Beyond Catastrophic Forgetting in Continual Learning: An Attempt with SVM

Diana Benavides-Prado [1]

Abstract

A big challenge in continual learning is avoiding catastrophically forgetting previously learned tasks. The possibility of improving existing knowledge whilst integrating new information has been much less explored. In this paper we describe a method that aims to improve the performance of previously learned tasks by refining their knowledge while new tasks are observed. Our method is an example of this ability in the context of Support Vector Machines for binary classification tasks, which encourages retention of existing knowledge whilst refining. Experiments with synthetic and real-world datasets show that the performance of these tasks can be continually improved by transferring selected knowledge, leading to the improvement on the performance of the learning system as a whole.

1. Introduction

Continual machine learning has become a very active area of research (Parisi et al., 2018; De Lange et al., 2020). The ability of supervised machine learning systems to learn continually is a fundamental property in domains such as object recognition, text classification and sentiment categorization, where examples from different classes may arrive at different points in time (Chen & Liu, 2016). Learning a group of tasks continually is particularly challenging in the context of deep neural networks. An outstanding challenge is that of *catastrophically forgetting* knowledge from previous tasks (French, 1999). Several studies have shown that maintaining the performance for previous tasks becomes more challenging as more tasks are learned. Previous research has proposed mechanisms to mitigate the effects of catastrophic forgetting in three settings: regularization-based methods to control the change of previously learned weights, methods that store or generate examples from previous tasks and replay these tasks while learning new ones, and methods to isolate or freeze weights learned on previous tasks (De Lange et al., 2020). The possibility of improving knowledge whilst integrating new information, which has been mentioned as a fundamental capacity of lifelong machine learning systems (Chen & Liu, 2016; 2018) and was also explored in previous studies on knowledge consolidation (Silver et al., 2015), has been much less studied in the context of continual learning.

We present a method that points in that direction in the simple context of binary classification tasks learned using Support Vector Machines (SVM). The method *transfers backward* knowledge to previous tasks, each of which is represented as an SVM model. Experiments on synthetic and real-world datasets of different number of tasks suggest that the performance of a continual learning system could be improved by explicitly refining previous SVM models. Our aim in this paper is to summarise ideas presented in Benavides-Prado et al. (2020) and to remark key points for the extension of these ideas to deep neural networks.

2. Previous Research

Continual learning has gained increasing interest in the context of deep neural networks. A recent survey categorised continual learning methods in three groups: regularization-based, memory replay methods and techniques for parameter isolation or freezing (De Lange et al., 2020). The problem of *catastrophic forgetting* was also studied in the context of knowledge consolidation in neural networks (Robins, 1995; 1996), which considered integrating new information into a learning system by rehearsing previous tasks. The possibility of improving knowledge whilst integrating new information was also explored in previous studies on knowledge consolidation (Silver et al., 2015). Silver, Yang and Li (2013) surveyed research in lifelong machine learning for supervised, unsupervised and reinforcement learning problems. Chen and Liu (2016, 2018) revived interest in lifelong machine learning systems and defined three core properties of these kinds of systems: 1) learning new tasks using knowledge from previous tasks, 2) learning continuously and incrementally, 2) retaining knowledge in a knowledge base.

[1]Auckland University of Technology, Auckland, New Zealand. Correspondence to: Diana Benavides-Prado <diana.benavides.prado@aut.ac.nz>.

Proceedings of the 37th International Conference on Machine Learning, Vienna, Austria, PMLR 108, 2020. Copyright 2020 by the author(s).

3. Background: SVM and Transferring Knowledge from Previous Tasks

An SVM function for classification describes an optimal hyperplane that separates examples of different classes with the smallest error, whilst maintaining a maximal margin (Vapnik, 1998). The optimal hyperplane for non-perfectly separable problems can be obtained by solving:

$$\begin{aligned} &\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n}\xi_i \\ &s.t. \begin{cases} \forall i \quad y_i(\mathbf{w}^\top\phi(x_i)+b) \geq 1-\xi_i \\ \forall i \quad \xi_i \geq 0 \end{cases} \end{aligned} \tag{1}$$

where the weight vector $\mathbf{w}$ and the bias b are learned parameters, $\boldsymbol{\xi} = \{\xi_1,...,\xi_n\}$ is a set of slack variables that allow some examples to violate the margin constraints, ϕ applies a transformation to a higher dimensional space and C is a trade-off parameter. This primal can be reformulated as the dual (Schölkopf & Smola, 2002):

$$\begin{aligned} &\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n}\alpha_i \ - \ \frac{1}{2}\sum_{i=1,j=1}^{n,n}\alpha_i\alpha_j y_i y_j K(x_i,x_j) \\ &s.t. \sum_{i=1}^{n} y_i\alpha_i = 0, \forall i \ 0 \leq \alpha_i \leq C \end{aligned} \tag{2}$$

