. (1992). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence.* Cambridge, MA: Mit Press. 30, 35

Hommes, D., Baert, F., Assche, G. van, Caenepeel, P., Vergauwe, P., Tuynman, H. et al. (2005). A randomized controlled trial evaluating the ideal medical management for crohn's disease (cd): Top-down versus step-up strategies. 166

Hosking, J., Pednault, E. & Sudan, E. (1997). A statistical perspective on data mining. *Future Generation Computing System*, *13*(2), 117-134. 64

Hu, Y. (2008). *Gene selection based on consistency modelling, algorithms and applications - genetic algorithm application in bioinformatics data analysis.* Saarbrücken, Germany: Vdm Verlag. 65, 74, 146

Hu, Y. & Kasabov, N. (2009). Coevolutionary method for gene selection and parameter optimization in microarray data analysis. In C. Leung, M. Lee & J. Chan (Eds.), *Neural information processing* (p. 483-492). Berlin / Heidelberg: Springer-Verlag.

Hu, Y., Song, Q. & Kasabov, N. (2008). Personalized modeling based gene selection for microarray data analysis. In *the 15th international conference of neural information processing.* Auckland, New Zealand: Springer.

Hu, Y., Song, Q. & Kasabov, N. (2009). Personalized modeling based gene selection for microarray data analysis. In M. Kóppen, N. Kasabov & G. Coghill (Eds.), *Advances in neuro-information processing* (p. 1221-1228). Springer.

Huber, W., Von Heydebreck, A. & Vingron, M. (2003). Analysis of microarray gene expression data. In *in handbook of statistical genetics: 2nd edn.* Wiley. 15

Huerta, E. B., Duval, B. & Hao, J. (2006). A hybrid ga/svm approach for gene selection and classification of microarry data. *Lecture Notes in Computer Science*, *3907*, 34-44. 74

Inza, I., Sierra, B., Blanco, R. & Larranaga, P. (2002). Gene selection by sequential search wrapper approaches in microarray cancer class prediction. *Journal of Intelligent and Fuzzy Systems*, *12*(1), 25-33. 74

Ioannidis, J. P. A. (2005). Microarrays and molecular research: noise discovery? *Lancet*, *365*, 453-455. 20

Iwao-Koizumi, K., Matoba, R., Ueno, N., Kim, S. J. & al., e. (2005). Prediction of docetaxel response in human breast cancer by gene expression profiling. *American Society of Clinical Oncology*, *33*(3), 422-431. 71

Jaeger, J., Sengupta, R. & Ruzzo, W. (2003). Improved gene selection for classification of microarrays. In *Pacific symposium on biocomputing* (p. 53-64). Kauai, Hawaii. 67

Japkowicz, N. (2000). The class imbalance problem: Significance and strategies. In *In proceedings of the 2000 international conference on artificial intelligence (icai* (p. 111-117). 76

Japkowicz, N., Myers, C. & Gluck, M. (1995). A novelty detection approach to classification. In *In proceedings of the fourteenth joint conference on artificial intelligence* (pp. 518–523). 77

Japkowicz, N. & Stephen, S. (2002). The class imbalance problem: A systematic study. *Intell. Data Anal.*, *6*(5), 429-449. 76, 77

Jeon, J.-Y., Kim, J.-H. & Koh, K. (1997). Experimental evolutionary programming-based high-precision control. *IEEE Control Sys. Tech.*, *17*, 66-74. 33

Jorgensen, T. J. (2008, January). From blockbuster medicine to personalized medicine. *Personalized Medicine*, *5*(1), 55-64. 165

Juille, H. & Pollak, J. (1996). Co-evolving interwined spirals. In *the fifth annual conference on evolutionary programming* (p. 461-468). MIT Press. 140

Kasabov, N. (1998). Ecos - a framework for evolving connectionist systems and the 'eco' training method. In *Iconip'98 - the fifth international conference on neural information processing* (Vol. 3, p. 1232-1235). Kitakyushu, Japan: IOS Press. 36

Kasabov, N. (2001). Evolving fuzzy neural networks for supervised / unsupervised online knowledge-based learning. *Systems, Man and Cybernetics, Part B, IEEE Transactions on*, *31*(6), 902-918. x, 37, 38

Kasabov, N. (2002). Evolving connectionist systems. In *Methods and applications in bioinformatics, brain study and intelligent machines.* London: Springer-Verlag. 37, 51, 214

Kasabov, N. (2003). *Evolving connectionist systems.* Springer-Verlag London. 6, 35, 36, 217

Kasabov, N. (2007a). *Evolving connectionist systems: The knowledge engineering approach.* London: Springer. 6, 25, 35, 98

Kasabov, N. (2007b). Global, local and personalized modelling and pattern discovery

in bioinformatics: An integrated approach. *Pattern Recognition Letters*, *28*(6), 673-685. xxiv, 5, 42, 46, 49, 56, 61, 98

Kasabov, N. (2009). Soft computing methods for global, local and personalised modeling and applications in bioinformatics. In V. E. Balas, J. Fodor & A. Varkonyi-Koczy (Eds.), *Soft computing based modeling in intelligent systems* (p. 1-17). Springer. 46

Kasabov, N., Benuskova, L. & Wysoski, S. G. (2005). *Computational neurogenetic modeling: Integration of spiking neural networks, gene networks, and signal processing techniques* (Vol. 3697). 189

Kasabov, N., Hu, Y. & Liang, L. (2009). Personalised modelling for risk and outcome prognosis on a case study of brain disease. In *1st international congress on clinical neurology & epidemiology.* Munich, Germany. 98

Kasabov, N. & Pang, S. (2004). Transductive support vector machines and applications in bioinformatics for promoter recognition. In *Proc. of international conference on neural network and signal processing.* IEEE Press. 45

Kasabov, N. & Song, Q. (2002). Denfis: dynamic evolving neural-fuzzy inference system and its application for time-series prediction. *Fuzzy Systems, IEEE Transactions on*, *10*(2), 144-154. 37, 51

Kato, T., Kamoto, S., Hyuga, M. & Karube, I. (2007). Snps typing based on the formation of fluorescent signaling dna aptamers which bind to bile acids. *NUCLEIC ACIDS SYMP SER (OXF)*, *51*(1), 97-98. 22

Keller, A. D., Schummer, M., Hood, L. & Ruzzo, W. L. (2000). *Bayesian classification of dna array expression data* (Tech. Rep.). University of Washington. 80

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International joint conference on artificial intelligence (ijcai).* Montreal, Quebec, Canada. 86, 126

Kohavi, R. & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, *97*(1-2), 273-324. xi, 72, 75

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, *43*, 59-69. 51

Krishnakumar, K. & Goldberg, D. E. (1992). Control system optimization using genetic algorithms. *Journal of Guidance, Control and Dynamics,*, *15*(3), 735-40. 30

Kroczak, T., Baran, J., Pryjma, J., Siedlar, M., Reshedi, I., Hernandez, E. et al.

(2006). The emerging importance of dna mapping and other comprehensive screening techniques, as tools to identify new drug targets and as a means of (cancer) therapy personalisation. *Expert Opin Ther Targets*, *10*(2), 289-302. 19

Kubat, M., Holte, R. C., Matwin, S., Kohavi, R. & Provost, F. (1998). Machine learning for the detection of oil spills in satellite radar images. In *Machine learning* (pp. 195–215). 77

Kubat, M. & Matwin, S. (1997). Addressing the curse of imbalanced training sets: One-sided selection. In *In proceedings of the fourteenth international conference on machine learning* (pp. 179–186). Morgan Kaufmann. 77

Kukar, M. (2002). Transductive reliability estimation for medical diagnosis. *Artificial Intelligence in Medicine*, *29*, 2003. 45

Lai, C., Reinders, M. & Wessels, L. (2004). On univariate selection methods in gene expression datasets. In *Tenth annual conference of the advanced school for computing and imaging* (p. 335-341). Port Zelande, The Netherlands. 71

Lavrac, N., Keravnou, E. & Zupan, B. (1997). Intelligent data analysis in medicine and pharmacology: An overview. In N. Lavrac, E. Keravnou & B. Zupan (Eds.), *Intelligent data analysis in medicine and pharmacology* (p. 1-13). Kluwer. 2

Lazarova, M. (2008). Efficiency of parallel genetic algorithm for solving n-queens problem on multicomputer platform. In *the 9th wseas international conference on evolutionary computing* (p. 51-56). Sofia, Bulgaria: World Scientific and Engineering Academy and Society (WSEAS). 30

Lee, K. E., Sha, N., Dougherty, E. R., Vannucci, M. & Mallick, B. K. (2003). Gene selection: a Bayesian variable selection approach. *Bioinformatics*, *19*(1), 90-97. 67

Levene, H. (1960). Robust tests for equality of variances. In I. Olkin & P. Alto (Eds.), *Contributions to probability and statistics: Essays in honor of harold hotelling* (p. 278-292). Stanford, CA: Stanford University Press. 70

Levey, A. S., Bosch, J. P., Lewis, J. B., Greene, T., Rogers, N. & Roth, D. (1999). A more accurate method to estimate glomerular filtration rate from serum creatinine: a new prediction equation. modification of diet in renal disease study group. *Annals of Internal Medicine*, *130*, 461-470. 4, 44

Li, W. & Yang, Y. (2002). How many genes are needed for a discriminant microarray data analysis? In S. Lin & K. Johnson (Eds.), *Methods of microarray data*

analysis (p. 137-150). Kluwer Academic. 64, 104, 118

Liepins, G. E., Hilliard, M. R., Palmer, M. R. & Rangarajan, G. (1989). Credit assignment and discovery in classifier systems. *Intern. J. of Intelligent Sys.*, *6*(1), 55-69. 30

Lin, C.-T. & Lee, C. S. G. (1996). *Neural fuzzy systems: a neuro-fuzzy synergism to intelligent systems*. Prentice-Hall, Inc. 37

Ling, C., , Ling, C. X. & Li, C. (1998). Data mining for direct marketing: Problems and solutions. In *In proceedings of the fourth international conference on knowledge discovery and data mining (kdd-98* (p. 73-79). AAAI Press. 77

Liu, Y., Li, M., Cheung, Y. M., Sham, P. C. & Ng, M. K. (2009). Skm-snp: Snp markers detection method. *Journal of Biomedical Informatics*, *In Press, Corrected Proof*. 167

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE Transactions on Information Theory*, *28*(2), 129-137. 51

Loftus, E. V. (2004). Clinical epidemiology of inflammatory bowel disease: incidence, prevalence, and environmental influences. *Gastroenterology*, *126*(6), 1504 - 1517. 166

Lu, Y. & Han, J. (2003). Cancer classification using gene expression data. *Information Systems*, *28*, 243-268. xvi, 11, 14, 81, 82, 83

Lukashin, A. V. & Fuchs, R. (2001). Analysis of temporal gene expression profiles: clustering by simulated annealing and determining the optimal number of clusters. *Bioinformatics*, *17*(5), 405-414. 51

Luscombe, N. M., Greenbaum, D. & Gerstein, M. (2001). What is bioinformatics? an introduction and overview. In *2001 international medical informatics association yearbook* (p. 83-100). 17

Mandel, S., Weinreb, O. & Youdim, M. (2003). Using cdna microarray to assess parkinson's disease models and the effects of neuroprotective drugs. *Trends Pharmacol Sci.*, *24*(4), 184-91. 17

Maojo, V. (2004). Domain-specific particularities of data mining: Lessons learned. In *Isbmda* (p. 235-242). 2

Marshall, E. (2004). Getting the noise out of gene arrays. *Science*, *306*(5696), 630-631. 20

