Proceedings of International Joint Conference on Neural Networks, Montreal, Canada, July 31 - August 4, 2005

Transductive Modeling with GA Parameter Optimization

Nisha Mohan and Nikola Kasabov

Knowledge Engineering & Discovery Research Institute, Auckland University of Technology
Private Bag 92006, Auckland 1020, New Zealand
Email: nmohan@aut.ac.nz, nkasabov@aut.ac.nz

Introduction - While inductive modeling is used to develop a model (function) from data of the whole problem space and then to recall it on new data, transductive modeling is concerned with the creation of single model for every new input vector based on some closest vectors from the existing problem space. The model approximates the output value only for this input vector. However, deciding on the appropriate distance measure, on the number of nearest neighbors and on a minimum set of important features/variables is a challenge and is usually based on prior knowledge or exhaustive trial and test experiments.

This paper proposes a Genetic Algorithm (GA) approach for optimizing these three factors. The method is tested on several datasets from UCI repository for classification tasks and results show that it outperforms conventional approaches. The drawback of this approach is the computational time complexity due to the presence of GA, which can be overcome using parallel computer systems due to the intrinsic parallel nature of the algorithm.

KEYWORDS - Transductive inference, GA, Multi Linear Regression, Inductive Models

1. INTRODUCTION

Transductive inference, introduced by Vapnik [1] is defined as a method used to estimate the value of a potential model (function) only for a single point of space (that is, the new data vector) by utilizing additional information related to that vector. This inference technique is in contrast to inductive inference approach where a general model (function) is created for all data representing the entire problem space and the model is applied then on new data (deduction). While the inductive approach is useful when a global model of the problem is needed in an approximate form, the transductive approach is more appropriate for applications where the focus is not on a model, rather on individual cases, for example, clinical and medical applications where the focus needs to be centered on individual patient's conditions rather than on the global, approximate model.

The transductive approach is related to the common sense principle [2] which states that to solve a given problem one should avoid solving a more general problem as an intermediate step. The reasoning behind this principle is that, in order to solve a more general problem, resources are wasted or compromised which is unnecessary for solving the individual problem at hand (that is, function estimation only for given points). This common sense principle reduces the more general problem of inferring a functional dependency on the whole input space (inductive approach) to the problem of estimating the values of a function only at given points (transductive approach).

In the past 5 years, transductive reasoning has been implemented for a variety of classification tasks such as text classification [3], heart disease diagnostics [4], synthetic data classification using graph based approach[5], digit and speech recognition [6], promoter recognition in bioinformatics [7], image recognition [8] and image classification [9], micro array gene expression classification [10] and biometric tasks such as face surveillance [11]. This reasoning method is also used in prediction tasks such as predicting if a given drug binds to a target site [12] and evaluating the prediction reliability in regression [2] and providing additional measures to determine reliability of predictions made in medical diagnosis [13].

In transductive reasoning, for every new input vector x_i that needs to be processed for a prognostic/classification task, the Ni nearest neighbors, which form a data subset Di, are derived from an existing dataset D and a new model Mi is dynamically created from these samples to approximate the function in the locality of point x_i only. The system is then used to calculate the output value y_i for this input vector x_i. This approach has been implemented with radial basis function as the base model [14] in medical decision support systems and time series prediction problem, where individual models are created for each input data vector (that is, specific time period or specific patient). The approach gives a good accuracy for individual models and has promising applications especially in medical decision support systems. Transductive approach has also been applied using support vector machines as the base model in area of bioinformatics [7, 10] and the results indicate that transductive inference performs better than inductive inference models mainly because it exploits the structural information of the new, unlabeled data. However, there are a few open questions that need to be addressed while implementing transductive modeling.

Question 1: How many nearest neighbors should be used to derive a model for every new input vector?

A standard approach, adopted in other research papers [15-17] is to consider a range starting with 1, 2, 5, 10, 20 and so on and finally select the best value based on the classifier's performance. Alternatively, in the presence of

0-7803-9048-2/05/$20.00 ©2005 IEEE

unbalanced data distribution among classes in the problem space, Hand and Vinciotti [18] recommend the value of nearest neighbors to range from 1 to a maximum of number of samples in the smaller class. In cases of datasets with large number of instances in hand, Jonnson et al [19] recommend 10 neighbors based on results from a series of exhaustive experiments. In contrast to this recommendation, Duda and Hart [20] proposed using square root of the number of all samples based on the concept of probability density estimation. Alternatively, Enas and Choi [21] suggested that the number of nearest neighbors depends on two important factors: a) Distribution of sample proportions in the problem space; b) Relationship between the samples in the problem space measured using covariance matrices. Based on exhaustive empirical studies, they suggested using the value of k as $N^{3/8}$ or $N^{2/8}$ based on the differences between covariance matrices for class proportions and difference between class proportions. The problem of identifying the optimal number of neighbors that help improve the classification accuracy in transductive modeling remains an open question that needs to be addressed.