The problem is to find a set of coefficients $\alpha_i, 1 \leq i \leq n$. Non-zero coefficients correspond to support vectors, training examples defining the boundary that separates the examples from different classes. K is a kernel function.

AccGenSVM (Benavides-Prado et al., 2017) identifies a subset $F \subseteq S$ of SVM models from which to transfer selected knowledge to a target SVM task. This subset is selected based on the relatedness of each $f(s) \in S$ and the target training data. Related source support vectors $x_i(s)$, $i \in \{1,\ldots,l\}$, for a $f(s) \in F$ are identified for each target training example $x_i(t)$, $i \in \{1,\ldots,n\}$. Fragments of source SVM models $f(s)$ are transferred by extracting $\alpha_i(s)$, $i \in \{1,\ldots,l\}$ coefficients that are later used to upper-bound coefficients to be learned on the target task. As a result, target training examples $x(t)$ which are more related to source support vectors $x(s)$ contribute more to the target objective to optimize. The learning problem in the dual representation is:

$$\begin{aligned} &\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n}\alpha_i \ - \ \frac{1}{2}\sum_{i,j=1}^{n,n}\alpha_i\alpha_j y_i y_j K(x_i,x_j) \\ &s.t. \sum_{i=1}^{n} y_i\alpha_i = 0, \forall i \ 0 \leq \alpha_i \leq C + c_i, c_i = \frac{|F|}{|S|}\sum_{k=1}^{s_i}{\alpha^{(s)}}_{ik} \end{aligned} \tag{3}$$

which uses training examples $(x_i, y_i), 0 \leq i \leq n$, to learn a weight vector $\mathbf{w}$. The upper-bound of a coefficient α_i is $C + c_i$. This constraint considers the original upper-bound C and c_i, an aggregation of $\alpha^{(\mathbf{s})} = \{\alpha_1^{(s)},\ldots,\alpha_s^{(s)}\}$ coefficients transferred from s_i source support vectors related to a target $x_i^{(t)}$ under evaluation. A factor $|F|/|S|$ accounts for the number of $f^{(s)}$ contributing to the target task.

4. Knowledge Improvement of Previous Tasks

AccGenSVM proposed to transfer selected support vectors from a set of source models S to a target task. As a result of this transfer, tuples of the form:

$$Z = [(x^{(s)}, y^{(s)}, \alpha^{(s)}), (x^{(t)}, y^{(t)}, \alpha^{(t)})] \tag{4}$$

that match a particular source support vector $(x^{(s)}, y^{(s)}, \alpha^{(s)})$ with a related target support vector $(x^{(t)}, y^{(t)}, \alpha^{(t)})$, which were involved in transfer while transferring forward using Eq. 3, can be identified. A pair of this kind represents a subspace of knowledge that is shared by $f^{(s)}$ and $f^{(t)}$. Exploiting these tuples could be potentially useful for refining existing $f^{(s)}$.

HRSVM, which stands for Hypothesis Refinement using SVM, is a method that transfers backward selected knowledge from an SVM hypothesis or model trained recently to related source SVM models. The aim is to emphasise refinement towards the knowledge shared between the source and the recent target, particularly when knowledge sharing occurs across different subspaces of these models. A local function that uses one of the tuples in Eq. 4 as training examples is learned as a representation of a subspace of shared knowledge. Such function acts as intermediate knowledge (Jonschkowski et al., 2015). This knowledge is stored only temporarily during refinement. We propose to learn as many of these functions as tuples can be collected while transferring to a target task.