McCarthy, M. I. & Hirschhorn, J. N. (2008). Genome-wide association studies: potential next steps on a genetic journey. *Human Molecular Genetics*, *17*(R2), R156-R165. 165

Michalewicz, Z. & Fogel, D. B. (2004). *How to solve it: Modern heuristics*. Springer. 25

Minkel, J. (2006, 12-October). *Tiny genome may reflect organelle in the making.* `http://www.scientificamerican.com/article.cfm?id=tiny-genome-may-reflect-o`. 11

Mitchell, M. (1996). *An introduction to genetic algorithms*. MIT Press. 27, 28, 30

Mitchell, T., Keller, R. & Kedar-Cabelli, S. (1986). Explanation-based generalization: A unifying view. *Machine Learning*, *1*(1), 47-80. 54

Muhlenbein, H., Bendisch, J. & Voigt, H. (1996). From recombination of genes to the estimation of distributions i. binary parameters. In (pp. 178–187). Springer-Verlag. 30

Nevins, J. R., Huang, E. S., Dressman, H., Pittman, J., Huang, A. T. & West, M. (2003). Towards integrated clinico-genomic models for personalized medicine: combining gene expression signatures and clinical factors in breast cancer outcomes prediction. *Human Molecular Genetics*, *12*(2), R153-R157. 98, 165

N.Kasabov, Middlemiss, M. & Lane, T. (2003). A generic connectionist-based method for on-line feature selection and modelling with a case study of gene expression data analysis. In *Conferences in research and practice in information technology series: Proceedings of the first asia-pacific bioinformatics conference on bioinformatics 2003* (Vol. 33, p. 199-202). Dallinghust, Australia: Australian Computer Society, Inc. 67

Pang, S., Havukala, I., Hu, Y. & Kasabov, N. (2007). Classification consistency analysis for bootstrapping gene selection. *Neural Computing and Applications*, *16*(6), 527-539. 74

Pang, S., Havukkala, I., Hu, Y. & Kasabov, N. (2008). Bootstrapping consistency method for optimal gene selection from microarray gene expression data for classification problems. In Y.-Q. Zhang & J. C. Rajapakse (Eds.), *Machine learning for bioinformatics* (p. 89-111). New Jersey: John Wiley & Sons, Inc. 146

Pang, S. & Kasabov, N. (2004). Inductive vs transductive inference, global vs local models: Svm, tsvm, and svmt for gene expression classification problems. In *Neural networks, 2004 ieee international joint conference* (Vol. 2, p. 1197-1202). 45

Park, J., Hwang, S., Lee, Y. S., Kim, S.-C. & Lee, D. (2007). SNP@Ethnos: a database of ethnically variant single-nucleotide polymorphisms. *Nucl. Acids*

Res., *35*(suppl$_1$), D711-D715. 167

Pawitan, Y., Bjohle, J., Amler, L., Borg, A. L., Egyhazi, S., Hall, P. et al. (2005). Gene expression profiling spares early breast cancer patients from adjuvant therapy: derived and validated in two population-based cohorts. *Breast Cancer Res*, *7*, R953 - 964. 64

Pearson, H. (2006). Genetics: What is a gene? *Nature*, *441*(7092), 398-401. 12

Peng, Y., Li, W. & Liu, Y. (2006). A hybrid approach for biomarker discovery from microarray gene expression data for cancer classification. *Cancer Informatics*, *2*, 301-11. 74

Petricoin, E. F., Ardekani, A. M., Ben A Hitt, P. J. L., Fusaro, V. A., Steinberg, S. M., Mills, G. B. et al. (2002). Use of proteomic patterns in serum to identify ovarian cancer. *Lancet*, *359*, 572-77. xxx, 20, 147, 149, 186

Pomeroy, S. L., Tamayo, P., Gaasenbeek, M., Sturla, L. M., Angelo, M., McLaughlin, M. E. et al. (2002). Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, *415*(6870), 436-442. xxx, 104, 110

Ponulak, F. & Kasiński, A. (2010). Supervised learning in spiking neural networks with resume: Sequence learning, classification, and spike shifting. *Neural Computation*, *22*(2), 467-510. 189

Potter, M. A. & De Jong, K. A. (1994). A cooperative coevolutionary approach to function optimization. In *the third parallel problem solving from nature* (p. 249-257). Springer-Verlag. 139, 141

Potter, M. A. & De Jong, K. A. (2000). Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary Computation*, *8*(1), 1-29. 140, 141

Qiu, X., Xiao, Y., Gordon, A. & Yakovlev, A. (2006). Assessing stability of gene selection in microarray data analysis. *BMC Bioinformatics*, *7*(50). 86

Ramaswamy, S. & Perou, C. (2003). Dna microarrays in breast cancer: the promise of personalised medicine. *Lancet*, *361*(9369), 1590-96. 85

Ransohoff, D. F. (2004). Rules of evidence for cancer molecular marker discovery and validation. *Nature Reviews Cancer*, *4*, 309-314. 85, 86

Ransohoff, D. F. (2005a). Bias as a threat to the validity of cancer molecular-marker research. *Nat Rev Cancer*, *5*(2), 142-9. 20

Ransohoff, D. F. (2005b). Lessons from controversy: Ovarian cancer screening and serum proteomics. *Journal of National Cancer Institute*, *97*(4), 315-319. 20

Rastegar, R. & Hariri, A. (2006). The population-based incremental learning al-

gorithm converges to local optima. *Neurocomputing*, *69*(13-15), 1772 - 1775. 188

Raudys, S. (1976). On dimensionality, learning sample size and complexity of classification algorithms. In *Third int. conf. pattern recognition* (p. 166-169). San Diego, USA. 65

Rechenberg, I. (1973). *Evolutions strategie - optimierung technischer system nach prinzipien der biologischen evolution*. Stuttgart: Fromman-Holzboog. 30

Reuters. (2009, 6th January). *Accelerate progress remembers the 500,000 americans who died of cancer in 2008 and launches its plan for faster progress in their memory*. `http://www.reuters.com/article/pressRelease/idUS138157+06-Jan-2009+BW20090106`. 18

Robert, M., Holte, R. C., Acker, L. E. & Porter, B. W. (1989). Concept learning and the problem of small disjuncts. In *In proceedings of the eleventh international joint conference on artificial intelligence* (p. 813-818). Morgan Kaufmann. 76

Roger Jang, J. shing. (1993). Anfis: Adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, *23*, 665–685. 37

Russell, P. J. (2009). *igenetics: A molecular approach*. Benjamin Cummings. 12, 14

Saeys, Y., Inza, I. & Larranaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics*, *23*(19), 2507-2517. 75

Sartor, R. B. (1997). Pathogenesis and immune mechanisms of chronic inflammatory bowel diseases. *Am J Gastroenterol*, *92*(12 Suppl), 5S-11S. 165

Schena, M. (Ed.). (2000). *Microarray biochip technology*. Natick, MA.: Eaton Publishing Co. 15

Schena, M. (2002). *Microarray analysis*. New York: John Wiley & Sons. 20

Schwefel, H.-P. (1974). *Numerische optimierung von computer-modellen*. Phd, Technical University of Berlin. 30

Schwefel, H.-P. (1981). *Numerical optimization of computer models*. Chichester: Wiley. 30

Schwefel, H.-P. (1987). Collective phenomena in evolutionary systems. In P. Checkland & I. Kiss (Eds.), *the 31st annual meeting of the intl soc. for general system research* (Vol. 2, p. 1025-33). Budapest. 32

Schwefel, H.-P. (1995). *Evolution and optimum seeking*. New York: Wiley Interscience. 30

Senate Health, Education, Labor, and Pensions. (2007). *A bill to secure the promise of personalized medicine for all americans by expanding and accelerating ge-*

nomics research and initiatives to improve the accuracy of disease diagnosis, increase the safety of drugs, and identify novel treatments. The Library of Congress. 5

Shabo, A. (2007). Health record banks: integrating clinical and genomic data into patient-centric longitudinal and cross-institutional health records. *Personalised Medicine*, *4*(4), 453-455. 4

Shah, A. R., Oehmen, C. S. & Webb-Robertson, B.-J. (2008). Svm-hustle-an iterative semi-supervised machine learning approach for pairwise protein remote homology detection. *Bioinformatics*, *24*(6), 783-790. 39

Shastry, B. (2006). Pharmacogenetics and the concept of individualized medicine. *Pharmacogenetics*, *6*(1), 16-21. 98

Shawe-taylor, J. & Cristianini, N. (1999). Further results on the margin distribution. In *In proc. 12th annu. conf. on comput. learning theory* (pp. 278–285). ACM Press. 81

Shi, L., Perkins, R. G., Fang, H. & Tong, W. (2008). Reproducible and reliable microarray results through quality control: good laboratory proficiency and appropriate data analysis practices are essential. *Current Opinion in Biotechnology*, *19*(1), 10-18. 21

Shipp, M. A., Ross, K. N., Tamayo, P., Weng, A. P., Kutok, J. L., Aguiar, R. C. T. et al. (2002). Diffuse large b-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nat Med*, *8*(1), 68-74. xxix, 58, 59

Sims, K. (1994). *Evolving 3d morphology and behavior by competition* (R. A. Brooks & P. Meas, Eds.). Cambridge, Massachusetts: MIT Press. 139

Smith, J. M. (1989). *Evolutionary genetics.* Oxford University Press. 140

Smola, A. J., Smola, A. J., Bartlett, P., (Eds.), D. S., Bartlett, P., Sch02lkopf, B. et al. (1999). *Advances in large margin classifiers.* MIT Press. 81

Snedecor, G. W. & Cochran, W. G. (1989). *Statistical methods.* Iowa State University Press. 70

Solomatine, D. P. (1998). Genetic and other global optimization algorithms – comparison and use in calibration problems. In *Proc., 3rd int. conf. on hydroinformatics, balkema* (p. 1021-1027). 162

Solomonoff, R. (1964a). A formal theory of inductive inference, part i. *Information and Control, Part I*, *7*(1), 1-22. 44

Solomonoff, R. (1964b). A formal theory of inductive inference, part ii. *Information*

and Control, *7*(2), 224-254. 44

Song, Q. & Kasabov, N. (2004). Twrbf: Transductive rbf neural network with weighted data normalization. *Lecture Notes in Computer Science*, *3316*, 633-640. 45, 98

Song, Q. & Kasabov, N. (2006). Twnfi - a transductive neuro-fuzzy inference system with weighted data normalization for personalized modeling. *Neural Networks*, *19*(10), 1591 56. xv, 5, 45, 98, 157, 158, 218, 219

Sun, L., Miao, D. & Zhang, H. (2008). Efficient gene selection with rough sets from gene expression data. In *Rough sets and knowledge technology* (Vol. 5009/2008, p. 164-171). Berlin: Springer. 67

Sun, Y. (2006). Boosting for learning multiple classes with imbalanced class distribution. In *In 2006 ieee international conference on data mining (accepted), hongkong.* 76

Sureka, A. & Indukuri, K. V. (2008). Using genetic algorithms for parameter optimization in building predictive data mining models. In *Adma* (p. 260-271). 117

Suykens, J. A. K. & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters*, *9*(3), 293-300. 41

Swets, J. (1988). Measuring the accuracy of diagnostic systems. *Science*, *240*(4857), 1285-1293. 90

Swets, J. A., Dawes, R. M. & Monahan, J. (2000, October). Better decisions through science. *Scientific American*, *283*(4), 82–87. 90

Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E. et al. (1999). Interpreting patterns of gene expression with self-organizing maps. *P.N.A.S.*, *96*(6), 2907-2912. 71