Question 2: What type of distance measure to use in order to define the neighbourhood for every new input vector?

There exist different types of distance measures that can be considered to measure the distance of two vectors in a different part of the problem/feature space such as Euclidean distance, Mahalanobis distance, Hamming distance, Cosine distance, Correlation distance, Manhattan distance among others. It has been proved mathematically that using an appropriate distance metric can help reduce classification error while selecting neighbors without increasing number of sample vectors [22]. Hence it is important to recognise which distance measure will best suit the data in hand. In spite of this fact, it has been observed that Euclidean distance forms the most common form of distance metric mainly due to ease of calculations [15, 18, 23] and several others. In contrast to this, in a case study of gene expression data, Brown et al [24] recommend Cosine measure over Euclidean distance, as Cosine considers angle of data and is not affected by the length of data or outliers which could be a problem with Euclidean distance metric. On the other hand, Troyanskaya et al [16] suggested that effect of outliers can be reduced using log-transform or any other normalization technique and thus recommend Euclidean measure after performing a comparison of Euclidean, variance minimization and correlation measures for gene expression data. In view of these contradicting suggestions for selection of distance measure to identify neighboring data vectors, there is a need to follow standardization while considering the appropriate distance measure.

Hirano et al [25] arrived at one such standardization for selecting the type of distance measure based on properties of the dataset. They suggest that in case the dataset consists of numerical data, Euclidean distance measure should be used when the attributes are independent and commensurate with each other. However, in case of high interdependence of the input variables, Mahalanobis distance should be considered as this distance measure takes inter-dependence between the data into consideration. On the other hand, if the data consists of categorical information, Hamming distance should be used as it can appropriately measure the difference between categorical data. Also, in case the dataset consists of a combination of numerical and categorical values, for example - a medical dataset that includes numerical variables such as gene expression values and categorical variables such as clinical attributes, then a weighted sum of Mahalanobis or Euclidean for numerical data and Hamming distance for nominal data is recommended.

Keeping these suggestions in perspective, it is important to provide a wide range of options to select the distance measure based on type of dataset in a particular part of the problem space for a particular set of features.

Question 3: What features are important for every new input vector?

Feature selection is a search problem [26] that consists of feature subset generation, evaluation and selection. Feature selection is useful for 3 main purposes: reduce the number of features and focus on those features that have a strong influence on the classification performance; improve classification accuracy; simplify the knowledge representation and the explanation derived from the individual model.

The three questions above are addressed here by introducing a GA approach. Generally speaking, GAs provide an useful strategy for solving optimization tasks when other optimization methods, such as forward search, gradient descent or direct discovery, are not feasible with respect to their computational complexity. Moreover, since we need to optimize several parameters at the same time and find a combination that gives optimal results, the intrinsic parallelism of GA seems most appropriate to perform the implementation on a largely parallel architecture. In the paper, a selection is made from the types of distance measures as shown in Fig 1. The number of neighbors to be optimized lies in the minimum range of 1 (in case of kNN classification algorithm) or number of features selected (in case of linear classifier function) and a maximum of number of samples available in the problem space.

We also use GA to identify the reduced feature set. There has been a controversy over the application of GA for feature selection [27] as some authors find this approach very useful [28-31] while others are skeptical [32, 33]

and not impressed with the results presented by GA in comparison to other feature selection algorithms [34]. The main reason for using GA for feature selection in a transductive setting are: 1) Features are selected in their combination with the number of nearest neighbors and the optimal distance measure; 2) This technique allows for an unbiased feature selection where the test data is not considered while selecting the optimal features thus avoiding bias in the final feature-subset obtained; 3) The entire range of features is represented in a chromosome making the encoding task straightforward.