To avoid catastrophic forgetting, **HRSVM** encourages retention by using ν-SVM (Schölkopf et al., 2000). ν-SVM was proposed as an alternative to the C-SVM problem for classification. The parameter ν has three properties: 1) it acts as an upper bound on the fraction of margin errors, 2) it acts as a lower bound on the fraction of support vectors, 3) for *i.i.d.* samples and with analytic and non-constant kernels, ν equals both the fraction of support vectors and the fraction of errors. In our case, the desired effect is to control the margin size by: 1) discouraging compression, *i.e.* maximising the number of examples selected as support vectors in the refined model, 2) refining appropriately on the training (support vectors) set, *i.e.* minimising the training error. The need to control for the margin size whilst transferring was pointed out by Aytar et al. (2011), for cases of transfer to a target task. We formulate a modified ν-SVM problem that regularizes the distance between the parameter vector $\mathbf{w}$ to be learned and a new parameter vector derived from representations of subspaces of shared knowledge, $\mathbf{w^{(d)}}$. Our (soft-margin) ν-SVM problem is:

$$\min_{\mathbf{w},\boldsymbol{\xi},\rho} \frac{1}{2}\|\mathbf{w}-\Gamma\mathbf{w^{(d)}}\|_2^2 - \nu\rho + \frac{1}{l}\sum_{i=1}^{l}\xi_i$$
$$s.t. \begin{cases} \forall i \quad y_i(\mathbf{w}^\top\phi(x_i)+b) \geq \rho - \xi_i \\ \forall i \quad \xi_i \geq 0, \rho \geq 0 \end{cases} \tag{5}$$

which maximises for l support vectors on the source SVM, $(x_i, y_i), 0 \leq i \leq l$, by learning a weight vector $\mathbf{w}$ of the function separating these support vectors. A set of slack variables $\boldsymbol{\xi} = \{\xi_1,...,\xi_n\}$ allows some examples to violate the margin constraints, whilst ρ is a learned parameter. Note that for $\boldsymbol{\xi} = 0$, *i.e.* a hard margin, the first constraint implies that the two classes are separated by the margin $2\rho/||\mathbf{w}||$. Therefore, both $\mathbf{w}$ and ρ define the margin size. The proposed problem is inbetween regularizing fully between a target and a source such as in A-SVM (Yang et al., 2007), and learning with privileged information using SVM+ (Wang & Hebert, 2016) which learns these two sets in parallel. We fix the parameter vector $\mathbf{w^{(d)}}$ while optimizing Eq. 5, since it is derived before solving (minimising) this problem, as will be explained in the dual formulation. In practice $\mathbf{w^{(d)}}$ are multiple sets of parameters, one for each subspace of shared knowledge. The influence of $\mathbf{w^{(d)}}$ is controlled by Γ. This value can be set experimentally by, for example, grid-search over a range of values. The dual objective of **HRSVM** is:

$$\max_{\boldsymbol{\alpha}} -\frac{1}{2}\sum_{i,j=1}^{l,l}\alpha_i\alpha_j y_i y_j K(x_i,x_j)$$
$$-\Gamma\sum_{i,k=1}^{l,2m}\alpha_i y_i \alpha_k^{(d)} y_k^{(d)} K(x_k^{(d)},x_i) \tag{6}$$
$$s.t. \ \forall i \ 0 \leq \alpha_i \leq 1/l, \sum_{i=1}^{l} y_i\alpha_i = 0, \sum_{i=1}^{l}\alpha_i \geq \nu$$

with $\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j K(x_i,x_j)$ the space of l source support vectors in the current $f^{(s)}$ and $\sum_{i,k=1}^{l,2m}\alpha_i y_i \alpha_k^{(d)} y_k^{(d)} K(x_k^{(d)},x_i)$ the representations of subspaces of shared knowledge between source support vectors in $f^{(s)}$ and target support vectors in $f^{(t)}$. Here, each $(\alpha_k^{(d)}, y_k^{(d)}, x_k^{(d)})$, with $1 \leq k \leq 2m$, are terms extracted from m functions $f^{(d)}$ learned with one-class SVM (Schölkopf et al., 2001). Each of these functions uses one tuple of the form in Eq. 4, $Z = \{(x^{(s)}, y^{(s)}, \alpha^{(s)}), (x^{(t)}, y^{(t)}, \alpha^{(t)})\}$ as training examples. Although other mechanisms may be applicable, one-class SVM makes the inclusion of this intermediate representation into the final objective straightforward. In binary classification tasks $Z = \{(x^{(s)}, y^{(s)}, \alpha^{(s)}), (x^{(t)}, y^{(t)}, \alpha^{(t)})\}$ should be such that $y^{(s)} = 1$ and $y^{(t)} = 1$ or $y^{(s)} = -1$ and $y^{(t)} = -1$, for transfer to occur among corresponding classes.