Tamura, K. & Fukuoka, M. (2005). Gefitinib in non-small cell lung cancer. *Expert Opin. Pharmacother*, *6*(6), 985-993. 19

Tibshirani, R. (2006). A simple method for assessing sample sizes in microarray experiments. *BMC Bioinformatics*, *7*(106). 67

Turing, A. (1950). Computing machinery and intelligence. *Mind*, *59*, 433-460. 25

Tusher, V., Tibshirani, R. & Ghu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *Proc Natl Acad Sci U.S.A.*, *98*(9), 5116-21. 67

U.S. the National Library of Medicine. (2009). *What is dna?* `http://ghr.nlm.nih.gov/handbook/basics/dna`. x, 12

Vapnik, V. N. (1998). *Statistical learning theory.* New York: Wiley. 5, 39, 44

Varma, S. & Simon, R. (2006). Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics*, *7*(91). 21, 64

Veer, L. J. van't, Dai, H., Vijver, M. J. van de, He, Y. D., Hart, A. A. M., Mao, M. et al. (2002). Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, *415*(6871), 530-536. 64, 71

Vermeire, S., Van Assche, G. & Rutgeerts, P. (2007). Review article: altering the natural history of crohn's disease. *Alimentary Pharmacology & Therapeutics*, *25*(1), 3-12. 165

Wang, D. G., Fan, J.-B., Siao, C.-J., Berno, A., Young, P., Sapolsky, R. et al. (1998). Large-Scale Identification, Mapping, and Genotyping of Single-Nucleotide Polymorphisms in the Human Genome. *Science*, *280*(5366), 1077-1082. 22

Wasson, J., Sox, H., Neff, R. & Goldman, L. (1985). Clinical prediction rules. applications and methodological standards. *N Engl J Med*, *313*(13), 793-799. 2

Welch, B. (1938). The significance of the difference between two means when the population variances are unequal. *Biometrika*, *29*, 350-362. 70

West, M., Blanchette, C., Dressman, H., Huang, E., Ishida, S., Spang, R. et al. (2001). Predicting the clinical status of human breast cancer by using gene expression profiles. *Proceedings of the National Academy of Sciences of the United States of America*, *98*(20), 11462-11467. 45

Wiegand, R. P. (2003a). *An analysis of cooperative coevolutionary algorithms.* Unpublished doctoral dissertation, George Mason University. 138, 142

Wiegand, R. P. (2003b). *An analysis of cooperative coevolutionary algorithms.* Unpublished doctoral dissertation, George Mason University. 139

wikipedia. (2009, 20-Nov-2009). *Human genome.* `http://en.wikipedia.org/wiki/Human_genome`. 11

Wolf, L., Shashua, A. & Mukherjee, S. (2004). *Selecting relevant genes with a spectral approach* (Tech. Rep. No. CBCL Paper No.238). Massachusetts Institute of Technology. 65, 67

WTCCC. (2007). Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature*, *447*(7145), 661-678. 167

Wu, D., Bennett, K. P., Cristianini, N. & Shawe-taylor, J. (1999). *Large margin trees for induction and transduction.* 45

Wu, Q. (2009). The forecasting model based on wavelet -support vector machine. *Expert Systems with Applications: An International Journal*, *36*(4), 7604-7610. 39

Zadeh, L. A. (1988). Fuzzy logic. *IEEE Computer*, *21*(4), 83-93. 157

Zhang, C., Lu, X. & Zhang, X. (2006). Significance of gene ranking for classification of microarray samples. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *3*(3), 312-320. 67

Zhang, M., Yao, C., Guo, Z., Zou, J., Zhang, L., Xiao, H. et al. (2008). Apparently low reproducibility of true differential expression discoveries in microarray studies. *Bioinformatics*, *24*(18), 2057-2063. 21

Zhu, W., Wang, X., Ma, Y., Rao, M., Glimm, J. & Kovach, J. S. (2003). Detection of cancer-specific markers amid massive mass spectral data. *PNAS*, *100*, 14666-671. 20, 85

APPENDIX A

sGA - the Pseudo Code of a Simple Genetic Algorithm

Algorithm 5 Pseudo code for a classical GA

1: $gen = 0$;
2: $P(gen) = F_p(\mu)$; {creates a random population}
3: $fitness(gen) = F_{evl}(P(gen))$; {$F_{evl}$ is a fitness function}
4: **while** $fitness(gen) < \zeta$ **do**
5: gen++;
6: $S(gen) = F_{sel}(P(gen-1), \mu/2)$; {select $\mu/2$ pairs of fittest individuals}
7: $O(gen) = crossover(S(gen), \mu/2, p_c)$; {perform crossover on the $\mu/2$ pairs}
8: $O(gen) = mutate(O(gen), p_m)$; {perform mutation}
9: $P(gen) = S(gen) + O(gen)$; {form a new generation}
10: $fitness(gen) = F_{evl}(P(gen))$;
11: **end while**

Note:

ζ:	the desired optimal level;
μ:	population size(the number of individuals in each generation);
p_c:	the crossover probability(e.g.,0.7);
p_m:	the mutation probability(e.g.,0.001);

APPENDIX B

Pseudo Code of a Simple Evolutionary Strategy Algorithm

Algorithm 6 A simple evolutionary strategy algorithm

1: Initialization: randomly generate a parent population with μ individuals $\mathbf{P}_\mu = \{\mathbf{a}_1, \ldots, \mathbf{a}_\mu\}$.

2: Generate λ offsprings $\tilde{\alpha}$ to form an offspring population $\tilde{\mathbf{P}}_\lambda = \{\tilde{\alpha}_1, \ldots, \tilde{\alpha}_\lambda\}$, where each offspring $\tilde{\alpha}_\mathbf{i}$ is generated by:

(1).Randomly select ρ parents($\mathbf{P}_\rho$) from $\mathbf{P}_\mu$ (if $\rho = \mu$ all parental individuals are selected).

(2).Recombine the selected parents $\mathbf{P}_\rho$ to form a new offspring population ß$_o$.

(3).Mutate the *endogenous strategy parameters* **s**.

(4).Mutate the objective parameter set **y** of ß$_o$ using the mutated *endogenous strategy parameters*

3: Each individual in ß$_o$ is evaluated by a fitness function F

4: Select new parent population $\mathbf{P}^*_\mu$ using either:

(1) comma selection - (μ, λ) from the selection pool of offspring population $\tilde{\mathbf{P}}_\lambda$, or

(2) plus selection - $(\mu + \lambda)$ from the selection pool of offspring $\tilde{\mathbf{P}}_\lambda$ and parent $\mathbf{P}_\mu$ population.

5: The new population $\mathbf{P}^*_\mu$ becomes the current population $\mathbf{P}_\mu(gen + 1)$.

6: Terminate if the stopping criterion is fulfilled, otherwise go to step **2**.

APPENDIX C

Pseudo Code of a Compact Genetic Algorithm (cGA)

Algorithm 7 A compact genetic algorithm (cGA)

1: Initialization: generate a probability vector p
   $p(i) = 0.5,\ i = 1, \ldots, l$;
2: Generate two individuals a and b based on the comparison with p:
   $a(gen) =$ **generate**(p);
   $b(gen) =$ **generate**(p);
3: Competition between a and b:
   $winner, loser =$ **compete**(a, b)
4: Update the probability vector p towards the winner:
   if $winner(i) \neq loser(i)$ then
      if $winner(i) == 1 \mid i = 1 : l$
         then $p(i) = p(i) + \frac{1}{\mu}$;
         else $p(i) = p(i) - \frac{1}{\mu}$;
      end
   end
5: Check whether the probability vector p has converged:
   if **no** then go to step 2;
6: p is the *optimal* solution;

APPENDIX D

EFuNN - Evolving Fuzzy Neural Networks

The algorithm for evolving EFuNNs from incoming data vectors can be described as follows:

1. Initialization: create an EFuNN structure with maximum number of neurons without any connections. If no rule nodes exists, then create the first node $r_i = 1$ to represent the first data vector X_1 and assign its connection weight vectors of input weight vector $\omega_1(r_i)$ and output $\omega_2(r_i)$ as follows:

$$\begin{aligned} \omega_1(r_i) &= \mathbb{EX} \\ \omega_2(r_i) &= \mathbb{TE} \end{aligned} \tag{D.1}$$

   where $\mathbb{EX}$ is the the fuzzy input vector of the current data vector X_i, and $\mathbb{TE}$ denotes the fuzzy output vector X_i.

2. ***if*** new variables from incoming data vectors appear in the current data vector (X_i) and are absent in the previous data, ***then*** create new input and/or output nodes with their corresponding membership functions.

3. Compute the normalised fuzzy local distance between the fuzzy input vector $\mathbb{EX}$ and the stored patterns (prototypes) in the rule (case) nodes $r_j (j = 1, 2, \cdots, n)$ as follows:

$$d(\mathbb{EX}, r_j) = \frac{\sum \frac{|\mathbb{EX} - \omega_1(j)|}{2}}{\sum \omega_1(j)} \tag{D.2}$$

where d is the distance.

4. Find the activation $A_1(r_j)$ of the rule r_j, $j = 1, 2, ...n$. $A_1(r_j)$ can be calculated through two ways: radial basis (f_{radbas}) or a saturated linear (f_{satlin}) function:

$$\begin{aligned} A_1(r_j) &= f_{radbas}(d(\mathbb{EX}, r_j)), or \\ A_1(r_j) &= f_{satlin}(1 - d(\mathbb{EX}, r_j)) \end{aligned} \tag{D.3}$$

   The former is more appropriate for function approximate tasks, while the latter is usually used for classification tasks (Kasabov, 2002).

5. Update the pruning parameter value for the rule nodes which are pre-specified in EFuNN neurons.

6. Find all rule nodes r_j with an activation value $A_1(r_j)$ greater than a sensitivity threshold θ_s.

7. ***if*** no such rule nodes exists, ***then*** create a new rule node from step 1.
   else , find the rule node r_{max} with the maximum activation value γ_{maxa1}.

8. Two modes of EFuNNs:
   (1) *one-of-n* mode: propagate the maximum activation value of the rule node r_{max} to the fuzzy output neurons:

$$A_2 = f_{satlin}(A_1(r_{max}) * \omega_2(r_{max})) \tag{D.4}$$

   (2) *many-of-n mode*: the activation values of all rule nodes that above an activation threshold θ_a are propagated to the next neural layer. Find the winner of fuzzy output neuron r_{max2} with its activation γ_{maxa2}.

9. Find the desired winner fuzzy output neuron r_{maxt2} and its activation γ_{maxt2}.

10. Calculate the fuzzy output error: $\mathbf{Err}_{out} = A_2 - \mathbb{TE}$.

11. ***if*** $r_{max2} <> r_{maxt2} \parallel d(A_2, \mathbb{TE}) > \theta_{err}$, ***then*** go to step 1 to create a new rule node.
    else , update parameters, including A, θ_s, γ_{max}, etc, for rule node r_{max2}.

12. if necessary, prune rule nodes r_j and connections by the following fuzzy rules:
    if a rule node r_j is *OLD* **AND** average activation $A_{1av}(r_j)$ is *LOW* **AND** the density of the neighbourhood of neurons is *HIGH* or *MODERATE*; ***then*** rule node r_j has a high probability to be pruned.
    Here *OLD, MODERATE and HIGH* are pre-defined fuzzy concepts, e.g. a node is

considered *OLD* if it has existed during an EFuNN evolving process for more than 500 samples.