Euclidean Distance	$D(x,y)=\sqrt{\sum_{i=1}^{m}(x_i-y_i)^2}$
City Block Distance	$D(x,y)=\sum_{i=1}^{m}\lvert x_i-y_i\rvert$
Correlation Distance	$D(x,y)=\frac{\sum_{i=1}^{m}(x_i-\overline{x_i})(y_i-\overline{y_i})}{\sqrt{\sum_{i=1}^{m}(x_i-\overline{x_i})^2\sum_{i=1}^{m}(y_i-\overline{y_i})^2}}$
Cosine Distance	$D(x,y)=\frac{1-\sum_{i=1}^{m}x_i*y_i}{\sqrt{\sum_{i=1}^{m}(x_i)^2\sum_{i=1}^{m}(y_i)^2}}$

Fig 1: Equations of selected distance measures (where x and y are vectors of m attribute values)

For this research work, multi linear regression[35] is used as the base model for applying the transductive approach. The model is represented by a linear equation which links the features/variables in the problem space to the output of the classification (or prediction) task and is represented in Equation (1):

$$r = w_0 + w_1X_1 + \ldots + w_nX_n \quad (1)$$

Where r represents the output and w_0 represents the bias and w_i represent the weights for the features/variables of the problem space which are calculated using least square method. The descriptors X_n are used to represent the features/variables and n represents the number of these features/variables. The reason for selecting multi linear regression model is the simplicity of this model that will make the comparative analysis of the Transductive approach using GA with the inductive approach easier to understand and interpret.

2. PROPOSED ALGORITHM

The main objective of this algorithm is to develop an individualized model for every data vector in a semi-supervised manner by exploiting the data vector's structural information, identifying its nearest neighbors in the problem space and finally testing the model using the neighboring data vectors to check the effectiveness of the model created. GA is used to locate an effective set of features that represent most of the data's significant structural information along with the optimal number of neighbors and the optimal distance measure to identify the neighbors. The complete algorithm is described in two parts: the transductive inference procedure; and the GA optimization procedure.

Transductive_MLR_withGA-Optimization (Sample *S*, Dataset *D*);

Calculate *D* = Linear_normalization (*D*)
For *Sample*=1 to size (*D*)
  Set *GA_parameters* to *Generations, Populations, Crossover rate & Mutation rate*
  Initialize *CurrentGen*=1
  Initialize random start values for
  1. Number of neighbors ($K_{CurrentGen}$) between number of features selected and maximum size(*EntireData*)
  2. Distance function ($D_{CurrentGen}$) between Euclidean, Manhattan/City Block, Correlation and Cosine
  3. Number of features ($F_{CurrentGen}$) based on binary selection.
  While *CurrentGen* <*Generations*
    Initialize *CurrentPop*=1
    While *CurrentPop*<*Populations*
      Select $K_{CurrentGen}$ neighbours $Neighbors_{sample}$ of the *sample S* with distance measure as $D_{CurrentGen}$ and feature list as $F_{CurrentGen}$.
      Calculate the fitness function for *CurrentPop* as follows. We assume that that if the parameters [$K_{CurrentGen}$ $D_{CurrentGen}$ $F_{CurrentGen}$] produce a MLR model with high classification accuracy for $Neighbors_{sample}$, then these parameters will work well with the sample S as well. We calculate the classification accuracy for $Neighbors_{sample}$ with MLR model using data extracted with parameters [$K_{CurrentGen}$ $D_{CurrentGen}$ $F_{CurrentGen}$] as follows:
      For *m*=1 to size($Neighbors_{sample}$)
        Find $K_{CurrentGen}$ neighbors of $Neighbors_{sample}(m)$ using $D_{CurrentGen}$ distance measure and $F_{CurrentGen}$ features.
        Run Multi Linear Regression model with $K_{CurrentGen}$ neighbors of $Neighbors_{sample}(m)$ as the *traindata* and $Neighbors_{sample}(m)$ as *testdata*. Calculate classification accuracy as *Accuracy (m)*.
      EndFor
      Calculate average (*Accuracy*) as fitness function for the GA.
    EndWhile
    Sort the fitness function values across all *populations* and select the two best populations with maximum fitness function as parents for GA.
    Mutate and Crossover to create offspring with parameters for [$K_{CurrentGen}$ $D_{CurrentGen}$ $F_{CurrentGen}$] using *Crossover rate & Mutation rate.*
  EndWhile.
  Select the final population with maximum fitness function as the best model parameters [*K D F*] for the *Sample S.*
  Test these model parameters [*K D F*] for *D(Sample)* as *testdata* and calculate classification accuracy Acc_{sample}
EndFor.

Fig 2: Algorithm for Transductive modeling with GA parameter optimization

Transductive inference procedure

1. The dataset is normalized linearly (values between 0 and 1) to ensure a standardization of all values. This normalization procedure is based on the assumption that all variables/features have the same importance for the

output of the system in the whole problem space.