Chen et al. (2005) derived the maximal feasible value of $\nu = 2 * min(l_+, l_-)/l$, where l_+ and l_- correspond to the number of positive and negative examples, respectively. The maximal feasible value should be enforced, such that the maximal knowledge (number of source support vectors) is retained, whilst the second term in Eq. 6 drives refinement of $f^{(s)}$ towards the subspaces of shared knowledge. In sequential learning with **HRSVM**, ρ of the refined model (in Eq. 5) should be encouraged to be larger than the corresponding parameter of the current model. Appendix 1 provides theoretical analyses of the proposed method.

5. Experiments and Results

We generate two synthetic datasets, one of hyperplane problems and one of RBF concepts, composed of 500 tasks each. We also evaluate three real-world datasets. Each dataset is a learning system for which tasks are learned sequentially. For synthetic hyperplanes we randomly generate 500 problems of 100 features with values in the range $[0, 1]$, using an existing generator (Pedregosa et al., 2011). We generate $1{,}000$ random examples for each hyperplane problem. We repeatedly extract training (10%) and test samples (30%) without replacement for each problem, 30 times. We add 40% noise by shuffling training labels. We re-implement an existing method to generate synthetic RBF concepts (Bifet et al., 2010). We generate 100 centroids, defined by a random center of 100 features in the range $[0, 1]$, and a standard deviation in the same range. We then generate $1{,}000$ random examples for each RBF concept. We repeatedly extract training (10%) and test samples (30%) without replacement for each class, 30 times, and compose balanced binary classification problems of each RBF concept vs. rest. We add 40% noise by shuffling training labels. We also experiment with 20newsgroups (Mitchell, 1997), CIFAR-100 (Krizhevsky & Hinton, 2009), and a randomly selected ImageNet subset of 500 classes (Deng et al., 2009). For these datasets, we sample 10% training and 30% test sets, and compose binary classification tasks of each class vs. rest, 30 times. We experiment with DEN (Yoon et al., 2017), a recent competitive continual learning method. The number of hidden layers is set to 2 in all cases, with the following number of neurons: for the synthetic datasets 250 and 200 neurons, for ImageNet 500 and 250 neurons, for 20newsgroups 500 and 250 neurons, for CIFAR-100 $1{,}500$ and 500 neurons. For simplicity, all networks are FNN for classification tasks. After experimentation, we set the values of the parameters: $5{,}000$ maximum iterations, batch size of 500, learning rate of 0.001, L1 sparsity of 0.0001, L2 lambda of 0.0001, group LASSO lambda of 0.001, regularization lambda of 0.5, threshold for dynamic expansion of 0.1, threshold for split and duplication of 0.5. For the number of units of expansion, we experiment with: the default value of 10 and

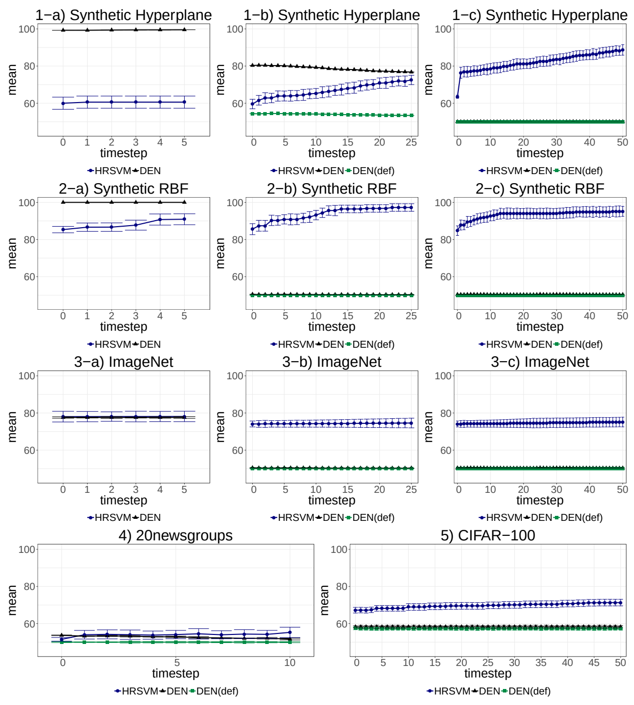

Figure 1. Mean classification accuracy at each timestep. 1-a), 2-a), 3-a): 10 tasks, 1-b), 2-b), 3-b): 50 tasks, 1-c), 2-c), 3-c): 100 tasks. DEN is a network where the number of units of expansion equals the number of classes. DEN(def) is a network with the default number of units of expansion (10). Error bars show 95% confidence intervals.

the number of tasks. We test different number of tasks for synthetic hyperplane, synthetic RBF and ImageNet.