13. Aggregate rule nodes.

14. Iterate the process from step **2** for a new presentation of the same input data sample.

APPENDIX E

ECF - Evolving Classification Function

Algorithm 8 The algorithm of ECF - a local learning model (Kasabov, 2003)

Learning algorithm of the ECF model:

1: Input a vector from the incoming dataset and calculate the distance between it and all rule nodes already created using a distance measurement function (e.g. Euclidean distance). If all nodes are created, create the first one that has the coordinates of the first input vector attached as input connection weights.

2: ***if*** all calculated distances between the new input vector and the existing rule nodes are greater than a max-radius parameter (R_{max}), a new rule node is created. The position of the new rule node is the same as the current vector in the input data space and the radius of its receptive field is set to the min-radius (R_{min}); the algorithm goes to step 1;
else it goes to the next step.

3: ***if*** there is a rule node with a distance to the current input vector less than or equal to its radius, and its class is the same as the class of the new vector, nothing will be changed; go to Step 1;
otherwise : go to next step.

4: ***if*** there is a rule node with a distance to the input vector less than or equal to its radius and its class is different from those of the input vector, its influence field should be reduced. The radius of the new field is set to the larger of the two numbers: $(distance - R_{min})$ and R_{min}. a New node is created as in step 2 to represent the new data vector.

5: ***if*** there is a rule node with a distance to the input vector less than or equal to the max-radius, and its class is the same as of the input vectors, enlarge the influence field by taking the distance as a new radius if only such enlarged field does not cover any other rule nodes that belong to a different class;
otherwise : create a new rule node in the same way as in step 2, and go to step 2.

Recall procedure (classification of a new input vector) in a trained ECF:

6: Input a new vector in the ECF trained system. If the new input vector exists within the field of one or more rule nodes associated with one class, the vector is classified in this class;

7: If the input vector is within the fields of two or more rule nodes associated with different classes, the vector should belong to the class corresponding to the closest rule node.

8: If the input vector does not lie within any field, then take m highest activated by the new vector rule nodes, and calculate the average distances from the vector to the nodes with the same class; the vector is assigned to the class corresponding to the smallest average distance.

APPENDIX F

TWNFI - a Transductive Neuro-fuzzy Inference System with Weighted Data Normalisation for Personalised Modelling

F.1 The Principle of TWNFI

TWNFI (Song & Kasabov, 2006) is a dynamic neuro-fuzzy inference system in which a local model is created for analysing each new data vector x_v. A basic block diagram of TWNFI is illustrated in Figure F.1.

Giving a training dataset X, for each new data vector x_v, TWNFI creates a unique model with the application of the following steps (Song & Kasabov, 2006):

1. Normalisation:
   - Normalise the training data X and the new data vector x_v (values range from 0 to 1);
   - Initialise the weights of every input variables (features) to 1;
2. Identifying an appropriate neighbourhood (D_v) for x_v;
   Find N_v samples from training data that are closest to x_v based on the weighted

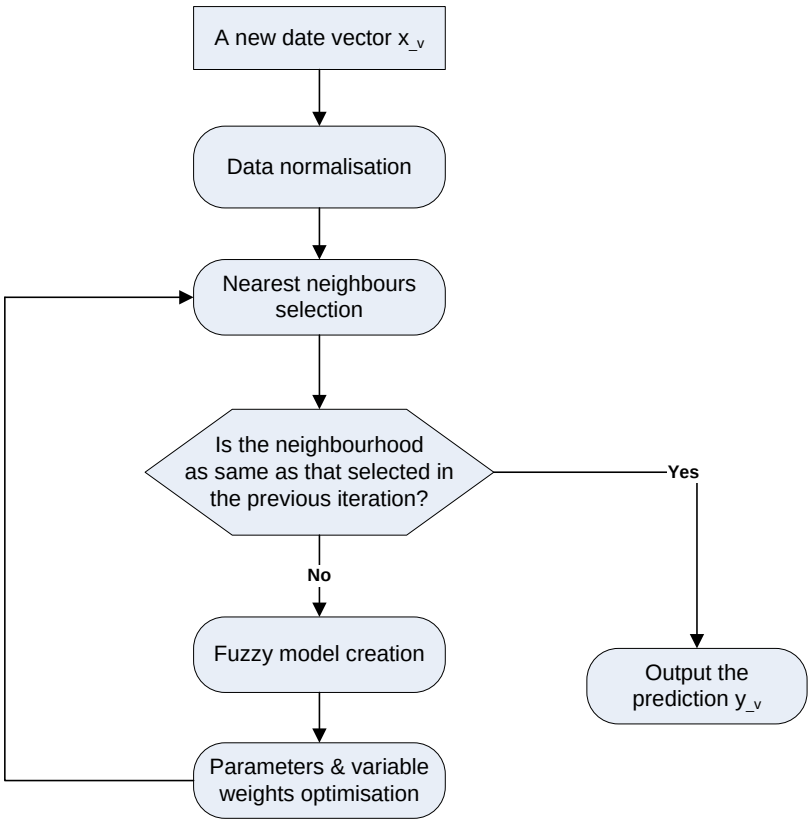


Figure F.1: *A basic block diagram of TWNFI, adapted from (Song & Kasabov, 2006)*

normalised Euclidean distance calculated as:

$$\|x - y\| = \sqrt{\frac{\sum_{j=1}^{P} w_j (x_j - y_j)^2}{P}} \tag{F.1}$$

where x_j and y_j are two vectors in the given problem space, N is the number of samples, and w is a weight vector.

3. Calculate the distance $d_i, i = 1, \cdots, N_v$ using Eq.F.1. d_i is the distance between each sample in D_q and x_v. Each sample's weight is calculated as:
   $v_i = 1 - (d_i - min(d)), i = 1, 2, \cdots, N_v,$
   where $min(d)$ is the minimum number of elements in the distance vector $d = [d_1, d_2, \cdots, d_{N_v}]$;

4. Cluster and partition the input subspace that consists of N_v selected training sam-

ples; Create fuzzy rules and set their initial parameter values based on the clustering results. Every fuzzy rule is created as:
the centroid of a cluster is the center of the fuzzy membership function (e.g. a Gaussian membership function) and the cluster radius is taken as the width;

5. Apply the steepest descent approach (back-propagation) to optimise the weights and the parameters of the fuzzy rules in a local model M_v;

6. Find a new set of N_v samples (D_v^*) nearest to x_v (*Step 2*):
if the same samples are found as in the last search, the algorithm goes to the next step;
otherwise, it repeats from *Step 3*.

7. Output the prediction y_v for the new data vector x_v using fuzzy inference on the set of fuzzy rules that constitute the local model M_v;

The weight and parameters can be optimised as follows: Consider a system having P inputs, one output and M fuzzy rules initially defined by a clustering algorithm, and the l^{th} rule is formed as:
R_l: if x_l is F_{l1} and x_2 is F_{l2} and $\cdots$ x_p is F_{lp}, then y is G_l,
where F_{lj} are the fuzzy sets defined by the following Gaussian membership function:

$$Gaussian\ \ MF = \alpha \exp(-\frac{(x-m)^2}{2\sigma^2}) \tag{F.2}$$

and G_l can be defined as:

$$Gaussian\ \ MF = \exp(-\frac{(y-n)^2}{2\delta^2}) \tag{F.3}$$

Thus, the output of the system for an input vector $x_i = [x_1, x_2, \cdots, x_p]$ can be calculated by a modified centre average defuzzification function as:

$$f(x_i) = \frac{\sum_{l=1}^{M} \frac{n_l}{\delta_l^2} \prod_{j=1}^{P} \alpha_{lj} \exp[-\frac{w_j^2 (x_{ij} - mlj)^2}{2\sigma_{lj}^2}]}{\sum_{l=1}^{M} \frac{1}{\delta_l^2} \prod_{j=1}^{P} \alpha_{lj} \exp[-\frac{w_j^2 (x_{ij} - mlj)^2}{2\sigma_{lj}^2}]} \tag{F.4}$$

where, w_j is the current weight vector for the input variables and n_l is the point having maximum membership value in the l^{th} output set.

APPENDIX G

Experimental results obtained using iPM with WKNN classifier for colon cancer gene data

The result obtained by iPM with WKNN classifier over colon cancer gene data

---------- Data: colon.txt ----------
========== Result Part 1 ==========
Num of samples: 62 Num of features: 2000
Classification threshold : 0.50

*** Overall Accuracy of LOO Cross validation: **75.81%** ***
Class 1 Overall Accuracy: 85.00%
Class 2 Overall Accuracy: 59.09%

========== Result Part 2 ==========

Sample ID	num of Features	output	predicted	actual	local accuracy	
1	11	1.08	1	1	95.24%	
2	3	1.78	2	2	81.48%	
3	43	1.08	1	1	92.43%	
4	10	1.93	2	2	91.53%	
5	4	1.00	1	1	93.60%	
6	7	1.31	1	1	91.42%	
7	3	1.85	2	2	80.77%	
8	28	1.15	1	1	91.42%	
9	14	1.16	1	2	91.70%	*
10	3	1.16	1	1	91.32%	
11	24	1.16	1	2	91.40%	*
12	6	1.85	2	2	93.20%	
13	95	1.15	1	1	95.35%	
14	5	1.08	1	1	91.32%	
15	5	1.15	1	1	90.91%	
16	3	1.00	1	1	81.82%	
17	18	1.00	1	1	95.16%	
18	16	1.08	1	1	91.14%	
19	10	1.15	1	1	91.86%	
20	3	1.00	1	1	81.82%	
21	20	1.00	1	1	94.22%	
22	3	1.15	1	1	86.36%	
23	3	1.23	1	2	92.00%	*
24	15	1.00	1	1	95.26%	
25	14	1.00	1	1	95.29%	
26	12	1.93	2	1	90.91%	*
27	3	1.08	1	2	91.72%	*
28	15	1.00	1	1	94.33%	
29	3	1.85	2	1	90.91%	*
30	3	1.69	2	1	92.00%	*
31	4	1.63	2	2	90.82%	
32	4	1.24	1	2	91.45%	*
33	7	1.62	2	1	90.97%	*
34	3	1.08	1	2	87.50%	*
35	5	1.31	1	1	90.80%	
36	4	1.85	2	2	90.87%	
37	3	1.00	1	1	91.50%	
38	3	1.85	2	2	86.36%	
39	3	1.31	1	1	86.96%	
40	7	1.62	2	2	91.01%	
41	5	1.08	1	1	92.95%	
42	4	1.08	1	2	91.96%	*
43	4	1.00	1	1	91.27%	
44	4	1.23	1	1	92.23%	
45	5	1.08	1	1	90.95%	
46	3	1.47	1	1	94.21%	
47	6	1.00	1	1	92.36%	
48	3	1.08	1	1	92.76%	
49	7	1.77	2	2	91.34%	
50	3	1.39	1	1	81.82%	
51	17	1.77	2	2	92.12%	
52	50	1.93	2	1	95.77%	*

53	83	1.00	1	1	91.33%	
54	43	1.62	2	2	93.49%	
55	4	1.08	1	2	90.95%	*
56	4	1.00	1	1	92.25%	
57	4	1.70	2	2	91.24%	
58	19	1.00	1	2	92.00%	*
59	4	1.77	2	1	91.76%	*
60	3	1.31	1	1	82.61%	
61	4	1.55	2	2	91.52%	
62	21	1.15	1	1	92.30%	

==========

Note: the genes selected less 30 times are removed from the above list

the union of selected features:

1772 1582 249 493 391 964 1406 1423 1648 1067 1153 513 652 1002 1414 1060 1808 1058 399 625 1263 43 1325 765 1042 1771 377 66 1334 1730 1346 1943

the frequency of each feature selected in LOOCV process:

Feature Index	Frequency
1772	47
1582	46
249	44
493	42
391	41
964	41
1406	41
1423	41
1648	41
1067	40
1153	40
513	39
652	39
1002	39
1414	39
1060	38
1808	38
1058	37
399	36
625	36
1263	36
43	35
1325	35
765	34
1042	34
1771	34
377	33
66	32
1334	32
1730	32
1346	31
1943	31

---------- Confusion Table ----------

	Class2	Class1	(Actual Class)
(Predicted Class): Class2	13	6	
(Predicted Class): Class1	9	34	

APPENDIX H

Experimental results obtained using cGAPM for sample 51 of colon cancer gene data

The result of cGAPM method on colon cancer data (sample 51)

---------- Data: colon.txt ----------
Num of training samples: 61 Num of features: 2000
Parameter Setting :
Classification threshold : 0.40
Evaluation model: WKNN

========== Result ==========
Sample: 51

------ **11 neighbours of sample 51** ------
--- Neighbour list:
Sample : 29 31 61 57 26 54 49 6 40 19 32

--- Best local accuracy on training data: 83.82%

*** **24 features are selected**:

Feature Id	EST number	Weighted SNR value
377	Z50753	0.0659
249	M63391	0.0625
765	M76378	0.0555
513	M22382	0.0533
286	H64489	0.0504
1884	R44301	0.0483
1623	T94993	0.0446
625	X12671	0.0442
137	D25217	0.0440
1582	X63629	0.0389
1208	H72965	0.0383
826	H22948	0.0366
1416	M28882	0.0365
1924	H64807	0.0362
1018	M14764	0.0352
1135	R44887	0.0349
689	X73358	0.0347
708	H17969	0.0346
1832	X15943	0.0345
1286	D16294	0.0343
1767	H73943	0.0343
437	H41129	0.0342
961	H91274	0.0342
43	T57619	0.0340

------ **Summary of testing data** ------

Feature ID	Mean Value(Cls1)	Mean Value(Cls2)	Sample 51's Value
377	233.8870	436.2061	686.6330
249	597.1193	2328.7151	1765.1850
765	260.3002	1081.0925	449.3950
513	1142.2057	396.9826	577.2560
286	1225.8794	2198.1646	4474.7640
1884	50.0862	99.6487	66.5650
1623	159.9953	247.2340	238.7980
625	850.5172	336.3976	267.5090
137	550.1411	693.1135	1508.0380
1582	154.7945	57.0313	64.0260
1208	200.6386	87.6992	168.3580
826	295.3520	343.5308	370.3840
1416	135.2856	210.2490	267.0030
1924	66.9099	90.9279	140.9940
1018	267.3680	265.9585	445.8060
1135	142.2807	155.5733	185.5830
689	389.9839	458.5370	451.5690
708	302.4763	299.7358	149.0090

1832	66.9631	76.7587	82.0760
1286	236.4929	304.1668	637.2730
1767	99.3440	117.4830	81.9690
437	453.5585	795.1633	479.3280
961	134.1740	133.2503	229.5440
43	2623.7725	1460.3710	1406.2030

------ **Weighted distance between sample 51 and the average class profile** ------

Cls1	Cls2
470.2062	**301.9498**

------ **testing** ------

sample ID	output	predicted cls	actual cls
51	1.72	2	2

------- A scenario of the improvement for a person ------

Feature ID	Actual value	Desired average profile	Desired Improvement	Weighted importance
Feature_377	686.6330	233.8870	-452.7460	0.0659
Feature_249	1765.1850	597.1193	-1168.0657	0.0625
Feature_765	449.3950	260.3002	-189.0948	0.0555
Feature_513	577.2560	1142.2057	564.9497	0.0533
Feature_286	4474.7640	1225.8794	-3248.8846	0.0504
Feature_1884	66.5650	50.0862	-16.4788	0.0483
Feature_1623	238.7980	159.9953	-78.8027	0.0446
Feature_625	267.5090	850.5172	583.0082	0.0442
Feature_137	1508.0380	550.1411	-957.8969	0.0440
Feature_1582	64.0260	154.7945	90.7685	0.0389
Feature_1208	168.3580	200.6386	32.2806	0.0383
Feature_826	370.3840	295.3520	-75.0320	0.0366
Feature_1416	267.0030	135.2856	-131.7174	0.0365
Feature_1924	140.9940	66.9099	-74.0841	0.0362
Feature_1018	445.8060	267.3680	-178.4380	0.0352
Feature_1135	185.5830	142.2807	-43.3023	0.0349
Feature_689	451.5690	389.9839	-61.5851	0.0347
Feature_708	149.0090	302.4763	153.4673	0.0346
Feature_1832	82.0760	66.9631	-15.1129	0.0345
Feature_1286	637.2730	236.4929	-400.7801	0.0343
Feature_1767	81.9690	99.3440	17.3750	0.0343
Feature_437	479.3280	453.5585	-25.7695	0.0342
Feature_961	229.5440	134.1740	-95.3700	0.0342
Feature_43	1406.2030	2623.7725	1217.5695	0.0340

APPENDIX I

Experiment results obtained using cGAPM for sample 31 of CNS cancer gene data

The experiment result obtained by cGAPM for sample 31 in CNS data

---------- Data: CNS.txt ----------

Num of training samples: **60** Num of features: **7129**

Parameter Setting :

Classification threshold : 0.50

Classification model: FuzzyKNN

========== Result ==========

Sample: 31

------ **21 neighbours of sample 31** ------

--- Neighbour list:

Sample : 48 21 20 43 26 29 41 39 8 28 45 27 30 50 7 24 13 18 54 47 53

--- Best local accuracy on training data: **98.54%**

*** **23 features are selected**:

Feature Idx	Weighted SNR value
3469	0.0550
245	0.0546
7033	0.0527
1988	0.0479
2593	0.0471
4799	0.0453
942	0.0445
4348	0.0443
5396	0.0438
1926	0.0429
6983	0.0416
5709	0.0407
786	0.0406
4214	0.0406
2380	0.0405
1370	0.0403
1462	0.0401
360	0.0398
2316	0.0398
3420	0.0396
540	0.0394
1683	0.0393
4936	0.0393

------ **Summary of testing data** ------

Feature ID	Mean Value(Cls1)	Mean Value(Cls2)	Sample 31's Value
3469	169.6667	229.9211	246.0000
245	-10.7619	-8.1053	36.0000
7033	2842.1429	2911.1316	2259.0000
1988	490.1905	871.1579	914.0000
2593	525.4286	423.6579	-383.0000
4799	-5.1429	29.8684	5.0000
942	3773.3810	4695.2632	1907.0000
4348	-5.3333	110.6316	21.0000
5396	-2.8095	70.8158	-10.0000
1926	81.6667	101.6316	162.0000
6983	142.5714	227.0789	520.0000
5709	22.2857	3.7368	204.0000
786	-21.8571	-39.8421	-168.0000
4214	263.0476	217.2105	47.0000

2380	-87.1905	-42.7895	-1113.0000
1370	4448.8095	3816.6316	3121.0000
1462	215.5238	335.5526	514.0000
360	117.1429	258.5526	231.0000
2316	156.6190	222.0000	851.0000
3420	87.3333	7.1053	236.0000
540	168.7619	159.9737	168.0000
1683	1414.9048	1830.0263	809.0000
4936	2929.2381	3165.1316	3310.0000

------ **Weighted distance between sample 31 and the average class profile** ------

Cls1	Cls2
410.9195	**405.5403**

------ **testing** ------

sample ID	output risk	predicted cls	actual cls
31	0.69	2	2

------- **A scenario of the improvement for a person** ------

Feature ID	Actual value	Desired average profile	Desired Improvement	Weighted importance
Feature_3469	246.0000	229.9211	-16.0789	0.0550
Feature_245	36.0000	-8.1053	-44.1053	0.0546
Feature_7033	2259.0000	2911.1316	652.1316	0.0527
Feature_1988	914.0000	871.1579	-42.8421	0.0479
Feature_2593	-383.0000	423.6579	806.6579	0.0471
Feature_4799	5.0000	29.8684	24.8684	0.0453
Feature_942	1907.0000	4695.2632	2788.2630	0.0445
Feature_4348	21.0000	110.6316	89.6316	0.0443
Feature_5396	-10.0000	70.8158	80.8158	0.0438
Feature_1926	162.0000	101.6316	-60.3684	0.0429
Feature_6983	520.0000	227.0789	-292.9211	0.0416
Feature_5709	204.0000	3.7368	-200.2630	0.0407
Feature_786	-168.0000	-39.8421	128.1579	0.0406
Feature_4214	47.0000	217.2105	170.2105	0.0406
Feature_2380	-1113.0000	-42.7895	1070.2105	0.0405
Feature_1370	3121.0000	3816.6316	695.6316	0.0403
Feature_1462	514.0000	335.5526	-178.4474	0.0401
Feature_360	231.0000	258.5526	27.5526	0.0398
Feature_2316	851.0000	222.0000	-629.0000	0.0398
Feature_3420	236.0000	7.1053	-228.8947	0.0396
Feature_540	168.0000	159.9737	-8.0263	0.0394
Feature_1683	809.0000	1830.0263	1021.0263	0.0393
Feature_4936	3310.0000	3165.1316	-144.8684	0.0393

APPENDIX J

Experimental results obtained using cEAP on colon cancer gene data through LOOCV

Table J.1: *The experiment result obtained by cEAP on colon cancer gene data through LOOCV*

Sample ID	No. of selected features	No. of selected neighbours	Local accuracy (%)	Outcome	Predict (T - Correct; F - Wrong)
1	29	12	84.27	1.08	T
2	23	8	88.28	1.75	T
3	28	16	85.14	1.12	T
4	29	7	84.03	1.86	T
5	29	6	90.46	1.00	T
6	24	16	94.20	1.06	T
7	38	19	87.49	1.74	T
8	37	5	86.42	1.21	T
9	29	7	84.69	1.57	T
10	31	10	83.46	1.20	T
11	32	6	83.78	1.51	T
12	14	19	85.93	1.64	T
13	29	21	89.45	1.24	T
14*	23	4	89.13	1.25	F
15	19	3	90.05	1	T
16	25	10	87.15	1	T
17	17	12	83.84	1.33	T
18	26	5	88.73	1	T
19	29	4	94.70	1	T
20	22	21	88.19	1.14	T
21	21	12	88.44	1	T
22	27	8	85.24	1	T
23	26	7	86.90	1.85	T
24	31	26	88.23	1.07	T
25	21	21	87.15	1	T
26*	35	3	90.79	1.66	F
27*	27	12	90.65	1	F
28	24	15	82.66	1.07	T
29*	25	8	92.37	1.75	F
30	30	8	83.28	1.12	T
31	28	6	90.35	1.66	T
32	30	3	86.47	2	T
33	23	5	95.72	1.81	T
34	36	22	87.80	1.31	T
35	34	8	91.71	1	T
36	31	5	91.60	1.8	T
37	29	24	88.54	1.04	T
38	23	10	86.81	1.9	T
39	27	20	88.68	1.15	T
40	29	7	86.23	1.86	T
41	27	21	83.45	1.14	T
42*	29	5	85.82	1.2	F
43	13	9	89.00	1.11	T
44	29	7	91.29	1	T
45	18	17	87.30	1	T
46	21	7	88.59	1.14	T
47	26	19	90.77	1.1	T
48	22	16	87.16	1.12	T
49	24	13	86.87	1.55	T
50	27	4	87.84	1	T
51	18	6	86.04	1.49	T
52*	20	5	87.54	2	F
53	20	10	87.96	1	T
54*	26	8	89.60	1.52	F
55	29	3	85.75	2	T
56	19	13	91.08	1.15	T
57	24	13	83.20	1.85	T
58*	24	6	91.75	1	F
59	28	4	85.52	1	T
60	25	10	92.50	1	T
61	23	18	85.21	1.45	T
62	30	19	86.60	1.11	T