2. For every test sample T_i, perform the following steps: Select closest neighboring samples, create a model and evaluate its accuracy. For every test sample T_i we select through the proposed GA procedure a set of features to be considered, the number of nearest neighbors, and the distance measure to locate the neighbors. A typical GA chromosome includes the following fields ("genes"): Distance Measure; Number of neighbors; Feature set. The accuracy of the selected set of parameters for T_i model is calculated by creating a model with these parameters for each of the neighbors of test sample T_i and calculating the accuracy of each of these models. The cross validation is run in a leave one out mode for all neighbors of T_i. If, for the identified set of parameters, the neighbors of T_i give a high classification accuracy rate, then we assume that the same set of parameters will also work for the sample T_i. This criterion is used as a fitness evaluation criterion for the GA optimization procedure.

3. Perform the set of operations in step 2 in a leave one out manner for all the samples in the dataset and calculate the overall classification accuracy for this transductive approach.

GA optimization procedure

GA [36-38] have found a considerable range of applications for global optimization of objective functions. In our optimization procedure, for every new input vector T_i, the initial populations of models begin with randomly selected, from a given constrained range of values, set of parameters for distance measure, K value and a subset of features. Here K values range from a minimum of number of features to a maximum of all samples in problem space. For the bits representing the features in the chromosome, a value of '1' represents the selection of the feature and a value of '0' denotes that the feature is not selected.

The fitness function represents the average classification accuracy obtained using a particular chromosome (model). The fitness function is calculated as the average of the classification accuracy for all selected neighbors of the test sample T_i. The hypothesis held here is that if the average classification accuracy is good for all neighbors of the test sample with a particular chromosome, then that chromosome will work well for the test sample.

Rank based selection with elitism strategy [37] is applied for selecting the best performing chromosome for further reproduction. Elitist strategy ensures that at least one copy of the best individual in the population is always passed onto the next generation. The main advantage of this strategy is that convergence is guaranteed, i.e., if global maximum is discovered, the GA converges to that maximum. However, by the same token, there is a risk of being trapped in a local maximum. The reproduction involves: a) Crossover - Once the selection procedure is complete, a uniform crossover is applied with a crossover rate kept as high as possible and the selected parents from the previous generation are crossovered with a high cross over probability; b) Mutation – Uniform mutation is carried out that makes small alterations to the values of all selected genes to create the next generation. A binary mutation procedure is applied which simply requires the mutated bit to become its complement. Probability of mutation is normally kept as 1/(number of bits in the chromosome) which acts as a fixed pre-determined probability for mutation for every gene (bit) in every chromosome. This mutation rate is kept very low to keep the search from diversifying rapidly. Fig. 2 shows the psuedocode for the transductive multi linear regression algorithm with GA parameter optimization.

3. EXPERIMENTS

We conducted experiments on various UC Irvine datasets [39] with their characteristics represented in table 1. The tests were carried out on the entire data using leave one out validation technique. The datasets were selected as the ones without any missing values except for breast cancer dataset that had 4 missing values. At the pre-processing stage, the four samples with missing values were deleted and the size of breast cancer dataset reduced from 198 to 194. As the next step of pre-processing, all the datasets were normalized using linear normalization resulting in values in the range of 0 and 1 to provide standardization.

Table 1: Characteristics of datasets

Dataset	# of classes	# of features	# of data points
Thyroid	3	5	215
Sonar	2	60	208
Glass	7	10	214
Breast Cancer	2	30	194

The Transductive MLR with parameter optimization is compared against inductive MLR and Transductive MLR without optimization. In the latter case, the parameters for nearest neighbors and distance measure are kept fixed after selecting which distance measure would be best to use for a particular case study problem (see the table in the appendix). Table 2 presents cross validation results for comparison between inductive and transductive modeling approach without and with GA parameter optimization only for MLR types of models.

Table 2: Leave one out cross validation accuracy obtained for different MLR modeling approaches: inductive MLR models; transductive MLR models without GA parameter optimization but using the best known so far values for the three parameters under optimization; transductive MLR modeling with GA parameter optimization (fig.2).