Figure 1 shows results of DEN and **HRSVM**. t_0 is the performance after half of the tasks are learned. Since the number of tasks to learn must be pre-determined in DEN, the performance at t_0 varies depending on the number of tasks to be learned. **HRSVM** does not require the number of tasks as an input. For synthetic datasets, DEN achieves better performance than **HRSVM** while learning 10 tasks (1-a, 2-a, 3-a). This is constant across timesteps. **HRSVM** achieves slightly increasing performance. For 50 tasks (1-b, 2-b, 3-b), maintaining a single DEN network is more challenging. Yoon et al. (2017) remarked the challenge of setting an appropriate network capacity for a large number of tasks. For 100 tasks (1-c, 2-c, 3-c) DEN faces more challenges to learn these tasks and the performance remains constant. After experimentation we encountered that, the larger the number of tasks, the more difficult for DEN to perform well even on the training examples. For ImageNet, 20newsgroups and CIFAR-100 the performance of both **HRSVM** and DEN remains mostly constant over time. **HRSVM** benefits from a larger number of tasks, as

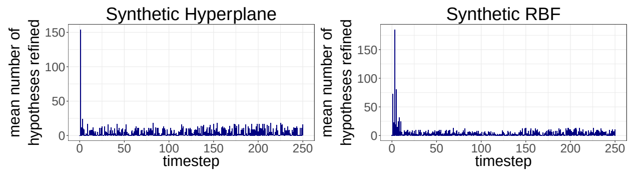

Figure 2. Number of SVM models refined at each timestep.

shown in Figure 2 for all tasks in the synthetic datasets, in the proposed simple setting of maintaining a separate model for each task.

6. Limitations and Discussion

HRSVM attempts to improve the performance of previous binary classification tasks by exploiting knowledge collected in more recent tasks. The method relies heavily on the well-known ν-SVM. **HRSVM** is currently limited to the same input types, *i.e.* the same feature representation for all tasks, and to the same output types, *i.e.* binary classification tasks for all the tasks expected by the learning system. The method does not currently support rehearsal of a task except by the explicit refinement of knowledge of that task using **HRSVM**. The accumulation of practice is given by the sequential learning of multiple binary classification tasks received one after the other, rather than as a continuum that does not distinguish tasks or as a curriculum of tasks. Each task is stored as an individual model, which simplifies the problem of continual deep learning methods which attempt to maintain a single network for all the tasks. However, the proposed method seems to align with some of the strategies of well-known deep continual learning methods as categorised by De Lange et al. (2020): 1) a replay strategy to use or generate training examples of previous tasks, which in **HRSVM** corresponds to the support vectors of these tasks; 2) a regularization strategy to learn new parameters constrained to the parameters of previous tasks, which in **HRSVM** is achieved by Eq. 5.

HRSVM pursues the goal of correcting errors on previous tasks. We believe that this could be potentially achieved in the context of deep neural networks by combining ideas from memory replay methods, to store or generate some correctly and incorrectly classified examples from previous tasks, and regularization-based methods to ensure that previous knowledge is not corrupted. Methods such as OWM (Zeng et al., 2019), which keep summary information of previous training examples, could be adapted to ensure orthogonality with weights of previous tasks whenever these contribute to correct classification of training examples whilst allowing *non-orthogonality* to incorrectly classified examples. This is part of our current research.

References

Abu-Mostafa, Y. S. Hints. *Neural Computation*, 7(4):639–671, 1995.

Aytar, t. and Zisserman, A. Tabula Rasa: Model Transfer for Object Category Detection. In *ICCV*, pp. 2252–2259. IEEE, 2011.

Benavides-Prado, D., Koh, Y. S., and Riddle, P. AccGenSVM: Selectively Transferring from Previous Hypotheses. In *IJCAI*, pp. 1440–1446, 2017.