APPENDIX K

Experimental results obtained using cEAP for sample 57 of colon cancer data

The experiment result obtained by cEAP for colon sample 57

---------- Data: colonc.txt ----------

Num of training samples: 61 Num of features: 2000

Parameter Setting:

Classification threshold: **0.55**

Classification function: **WKNN**

========== Result ==========

Sample: 57

------ **24 neighbors** of sample 57 ------

--- Neighbor list:

Sample : 51 31 28 55 8 32 49 14 47 61 12 29 54 22 27 30 59 6 15 1 38 26 36 41

--- Best local accuracy on training data: **82.58%**

*** **11 features are selected**:

Feature Idx	Weighted SNR value
249	0.1241
377	0.1218
267	0.0970
419	0.0942
1674	0.0914
548	0.0903
1982	0.0854
1582	0.0797
662	0.0745
1870	0.0735
43	0.0681

------ Summary of testing data ------

Feature ID	Mean Value(Cls1)	Mean Value(Cls2)	Sample 57's Value
249	597.1193	2263.4888	411.6240
377	233.8870	451.8635	179.9090
267	490.9205	1258.6685	397.7460
419	249.8221	351.9843	1370.3900
1674	56.9415	103.2970	98.2440
548	288.2512	371.3105	717.0060
1982	43.2651	57.9870	215.9140
1582	154.7945	59.4295	151.1990
662	428.0565	756.0929	262.8410
1870	142.6591	73.8150	90.0480
43	2623.7725	1432.3778	2997.3980

------ testing ------

sample ID	output	predicted cls	actual cls
57	1.65	2	2

------- A scenario of the potential improvement for a person ------

Feature ID	Actual value	Desired average profile	Desired Improvement	Weighted importance
Feature_249	411.6240	597.1193	185.4953	0.1241
Feature_377	179.9090	233.8870	53.9780	0.1218
Feature_267	397.7460	490.9205	93.1746	0.0970
Feature_419	1370.3900	249.8221	-1120.5679	0.0942
Feature_1674	98.2440	56.9415	-41.3025	0.0914
Feature_548	717.0060	288.2512	-428.7548	0.0903
Feature_1982	215.9140	43.2651	-172.6489	0.0854
Feature_1582	151.1990	154.7945	3.5955	0.0797
Feature_662	262.8410	428.0565	165.2155	0.0745
Feature_1870	90.0480	142.6591	52.6111	0.0735
Feature_43	2997.3980	2623.7725	-373.6255	0.0681

APPENDIX L

Experiment results for CD risk evaluation using SNPs testing data C

=== SNPs data for CD classification (106 samples) ===

Global SVM modelling:
Number of testing samples: 106
Overall Acc: 0.70
Class 1 Acc: 0.63 Class 2 Acc: 0.75

Personalized modelling (feature selection, parameter optimization)
Number of testing samples: 106
Sample Id: 52 98 266 243 186 16 112 83 432 352 433 381 457 447 168 336 166 224 258 219 206 218 163 216 307 422 338 261 401 263 109 386 365 359 214 329 108 129 136 424 170 20 282 230 43 65 79 385 360 708 533 752 915 982 563 855 918 738 832 961 956 989 838 958 886 803 873 919 682 525 1010 950 667 503 583 1000 798 744 912 979 865 706 868 641 934 506 729 595 571 924 691 828 807 611 762 907 676 815 853 497 794 988 826 841 696 938
========================
Actual: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Predicted: 0 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 1 1 0 1 0 1 1 1 0 1 0 0 0 1 0 0 1 1 0 0 1 0 0 0 0 1 0 1 0 1 0 1 1 1 1 0 1 0 0 1 1 0 1 1 1 1 1 1 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 0 0 1

Local Acc: 0.75 , 0.79 , 0.81 , 0.81 , 0.87 , 0.83 , 0.82 , 0.77 , 0.87 , 0.83 , 0.81 , 0.81 , 0.81 , 0.78 , 0.82 , 0.77 , 0.88 , 0.78 , 0.83 , 0.76 , 0.81 , 0.75 , 0.76 , 0.82 , 0.82 , 0.75 , 0.75 , 0.83 , 0.77 , 0.79 , 0.80 , 0.75 , 0.71 , 0.82 , 0.76 , 0.79 , 0.75 , 0.75 , 0.84 , 0.80 , 0.80 , 0.86 , 0.75 , 0.82 , 0.72 , 0.77 , 0.80 , 0.80 , 0.79 , 0.79 , 0.71 , 0.82 , 0.82 , 0.81 , 0.75 , 0.79 , 0.74 , 0.75 , 0.78 , 0.71 , 0.80 , 0.79 , 0.78 , 0.79 , 0.91 , 0.78 , 0.76 , 0.79 , 0.83 , 0.80 , 0.80 , 0.78 , 0.78 , 0.82 , 0.86 , 0.85 , 0.81 , 0.82 , 0.75 , 0.75 , 0.81 , 0.83 , 0.79 , 0.80 , 0.78 , 0.81 , 0.78 , 0.76 , 0.79 , 0.84 , 0.81 , 0.81 , 0.82 , 0.82 , 0.77 , 0.73 , 0.81 , 0.71 , 0.78 , 0.80 , 0.77 , 0.75 , 0.75 , 0.80 , 0.77 , 0.77
K neighbour: 64 , 64 , 64 , 64 , 135 , 64 , 102 , 64 , 80 , 64 , 72 , 64 , 88 , 67 , 74 , 64 , 67 , 64 , 96 , 64 , 64 , 64 , 64 , 64 , 64 , 72 , 64 , 94 , 64 , 64 , 64 , 64 , 64 , 68 , 64 , 64 , 64 , 64 , 108 , 64 , 76 , 105 , 64 , 77 , 64 , 64 , 64 , 64 , 64 , 64 , 64 , 83 , 64 , 87 , 64 , 64 , 64 , 64 , 71 , 66 , 64 , 68 , 64 , 102 , 39 , 64 , 68 , 64 , 91 , 64 , 64 , 93 , 66 , 67 , 125 , 90 , 64 , 77 , 64 , 64 , 64 , 92 , 67 , 64 , 70 , 74 , 64 , 64 , 78 , 99 , 64 , 64 , 88 , 82 , 64 , 73 , 64 , 64 , 64 , 64 , 64 , 64 , 64 , 64 , 64 , 64
c (SVM) : 209 , 197 , 183 , 226 , 223 , 185 , 212 , 216 , 201 , 197 , 168 , 195 , 224 , 212 , 195 , 224 , 206 , 196 , 222 , 236 , 190 , 172 , 227 , 219 , 206 , 225 , 231 , 213 , 233 , 223 , 217 , 205 , 191 , 221 , 188 , 173 , 199 , 202 , 210 , 222 , 221 , 173 , 238 , 232 , 217 , 218 , 216 , 212 , 196 , 217 , 195 , 181 , 190 , 219 , 216 , 179 , 195 , 193 , 223 , 167 , 184 , 220 , 212 , 174 , 174 , 215 , 187 , 215 , 201 , 196 , 221 , 187 , 215 , 214 , 186 , 181 , 218 , 196 , 220 , 231 , 230 , 201 , 237 , 231 , 235 , 221 , 223 , 185 , 187 , 238 , 183 , 181 , 184 , 198 , 175 , 205 , 222 , 228 , 219 , 207 , 222 , 218 , 208 , 201 , 212 , 239
gamma(SVM): 0.0187 , 0.0142 , 0.0091 , 0.0250 , 0.0238 , 0.0100 , 0.0197 , 0.0211 , 0.0157 , 0.0141 , 0.0038 , 0.0135 , 0.0243 , 0.0198 , 0.0136 , 0.0240 , 0.0174 , 0.0138 , 0.0236 , 0.0284 , 0.0118 , 0.0050 , 0.0254 , 0.0224 , 0.0174 , 0.0247 , 0.0266 , 0.0203 , 0.0276 , 0.0239 , 0.0217 , 0.0174 , 0.0122 , 0.0232 , 0.0111 , 0.0056 , 0.0151 , 0.0160 , 0.0191 , 0.0235 , 0.0232 , 0.0056 , 0.0294 , 0.0271 , 0.0216 , 0.0219 , 0.0211 , 0.0197 , 0.0138 , 0.0216 , 0.0135 , 0.0085 , 0.0116 , 0.0222 , 0.0213 , 0.0075 , 0.0137 , 0.0129 , 0.0237 , 0.0035 , 0.0095 , 0.0228 , 0.0197 , 0.0059 , 0.0059 , 0.0208 , 0.0106 , 0.0209 , 0.0158 , 0.0138 , 0.0231 , 0.0105 , 0.0210 , 0.0203 , 0.0104 , 0.0084 , 0.0221 , 0.0138 , 0.0226 , 0.0267 , 0.0264 , 0.0157 , 0.0291 , 0.0268 , 0.0281 , 0.0231 , 0.0239 , 0.0099 , 0.0106 , 0.0293 , 0.0090 , 0.0086 , 0.0094 , 0.0146 , 0.0064 , 0.0173 , 0.0234 , 0.0255 , 0.0224 , 0.0181 , 0.0233 , 0.0221 , 0.0183 , 0.0159 , 0.0199 , 0.0297

Overall Acc: 0.73
Class 1 Acc: 0.76 Class 2 Acc: 0.70

The selected feature list:

Sample 52: 17 selected features
Feature List: 1, 3, 5, 13, 15, 16, 21, 23, 24, 26, 31, 32, 34, 36, 37, 41, 44,
Sample 98: 25 selected features
Feature List: 1, 2, 5, 9, 10, 11, 13, 14, 15, 18, 20, 21, 23, 24, 27, 28, 29, 33, 34, 35, 36, 38, 42, 43, 44,
Sample 266: 19 selected features
Feature List: 1, 2, 3, 4, 7, 8, 9, 13, 21, 29, 30, 32, 35, 37, 38, 39, 40, 42, 44,
Sample 243: 12 selected features
Feature List: 1, 2, 3, 11, 16, 18, 23, 26, 30, 39, 41, 44,
Sample 186: 18 selected features
Feature List: 1, 3, 5, 9, 10, 11, 13, 17, 20, 22, 31, 33, 36, 37, 38, 39, 43, 44,
Sample 16: 18 selected features
Feature List: 1, 5, 6, 7, 8, 9, 11, 13, 16, 19, 21, 23, 24, 25, 26, 28, 31, 42,
Sample 112: 16 selected features
Feature List: 2, 6, 7, 8, 9, 11, 14, 15, 17, 28, 30, 36, 40, 41, 43, 44,
Sample 83: 18 selected features
Feature List: 1, 2, 3, 6, 10, 11, 13, 14, 15, 19, 21, 23, 25, 28, 30, 38, 39, 43,
Sample 432: 22 selected features
Feature List: 3, 5, 6, 8, 11, 12, 14, 18, 24, 27, 28, 29, 30, 34, 35, 36, 37, 39, 40, 41, 43, 44,
Sample 352: 19 selected features
Feature List: 1, 3, 5, 8, 9, 11, 12, 13, 16, 20, 21, 22, 25, 31, 32, 36, 38, 41, 43,
Sample 433: 27 selected features
Feature List: 1, 2, 3, 4, 5, 9, 10, 11, 12, 13, 14, 16, 17, 18, 19, 21, 23, 26, 28, 29, 30, 32, 33, 36, 40, 41, 43,
Sample 381: 14 selected features
Feature List: 1, 3, 5, 11, 18, 19, 31, 35, 36, 38, 39, 40, 41, 42,
Sample 457: 14 selected features
Feature List: 1, 2, 6, 8, 13, 18, 19, 20, 22, 25, 33, 39, 40, 42,
Sample 447: 14 selected features
Feature List: 1, 2, 6, 12, 14, 15, 17, 21, 24, 25, 34, 35, 38, 39,
Sample 168: 19 selected features
Feature List: 1, 4, 5, 6, 8, 10, 12, 16, 19, 24, 31, 33, 34, 37, 38, 39, 40, 42, 43,
Sample 336: 25 selected features
Feature List: 1, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 16, 21, 23, 26, 30, 31, 32, 33, 37, 38, 39, 40, 43, 44,
Sample 166: 20 selected features
Feature List: 2, 4, 7, 10, 11, 13, 14, 15, 17, 24, 26, 27, 30, 31, 32, 33, 34, 40, 41, 44,
Sample 224: 16 selected features
Feature List: 1, 2, 9, 11, 13, 17, 18, 20, 23, 26, 30, 34, 36, 37, 39, 42,
Sample 258: 18 selected features
Feature List: 1, 3, 5, 6, 8, 9, 10, 11, 15, 20, 22, 26, 28, 31, 32, 33, 36, 39,
Sample 219: 15 selected features
Feature List: 1, 3, 5, 7, 15, 16, 17, 18, 20, 24, 25, 39, 40, 41, 44,
Sample 206: 13 selected features
Feature List: 1, 3, 4, 9, 13, 14, 22, 25, 32, 36, 38, 39, 43,
Sample 218: 17 selected features
Feature List: 1, 6, 7, 8, 10, 11, 14, 15, 16, 19, 21, 31, 33, 39, 40, 42, 44,
Sample 163: 13 selected features
Feature List: 1, 6, 10, 14, 15, 16, 17, 19, 26, 28, 32, 39, 40,
Sample 216: 12 selected features
Feature List: 1, 5, 8, 10, 16, 18, 22, 23, 26, 27, 29, 36,
Sample 307: 15 selected features
Feature List: 1, 4, 5, 6, 12, 13, 14, 19, 20, 21, 29, 37, 39, 40, 44,
Sample 422: 14 selected features

Feature List: 1, 2, 4, 10, 13, 18, 20, 23, 26, 32, 37, 39, 42, 43,
Sample 338: 23 selected features
Feature List: 1, 2, 3, 4, 7, 12, 15, 16, 17, 18, 20, 22, 26, 27, 28, 31, 32, 36, 39, 41, 42, 43, 44,
Sample 261: 14 selected features
Feature List: 1, 2, 6, 8, 13, 14, 15, 24, 28, 30, 33, 39, 43, 44,
Sample 401: 19 selected features
Feature List: 1, 3, 5, 6, 9, 17, 18, 19, 22, 23, 24, 27, 30, 33, 35, 36, 38, 41, 42,
Sample 263: 16 selected features
Feature List: 1, 6, 7, 12, 14, 18, 19, 21, 24, 27, 29, 30, 31, 32, 36, 39,
Sample 109: 19 selected features
Feature List: 1, 5, 9, 12, 14, 15, 16, 18, 19, 20, 24, 25, 26, 29, 31, 32, 33, 39, 44,
Sample 386: 14 selected features
Feature List: 1, 3, 4, 5, 12, 15, 16, 22, 25, 29, 31, 35, 42, 44,
Sample 365: 15 selected features
Feature List: 1, 5, 6, 10, 14, 18, 20, 22, 23, 24, 25, 26, 29, 31, 37,
Sample 359: 19 selected features
Feature List: 1, 4, 5, 7, 9, 11, 12, 15, 16, 18, 22, 25, 26, 27, 28, 29, 31, 34, 38,
Sample 214: 15 selected features
Feature List: 1, 2, 3, 5, 6, 9, 13, 15, 19, 21, 28, 29, 33, 40, 44,
Sample 329: 24 selected features
Feature List: 1, 2, 6, 8, 9, 10, 12, 13, 14, 19, 20, 21, 23, 24, 25, 26, 27, 28, 34, 35, 36, 37, 42, 43,
Sample 108: 17 selected features
Feature List: 1, 2, 6, 11, 12, 14, 16, 18, 22, 23, 26, 27, 28, 34, 36, 40, 43,
Sample 129: 11 selected features
Feature List: 1, 4, 10, 12, 24, 31, 34, 38, 40, 42, 44,
Sample 136: 22 selected features
Feature List: 1, 2, 3, 4, 6, 7, 8, 11, 12, 13, 16, 22, 27, 28, 31, 33, 36, 37, 39, 41, 42, 44,
Sample 424: 18 selected features
Feature List: 1, 6, 7, 8, 12, 14, 15, 17, 18, 24, 25, 26, 30, 31, 32, 34, 41, 43,
Sample 170: 21 selected features
Feature List: 1, 6, 8, 9, 11, 12, 13, 14, 15, 17, 19, 22, 25, 28, 31, 32, 33, 34, 35, 36, 38,
Sample 20: 19 selected features
Feature List: 1, 3, 4, 5, 6, 8, 10, 12, 15, 17, 20, 22, 23, 28, 31, 36, 39, 42, 44,
Sample 282: 21 selected features
Feature List: 1, 2, 3, 9, 10, 12, 13, 15, 17, 18, 19, 29, 31, 33, 34, 36, 37, 38, 40, 41, 44,
Sample 230: 16 selected features
Feature List: 1, 4, 6, 9, 11, 12, 13, 14, 18, 19, 25, 26, 27, 38, 39, 44,
Sample 43: 16 selected features
Feature List: 1, 5, 8, 10, 12, 15, 20, 21, 22, 24, 29, 31, 34, 38, 39, 44,
Sample 65: 15 selected features
Feature List: 1, 2, 4, 6, 7, 13, 17, 20, 27, 28, 34, 36, 37, 42, 44,
Sample 79: 18 selected features
Feature List: 1, 4, 5, 6, 7, 8, 10, 14, 15, 16, 17, 22, 35, 36, 38, 39, 42, 44,
Sample 385: 22 selected features
Feature List: 1, 6, 7, 8, 9, 10, 11, 12, 15, 16, 20, 21, 24, 26, 28, 29, 34, 37, 39, 41, 43, 44,
Sample 360: 14 selected features
Feature List: 1, 2, 5, 6, 11, 12, 13, 25, 30, 31, 36, 37, 41, 44,
Sample 708: 15 selected features
Feature List: 1, 4, 5, 8, 10, 13, 14, 15, 19, 24, 28, 29, 33, 37, 38,

Sample 533: 8 selected features
Feature List: 1, 4, 23, 25, 26, 34, 35, 40,
Sample 752: 18 selected features
Feature List: 1, 5, 6, 7, 8, 10, 11, 12, 13, 14, 19, 20, 22, 24, 25, 26, 38, 42,
Sample 915: 15 selected features
Feature List: 1, 3, 6, 7, 9, 18, 21, 27, 30, 32, 33, 36, 37, 39, 41,
Sample 982: 20 selected features
Feature List: 1, 3, 4, 13, 14, 15, 16, 19, 21, 23, 26, 27, 28, 29, 32, 33, 34, 38, 40, 41,
Sample 563: 15 selected features
Feature List: 1, 5, 7, 8, 13, 18, 22, 23, 25, 27, 33, 36, 37, 42, 44,
Sample 855: 16 selected features
Feature List: 1, 2, 14, 15, 16, 19, 22, 25, 26, 27, 28, 29, 35, 37, 38, 43,
Sample 918: 21 selected features
Feature List: 1, 3, 6, 7, 9, 10, 11, 13, 15, 16, 22, 23, 24, 26, 27, 30, 31, 32, 36, 38, 41,
Sample 738: 18 selected features
Feature List: 1, 5, 7, 11, 12, 14, 17, 19, 22, 24, 25, 27, 28, 29, 33, 34, 40, 42,
Sample 832: 17 selected features
Feature List: 1, 2, 3, 4, 8, 10, 15, 16, 18, 21, 22, 25, 26, 28, 31, 32, 36,
Sample 961: 13 selected features
Feature List: 3, 5, 7, 9, 11, 12, 16, 19, 22, 25, 26, 34, 36,
Sample 956: 23 selected features
Feature List: 1, 7, 9, 10, 13, 14, 15, 16, 18, 20, 22, 24, 25, 27, 29, 30, 32, 33, 34, 36, 38, 40, 42,
Sample 989: 14 selected features
Feature List: 1, 5, 6, 7, 8, 11, 12, 13, 15, 17, 28, 32, 34, 35,
Sample 838: 19 selected features
Feature List: 1, 2, 3, 4, 5, 6, 12, 13, 15, 19, 22, 24, 25, 26, 30, 31, 35, 37, 44,
Sample 958: 8 selected features
Feature List: 5, 17, 19, 21, 25, 30, 33, 35,
Sample 886: 27 selected features
Feature List: 1, 3, 5, 6, 7, 9, 10, 12, 13, 14, 15, 18, 19, 20, 21, 24, 25, 27, 30, 31, 32, 33, 34, 35, 38, 39, 43,
Sample 803: 20 selected features
Feature List: 2, 4, 5, 6, 13, 17, 22, 26, 27, 28, 30, 31, 36, 37, 38, 39, 40, 41, 42, 44,
Sample 873: 15 selected features
Feature List: 1, 6, 7, 11, 15, 17, 18, 19, 21, 24, 25, 28, 29, 41, 42,
Sample 919: 17 selected features
Feature List: 1, 2, 3, 9, 10, 11, 13, 16, 18, 21, 25, 29, 30, 31, 32, 40, 42,
Sample 682: 18 selected features
Feature List: 1, 2, 3, 4, 11, 12, 14, 15, 17, 19, 21, 25, 28, 29, 30, 33, 36, 42,
Sample 525: 15 selected features
Feature List: 1, 5, 10, 11, 15, 16, 18, 19, 20, 30, 31, 32, 34, 38, 41,
Sample 1010: 22 selected features
Feature List: 1, 2, 4, 5, 11, 12, 14, 15, 17, 23, 24, 28, 29, 30, 31, 32, 34, 35, 36, 37, 41, 44,
Sample 950: 14 selected features
Feature List: 3, 9, 17, 19, 21, 28, 29, 30, 35, 39, 41, 42, 43, 44,
Sample 667: 21 selected features
Feature List: 1, 2, 3, 6, 7, 8, 10, 11, 18, 19, 23, 26, 29, 30, 34, 35, 38, 40, 41, 42, 44,
Sample 503: 11 selected features
Feature List: 1, 2, 6, 8, 23, 25, 27, 30, 33, 34, 36,
Sample 583: 19 selected features
Feature List: 1, 2, 4, 6, 7, 11, 15, 16, 19, 20, 23, 26, 27, 30, 34, 36, 39, 40, 41,