Data	Inductive. MLR	Best of Transd. MLR with fixed (best) parameters (see appendix)	Transd MLR with GA optim.
Thyroid	86.51	**94.88**	**94.88**
Breast Cancer	72.16	67.01	**73.71**
Sonar	75.48	78.81	**81.25**
Glass	60.75	68.69	**71.96**

4. DISCUSSIONS AND CONCLUSION

The results show that the transductive modeling approach with GA optimization significantly outperforms the inductive modeling and the parameter optimization procedure improves the accuracy of the individual models at average. An extension of the current work will deal with: Using connectionist models that enable rule extraction for each individualized models; GA implementation with real values instead of binary values for feature selection that will indicate the normalization range and the importance of each feature for each individual model [38]; Using a hybrid approach for the optimization task where the GA will provide the starting point for local search algorithms to find the optimal parameter combination.

5. ACKNOWLEDGEMENTS

This research work is supported in part by the CoRE for Bioprotection research, Lincoln University and the Knowledge Engineering and Discovery research Institute KEDRI. The software is available for academic purposes. We would like to thank P.Hwang, D.Greer, Dr Q.Song and all reviewers for their comments and support.

6. APPENDIX

Table 3: Classification accuracy of transductive MLR models using different fixed values for K and type of distance. The best model is reported in table 2.

Num. of Nearest neighbors	Distance measure	Thyroid	Sonar	Breast Cancer	Glass
$\sqrt{N}$	Euclidean	93.02	77.88	54.64	68.69
	Correlation	-	77.88	57.22	-
	Cityblock	92.56	79.81	54.12	64.95
	Cosine	-	77.40	60.82	-
Smaller class of samples in N	Euclidean	94.88	75.48	64.95	32.71
	Correlation	-	75.96	67.01	-
	Cityblock	94.88	74.04	61.86	32.71
	Cosine	-	75	65.46	-

7. REFERENCES

[1] V. Vapnik, *Statistical Learning Theory*, 1995.

[2] Z. Bosnic, I. Kononenko, M. Robnik-Sikonja, and M. Kukar, "Evaluation of prediction reliability in regression using the transduction principle," *EUROCON 2003. Computer as a Tool. The IEEE Region 8*, vol. 2, pp. 99 - 103, 2003.

[3] T. Joachims, "Transductive Inference for Text Classification using Support Vector Machines," presented at Proceedings of the Sixteenth International Conference on Machine Learning, 1999.

[4] D. Wu, N. Cristianini, J. Shawe-Taylor, and K. P. Bennett, "Large Margin Trees for Induction and Transduction," presented at Proceedings for 16th International conference of machine learning, Bled, Slovenia, 1999.

[5] C. H. Li and P. C. Yuen, "Transductive learning: Learning Iris data with two labeled data," presented at ICANN 2001, Berlin, Heidelberg, 2001.

[6] T. Joachims, "Transductive Learning via Spectral Graph Partitioning," presented at Proceedings of the Twentieth International Conference on Machine Learning, ICML-2003, Washington DC, 2003.

[7] N. Kasabov and S. Pang, "Transductive Support Vector Machines and Applications in Bioinformatics for Promoter Recognition," *Neural Information Processing - Letters and Reviews*, vol. 3, pp. 31-38, 2004.

[8] J. Li and C.-S. Chua, "Transductive inference for color-based particle filter tracking," presented at Proceedings of International Conference on Image Processing, 2003, Nanyang Technol. Univ., Singapore, 2003.

[9] K. Proedrou, I. Nouretdinov, V. Vovk, and A. Gammerman, "Transductive confidence machine for pattern recognition," presented at Proceedings of the 13th European Conference on Machine Learning, 2002.

[10] S. Pang and N. Kasabov, "Inductive vs Transductive Inference, Global vs Local Models: SVM, TSVM, and SVMT for Gene Expression Classification Problems," presented at International Joint Conference on Neural Networks, IJCNN 2004, Budapest, 2004.

[11] F. Li and H. Wechsler, "Watch List Face Surveillance Using Transductive Inference," *Lecture Notes in Computer Science*, vol. 3072, pp. 23-29, 2004.

[12] J. Weston, F. Pérez-Cruz, O. Bousquet, O. Chapelle, A. Elisseeff, and B. Schölkopf, "Feature selection and transduction for prediction of molecular bioactivity for drug design," *Bioinformatics*, vol. 19, pp. 764-771, 2003.

[13] M. Kukar, "Transductive reliability estimation for medical diagnosis," *Artifical intelligence in medicine*, vol. 29, pp. 81 - 106, 2003.

[14] Q. Song and N. Kasabov, "TWRBF - Transductive RBF Neural Network with Weighted Data Normalization," *Lecture Notes in Computer Science*, vol. 3316, pp. 633-640, 2004.