Benavides-Prado, D., Koh, Y. S., and Riddle, P. Towards knowledgeable supervised lifelong learning systems. *Journal of Artificial Intelligence Research*, 68: 159–224, 2020.

Bifet, A., Holmes, G., Kirkby, R., and Pfahringer, B. MOA: Massive Online Analysis. *JMLR*, 11(May):1601–1604, 2010.

Chen, P.-H., Lin, C.-J., and Schölkopf, B. A Tutorial on ν-Support Vector Machines. *Applied Stochastic Models in Business and Industry*, 21(2):111–136, 2005.

Chen, Z. and Liu, B. Lifelong Machine Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 10(3):1–145, 2016.

Chen, Z. and Liu, B. Lifelong Machine Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 12(3):1–207, 2018.

De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., and Tuytelaars, T. A Continual Learning Survey: Defying Forgetting in Classification Tasks. *arXiv preprint arXiv:1909.08383*, 2020.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A Large-Scale Hierarchical Image Database. `http://image-net.org/download-features`, 2009.

French, R. M. Catastrophic Forgetting in Connectionist Networks. *Trends in Cognitive Sciences*, 3(4):128–135, 1999.

Jonschkowski, R., Höfer, S., and Brock, O. Patterns for Learning with Side Information. *arXiv preprint arXiv:1511.06429*, 2015.

Krizhevsky, A. and Hinton, G. Learning Multiple Layers of Features from Tiny Images. *Technical Report*, 2009.

Mitchell, T. M. *Machine Learning*, volume 45. Burr Ridge, IL: McGraw Hill, 1997.

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. Continual Lifelong Learning with Neural Networks: A Review. *arXiv preprint arXiv:1802.07569*, 2018.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine Learning in Python - Maximum Margin Separating Hyperplane, 2011.

Robins, A. Catastrophic Forgetting, Rehearsal and Pseudorehearsal. *Connection Science*, 7(2):123–146, 1995.

Robins, A. Consolidation in Neural Networks and in the Sleeping Brain. *Connection Science*, 8(2):259–276, 1996.

Schölkopf, B. and Smola, A. J. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.

Schölkopf, B., Smola, A. J., Williamson, R. C., and Bartlett, P. L. New Support Vector Algorithms. *Neural Computation*, 12(5):1207–1245, 2000.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. Estimating the Support of a High-Dimensional Distribution. *Neural Computation*, 13(7): 1443–1471, 2001.

Silver, D. L., Yang, Q., and Li, L. Lifelong Machine Learning Systems: Beyond Learning Algorithms. In *AAAI Spring Symposium: Lifelong Machine Learning*, volume 13, pp. 49–53, 2013.

Silver, D. L., Mason, G., and Eljabu, L. Consolidation using Sweep Task Rehearsal: Overcoming the Stability-Plasticity Problem. In *Canadian Conference on Artificial Intelligence*, pp. 307–322. Springer, 2015.

Vapnik, V. *Statistical Learning Theory*. Wiley, New York, 1998.

Wang, Y.-X. and Hebert, M. Learning by Transferring from Unsupervised Universal Sources. In *AAAI*, pp. 2187–2193, 2016.

Yang, J., Yan, R., and Hauptmann, A. G. Cross-Domain Video Concept Detection using Adaptive SVMs. In *ACM International Conference on Multimedia*, pp. 188–197. ACM, 2007.

Yoon, J., Yang, E., et al. Lifelong Learning with Dynamically Expandable Networks. *arXiv:1708.01547*, 2017.

Zeng, G., Chen, Y., Cui, B., and Yu, S. Continuous Learning of Context-dependent Processing in Neural Networks. *Nature Machine Intelligence*, pp. 364–372, 2019.

Appendix 1 - Theoretical Properties of HRSVM

We use the VC-dimension as a framework to analyse the theoretical properties of HRSVM. We start by recalling the well-known VC-dimension bounds for SVM classifiers (Vapnik, 1998):

Theorem 1. *Let vectors $x \in X$ belong to a sphere of radius R. Then the set of Δ-margin separating hyperplanes has VC-dimension $VC\text{-}dim(\mathbb{H})$ bounded by:*

$$VC\text{-}dim(\mathbb{H}) \leq min\left(\frac{R^2}{\Delta^2}, d\right) + 1 \tag{7}$$

where d is the dimensionality of the problem, R the radius of a sphere enclosing the training examples and Δ the size of the margin. For sufficiently large d, and since $R = 1$ without loss of generality, the only way to alter these bounds is by changing the margin Δ.