Sample 1000: 16 selected features
Feature List: 1, 5, 8, 9, 10, 12, 13, 22, 23, 26, 28, 29, 31, 32, 39, 42,
Sample 798: 26 selected features
Feature List: 1, 2, 3, 4, 10, 11, 14, 16, 17, 20, 22, 23, 24, 26, 28, 29, 30, 32, 34, 37, 38, 39, 40, 41, 42, 44,
Sample 744: 22 selected features
Feature List: 1, 2, 3, 11, 12, 13, 18, 23, 24, 26, 28, 30, 31, 32, 34, 35, 36, 37, 38, 39, 40, 42,
Sample 912: 16 selected features
Feature List: 1, 2, 4, 12, 15, 21, 22, 23, 25, 26, 29, 30, 34, 36, 40, 44,
Sample 979: 22 selected features
Feature List: 1, 3, 9, 10, 12, 13, 14, 15, 17, 19, 21, 24, 25, 26, 27, 32, 33, 34, 35, 36, 38, 41,
Sample 865: 11 selected features
Feature List: 1, 2, 3, 5, 15, 16, 18, 19, 29, 33, 44,
Sample 706: 14 selected features
Feature List: 1, 4, 8, 9, 15, 23, 26, 28, 31, 32, 34, 35, 38, 40,
Sample 868: 21 selected features
Feature List: 1, 3, 4, 5, 6, 10, 11, 14, 15, 17, 18, 23, 24, 26, 32, 35, 36, 39, 41, 43, 44,
Sample 641: 15 selected features
Feature List: 1, 6, 7, 8, 17, 19, 23, 26, 28, 34, 36, 37, 39, 40, 42,
Sample 934: 18 selected features
Feature List: 1, 8, 10, 11, 16, 22, 23, 24, 28, 32, 33, 34, 35, 37, 39, 40, 42, 44,
Sample 506: 17 selected features
Feature List: 1, 6, 11, 12, 13, 14, 15, 16, 17, 19, 20, 23, 30, 32, 36, 39, 41,
Sample 729: 21 selected features
Feature List: 1, 2, 5, 7, 9, 10, 11, 13, 14, 15, 16, 18, 21, 23, 24, 26, 30, 31, 32, 35, 43,
Sample 595: 18 selected features
Feature List: 1, 8, 10, 17, 19, 21, 23, 24, 27, 28, 29, 32, 33, 34, 36, 37, 39, 40,
Sample 571: 20 selected features
Feature List: 1, 2, 5, 11, 15, 17, 18, 19, 20, 22, 24, 25, 29, 30, 31, 33, 34, 38, 39, 41,
Sample 924: 16 selected features
Feature List: 1, 2, 3, 5, 13, 14, 26, 27, 29, 30, 38, 40, 41, 42, 43, 44,
Sample 691: 14 selected features
Feature List: 1, 4, 9, 10, 11, 19, 26, 27, 28, 29, 34, 35, 40, 43,
Sample 828: 22 selected features
Feature List: 1, 3, 6, 7, 9, 10, 11, 14, 16, 17, 18, 19, 21, 27, 28, 29, 33, 35, 38, 41, 42, 44,
Sample 807: 26 selected features
Feature List: 1, 2, 4, 5, 6, 10, 12, 13, 14, 15, 17, 18, 19, 22, 23, 24, 25, 26, 27, 28, 30, 38, 41, 42, 43, 44,
Sample 611: 18 selected features
Feature List: 1, 3, 4, 5, 8, 11, 12, 17, 19, 20, 23, 24, 25, 30, 31, 38, 39, 40,
Sample 762: 21 selected features
Feature List: 1, 2, 3, 4, 5, 6, 7, 13, 16, 18, 19, 20, 22, 23, 24, 25, 26, 29, 31, 40, 44,
Sample 907: 18 selected features
Feature List: 1, 2, 4, 14, 15, 19, 20, 21, 22, 27, 29, 30, 32, 36, 37, 39, 40, 42,
Sample 676: 14 selected features
Feature List: 1, 2, 4, 5, 9, 10, 11, 14, 20, 22, 25, 26, 28, 39,
Sample 815: 16 selected features
Feature List: 1, 2, 5, 8, 13, 14, 15, 18, 19, 26, 31, 32, 37, 40, 42, 43,
Sample 853: 20 selected features
Feature List: 1, 2, 4, 6, 7, 9, 11, 13, 16, 17, 23, 24, 25, 26, 34, 35, 37, 40, 41, 44,
Sample 497: 22 selected features

Feature List: 1, 2, 3, 5, 6, 8, 10, 11, 12, 13, 15, 16, 21, 25, 26, 29, 34, 35, 37, 40, 42, 44,
Sample 794: 16 selected features
Feature List: 1, 8, 12, 16, 17, 18, 19, 21, 22, 25, 28, 29, 32, 38, 43, 44,
Sample 988: 16 selected features
Feature List: 3, 4, 7, 11, 14, 17, 19, 23, 26, 29, 31, 33, 35, 36, 38, 41,
Sample 826: 17 selected features
Feature List: 1, 8, 9, 11, 12, 18, 19, 22, 24, 25, 26, 27, 30, 31, 32, 35, 42,
Sample 841: 16 selected features
Feature List: 1, 6, 8, 9, 13, 14, 25, 27, 31, 32, 33, 37, 39, 40, 42, 44,
Sample 696: 15 selected features
Feature List: 1, 2, 6, 9, 10, 11, 15, 19, 22, 27, 28, 30, 33, 38, 43,
Sample 938: 17 selected features
Feature List: 1, 3, 12, 14, 17, 19, 21, 23, 29, 32, 33, 34, 35, 37, 39, 40, 42

The frequency of selected features:

Feature ID:	1	11	15	19	26	6	5	13	39	44	2	14	36	12	25	28
Selected times:	98	50	50	50	50	49	48	47	47	46	45	44	44	43	43	43

Feature ID:	31	3	10	34	40	42	29	30	32	18	23	24	38	22	4	8
Selected times:	43	42	42	42	42	42	41	41	41	40	40	40	40	39	38	38

Feature ID:	9	16	17	33	41	21	27	37	7	35	20	43
Selected times:	38	38	38	36	36	34	34	34	33	32	29	28

APPENDIX M

Validation results of SNPs data sample 392 for CD risk evaluation using

The personalized modeling based method over sample 392 of SNPs data for CD risk prediction

Sample Id: 392

==========

Run 1

actual -1

predicted -1

local Acc 0.82

K neighbor 83

c (SVM) 207

gamma(SVM) 0.0178

Sample 392; 20 selected features

Feature List: 1 2 4 5 8 9 10 11 12 16 20 22 23 24 27 28 33 37 40 43

==========

Run 2

actual -1

predicted -1

local Acc 0.83

K neighbor 83

c (SVM) 235

gamma(SVM) 0.0283

Sample 392; 17 selected features

Feature List: 1 3 4 8 9 10 11 12 16 20 22 24 25 27 28 33 37

==========

Run 3

actual -1

predicted -1

local Acc 0.83

K neighbor 83

c (SVM) 216

gamma(SVM) 0.0214

Sample 392; 16 selected features

Feature List: 1 3 6 7 8 9 11 14 15 20 23 24 31 33 39 40

==========

Run 4

actual -1

predicted -1

local Acc 0.82

K neighbor 83

c (SVM) 230

gamma(SVM) 0.0262

```
Sample  392;    16 selected features
  Feature List:  1   3  10  11  13  15  17  18  19  20  24  25  26  27  32  36

=======================
```

Run 5

```
actual       -1
predicted    -1
local Acc  0.81
K neighbor   83
c (SVM)     193
gamma(SVM) 0.0127
Sample  392;    11 selected features
  Feature List:  1   3   6  12  20  22  23  24  25  27  30

=======================
```

Run 6

```
actual       -1
predicted    -1
local Acc  0.83
K neighbor   83
c (SVM)     180
gamma(SVM) 0.0082
Sample  392;    10 selected features
  Feature List:  1   3   6   9  12  15  20  23  33  39

=======================
```

Run 7

```
actual       -1
predicted    -1
local Acc  0.82
K neighbor   83
c (SVM)     230
gamma(SVM) 0.0263
Sample  392;    14 selected features
  Feature List:  1   4   5   6   8  11  12  20  24  28  33  36  38  43

=======================
```

Run 8

```
actual       -1
predicted    -1
local Acc  0.82
K neighbor   83
c (SVM)     240
gamma(SVM) 0.0299
Sample  392;    13 selected features
```

```
  Feature List:  1   2   4  12  18  20  23  24  25  27  31  39  42

 ========================
```

Run 9

```
 actual       -1
 predicted    -1
 local Acc  0.83
 K neighbor   83
 c (SVM)     233
 gamma(SVM) 0.0273
 Sample  392;    14 selected features
  Feature List:  1   5   9  11  20  22  23  24  28  30  33  36  38  42

 ========================
```

Run 10

```
 actual       -1
 predicted    -1
 local Acc  0.82
 K neighbor   83
 c (SVM)     237
 gamma(SVM) 0.0288
 Sample  392;    13 selected features
  Feature List:  1   3   7  10  12  15  18  20  24  25  33  37  40

 ========================
```

Run 11

```
 actual       -1
 predicted    -1
 local Acc  0.83
 K neighbor   83
 c (SVM)     225
 gamma(SVM) 0.0247
 Sample  392;    18 selected features
  Feature List:  1   2   3   4   7   9  11  12  20  23  24  26  27  28  32  33  37  40

 ========================
```

Run 12

```
 actual       -1
 predicted    -1
 local Acc  0.83
 K neighbor   83
 c (SVM)     203
 gamma(SVM) 0.0166
 Sample  392;    16 selected features
  Feature List:  1   3   4   6   9  10  11  18  20  23  24  30  32  33  37  42
```

```
=======================
```

Run 13

```
actual       -1
predicted    -1
local Acc  0.83
K neighbor   83
c (SVM)     216
gamma(SVM) 0.0214
Sample  392;    15 selected features
  Feature List:  1   3   4   9  10  15  18  19  20  28  30  31  32  34  42

=======================
```

Run 14

```
actual       -1
predicted    -1
local Acc  0.82
K neighbor   83
c (SVM)     234
gamma(SVM) 0.0279
Sample  392;    13 selected features
  Feature List:  1   3   6  12  15  18  20  24  25  36  38  43  44

=======================
```

Run 15

```
actual       -1
predicted    -1
local Acc  0.83
K neighbor   83
c (SVM)     211
gamma(SVM) 0.0193
Sample  392;    15 selected features
  Feature List:  1   4   9  10  11  12  20  22  23  24  28  33  37  38  40

=======================
```

Run 16

```
actual       -1
predicted    -1
local Acc  0.84
K neighbor   83
c (SVM)     216
gamma(SVM) 0.0212
Sample  392;    17 selected features
  Feature List:  1   3   6   7   9  10  11  12  18  20  23  24  26  31  33  37  38
```

```
=======================
```

Run 17

```
actual       -1
predicted    -1
local Acc  0.82
K neighbor   83
c (SVM)     221
gamma(SVM) 0.0230
Sample  392;    17 selected features
  Feature List:  1   2   4   9  10  12  20  22  23  24  25  27  28  33  35  40  42

=======================
```

Run 18

```
actual       -1
predicted    -1
local Acc  0.82
K neighbor   83
c (SVM)     231
gamma(SVM) 0.0268
Sample  392;    13 selected features
  Feature List:  1   3   8   9  10  12  18  20  23  24  26  37  39

=======================
```

Run 19

```
actual       -1
predicted    -1
local Acc  0.82
K neighbor   83
c (SVM)     214
gamma(SVM) 0.0204
Sample  392;    13 selected features
  Feature List:  1   4   8   9  16  22  24  25  27  28  32  40  41

=======================
```

Run 20

```
actual       -1
predicted    -1
local Acc  0.82
K neighbor   83
c (SVM)     208
gamma(SVM) 0.0184
Sample  392;    13 selected features
  Feature List:  1   5  12  20  22  23  24  25  26  27  28  31  36
```