[15] H. Zhu and O. Basir, "A K-NN Associated Fuzzy Evidential Reasoning Classifier with Adaptive Neighbor Selection," presented at Third IEEE International Conference on Data Mining, Melbourne, Florida, 2003.

[16] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman, "Missing value estimation methods for DNA microarrays," *Bioinformatics*, vol. 17, pp. 520-525, 2001.

[17] Y. Huang and Y. Li, "Prediction of protein subcellular locations using fuzzy k-NN method," *Bioinformatics*, vol. 20, pp. 21-28, 2004.

[18] D. J. Hand and V. Vinciotti, "Choosing k for two-class nearest neighbour classifiers with unbalanced classes,," *Pattern Recognition Letters*, vol. 24, pp. 9-10, 2003.

[19] P. Jönsson and C. Wohlin, "An Evaluation of k-Nearest Neighbour Imputation Using Likert Data," *Software Metrics, 10th International Symposium on (METRICS'04)*, pp. 108-118, 2004.

[20] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*: John Wiley and Sons, 1973.

[21] G. Enas and S. Choi, "Choice of the smoothing parameter and efficiency of k-nearest neighbor classification," *Computer and Maths with Applications*, vol. 12A, pp. 235-244, 1986.

[22] R. Short and K. Fukunaga, "The optimal distance measure for nearest neighbor classification," *IEEE Transactions on Information Theory*, vol. 27, pp. 622-627, 1981.

[23] Holmes and Adams, "A probabilistic nearest neighbour method for statistical pattern recognition," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 64, pp. 295-306, 2002.

[24] M. P. S. Brown, W. N. Grundy, D. Lin, N. Cristianini, C. W. Sugnet, T. S. Furey, M. A. Jr, and D. Haussler, "Knowledge-based analysis of microarray gene expression data by using support vector machines," *Proceedings of National Academy of Sciences*, vol. 97, pp. 262-267, 2000.

[25] S. Hirano, X. Sun, and S. Tsumoto, " On Similarity Measures for Cluster Analysis in Clinical Laboratory Examination Databases," presented at 26th Annual International Computer Software and Applications Conference, Oxford, England, 2002.

[26] H. Motoda, H. Liu, and L. Yu, "Feature extraction, selection, and construction," in *Handbook of data mining and knowledge discovery*, N. Ye, Ed.: Lawrence Erlbaum Associates, Inc., 2002, pp. 409-423.

[27] L. I. Kuncheva and L. C. Jain, "Nearest Neighbor clasifer: Simultaneous editing and feature selection," *Pattern Recognition Letters*, vol. 20, pp. 1149-1156, 1999.

[28] B. Sahiner, H.-P. Chan, D. Wei, N. Petrick, M. A. Helvie, D. D. Adler, and M. M. Goodsitt, "Image feature selection by a genetic algorithm: application to classification of mass and normal breast tissue," *Medical Physics*, vol. 23, pp. 1671-1684, 1996.

[29] Y. Chtioui, D. Bertrand, and D. Barba, "Feature selection by a genetic algorithm. Application to seed discrimination by artificial vision," *Journal of the Science of Food and Agriculture*, vol. 76, pp. 77-86, 1998.

[30] R. Leardi, "Application of a genetic algorithm to feature selection under full validation conditions and to outlier detection," *Journal of Chemometrics*, vol. 8, pp. 65-79, 1994.

[31] R. Leardi, "Genetic algorithms in feature selection," *Journal of Chemometrics*, vol. 15, pp. 559 - 569, 1996.

[32] Siedlecki and Sklansky, "A note of genetic algorithms for large-scale feature selection," *Pattern Recognition Letters*, vol. 10, 1989.

[33] Pudil, Novovicová, and Kittler, "Floating search methods in feature selection," *Pattern Recognition Letters*, vol. 15, 1994.

[34] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 153-158, 1997.

[35] J. F. Hair, R. L. Tatham, R. E. Anderson, and W. Black, *Multivariate Data Analysis*, 5th Edition ed: Prentice Hall, 2004.

[36] Holland, *Adaptation in natural and artificial systems*: The University of Michigan Press, Ann Arbor, 1975.

[37] D. Goldberg, *Genetic algorithms in search, optimization, and machine learning*: Addison-Wesley Publishing Company, 1989.

[38] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*: Springer - Verlag, 1994.

[39] C. L. Blake and C. J. Merz, "UCI Repository of machine learning databases," University of California, 1998.