We derive VC-dimension bounds of the proposed **HRSVM** algorithm inspired by the research of Abu-Mostafa (1995). This framework studies the use of *hints* or additional information to encourage higher generalisation of a chosen function $f^{(s)}$. These hints can be, for example, invariances of the training examples to transformations such as rotations, translations, etc. The knowledge we propose to collect as $Z = \{(x^{(s)}, y^{(s)}, \alpha^{(s)}), (x^{(t)}, y^{(t)}, \alpha^{(t)})\}$, is an example of such hints.

First, let $D = \bigcup_c D_c$ be the set of examples D partitioned into D_c classes. In our case, each D_c is composed of one or more x examples and their corresponding $x^{(d)}$, such that $D_c = \{(x_1, x^{(d)}), \ldots, (x_l, x^{(d)}) \in Z\}$, with Z in Eq. 4. The value of the desired function $f^{(s)}$ is constant in each of these classes (Abu-Mostafa, 1995), *i.e.* for all of the $(x_i, x_i^{(d)}) \in D_c$ the value of a learned $f^{(s)}$ is more likely to be constant. When D_c contains a single $(x, x^{(d)})$ then $f^{(s)}(x) \simeq f^{(d)}(x^{(d)})$, *i.e.* D_c is only marginally useful since it chooses an $f^{(s)}$ that is likely to be constant on x only, given $x^{(d)}$. The extent to which this is useful depends on the *quality* of $x^{(d)}$ for the given task. At the other extreme, if a given D_c contains all pairs $(x, x^{(d)})$ then this hint is extremely useful since it denotes that $f^{(s)}$ is constant at D, with $f^{(s)} = 1$ or $f^{(s)} = -1$. In more realistic cases where a given D_c contains some $(x, x^{(d)})$, an $f^{(s)}$ to be learned will be more likely to be constant at these x points, with $f^{(s)}(x) = 1$ or $f^{(s)}(x) = -1$. Therefore the higher the chance of learning a decision boundary that fits these examples well. These x will have a higher chance of becoming margin support vectors with $0 < \alpha < C$.

Abu-Mostafa (1995) demonstrated that the VC-dimension of a hypotheses set $\mathbb{H}$ conditioned on a hypotheses set $\mathbb{G}$ given by hints $D = \bigcup_c D_c$, such as for example invariance hints, is such that $VC\text{-}dim(\mathbb{H}|\mathbb{G}) \leq VC\text{-}dim(\mathbb{H})$, as follows:

Theorem 2. *Let $VC\text{-}dim(\mathbb{H}|\mathbb{G})$ be the VC-dimension of a set of hypotheses $\mathbb{H}$ conditioned on a set of hypotheses $\mathbb{G}$ given by $D = \bigcup_c D_c$, where for each $\hbar \in \mathbb{H}$ either $\hbar$ satisfies $\mathbb{G}$ or does not satisfy it, i.e. for all $(x, x^{(d)}) \in D_c$ then $\hbar(x)$ is constant. Since the set of hypotheses that satisfies $\mathbb{G}$ is $\hat{\mathbb{H}}$:*

$$\hat{\mathbb{H}} = \{\hbar \in \mathbb{H} | \textit{for all pairs } (x, x^{(d)}) \in D_c \textit{ then } \hbar(x) \textit{ is constant}\} \tag{8}$$

Then:

$$VC\text{-}dim(\mathbb{H}|\mathbb{G}) \leq VC\text{-}dim(\mathbb{H}) \tag{9}$$

Proof. Since $\hat{\mathbb{H}} \subseteq \mathbb{H}$, then $VC\text{-}dim(\mathbb{H}|\mathbb{G}) \leq VC\text{-}dim(\mathbb{H})$. □

Nontrivial hints such as invariance hints lead to a substantial reduction from $\mathbb{H}$ to $\hat{\mathbb{H}}$, and as a result potentially $VC\text{-}dim(\mathbb{H}|\mathbb{G}) < VC\text{-}dim(\mathbb{H})$ (Abu-Mostafa, 1995).

We now derive specific bounds encouraged by our SVM refinement method as follows:

Theorem 3. *Let $D = \bigcup_c D_c$ be the training examples D partitioned into D_c classes, where each D_c is composed of one or more x examples and a corresponding $x^{(d)}$, such that $D_c = \{(x_1, x_1^{(d)}), \ldots, (x_l, x_l^{(d)})\}$ where both $\{x_1, \ldots, x_l\}$ and $\{x_1^{(d)}, \ldots, x_l^{(d)}\} \in Z$, with Z in Eq. 4. Let $f^{(s*)}$ be a function learned using Eq. 6, such that $f^{(s*)}$ is desired to be constant for a given D_c. Let $\mathbb{H}$, $\mathbb{G}$, $VC\text{-}dim(\mathbb{H})$ and $VC\text{-}dim(\mathbb{H}|\mathbb{G})$ as defined in Theorem 2. Let $\mathbf{w}^*$ be the weight vector learned for $f^{(s*)}$. Let $\Delta^* = 2/||\mathbf{w}^*||$ be the corresponding margin. Let $\mathbf{w}$ be the weight vector and $\Delta = 2/||\mathbf{w}||$ the margin of the existing $f^{(s)}$, learned using a regular SVM (Vapnik, 1998) or **HRSVM**. Since:*

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \phi(x_i) \tag{10}$$

and:

$$\mathbf{w}^* = \sum_{i=1}^{l^*} \alpha_i^* y_i \phi(x_i) \tag{11}$$

and:

$$l^* \leq l \tag{12}$$

For:

$$\boldsymbol{\alpha_m} = \{\alpha | 0 < \alpha < C\}, \boldsymbol{\alpha_b} = \{\alpha | \alpha = C\}, \quad \boldsymbol{\alpha_0} = \{\alpha | \alpha = 0\} \tag{13}$$

and:

$$\begin{gathered}\boldsymbol{\alpha^*_m} = \{\alpha^* | 0 < \alpha^* < C\}, \boldsymbol{\alpha^*_b} = \{\alpha^* | \alpha^* = C\}, \\ \boldsymbol{\alpha^*_0} = \{\alpha^* | \alpha^* = 0\}\end{gathered} \tag{14}$$

With:

$$|\boldsymbol{\alpha^*_m}| \geq |\boldsymbol{\alpha_m}| \tag{15}$$

Then:

$$\Delta^* \geq \Delta \tag{16}$$

and, by Theorem 3 in Vapnik (1998) :

$$VC\text{-}dim(\mathbb{H}|\mathbb{G}) \leq VC\text{-}dim(\mathbb{H}) \tag{17}$$

Proof. Since by $D = \bigcup_c D_c$, $f^{(s*)}$ is more likely to be constant on several $x \in D_c$, more α^* will be forced to have values between 0 and C, *i.e.* to lie exactly on the margin. Therefore, with $l^* \leq l$, then $||\mathbf{w}^*|| \leq ||\mathbf{w}||$. Since $\Delta^* = 2/||\mathbf{w}^*||$ and $\Delta = 2/||\mathbf{w}||$, then necessarily $\Delta^* \geq \Delta$. Furthermore, since from Eq. 7.15 in Schölkopf et al. (2000):

$$\rho = \frac{1}{2s}\Bigg(\sum_{x \in S_+} \sum_j \alpha_j y_j K(x, x_j) - \sum_{x \in S_-} \sum_j \alpha_j y_j K(x, x_j)\Bigg) \tag{18}$$

with S_+ the set of positive examples with $0 < \alpha < 1$ and S_- the set of negative examples with $0 < \alpha < 1$, where $|S_+| = |S_-| = s$. By requiring that $\rho^* \geq \rho$ necessarily the set $\{S_+, S_-\}$ has to be bigger (*i.e.* more training examples are support vectors lying exactly on the margin) or their difference is larger. In the first case, $||w^*|| < ||w||$, and with $\rho^* \geq \rho$ then $\Delta^* > \Delta$. In the second case, at least $||w^*|| = ||w||$, and therefore at least $\Delta^* = \Delta$. $\square$

The extent to which $D = \bigcup_c D_c$ will be beneficial for refining an $f^{(s)}$ will be driven by how appropriate are tuples in Eq. 4. This will highly depend on the relatedness of the underlying distributions of the source $f^{(s)}$ and the target $f^{(t)}$.