Wael Emara, Mehmed Kantardzic Marcel Karnsted, Kai-Uwe Sattler, Dirk Habich, Wolfgang Lehner

An Approach for Incremental Semi-supervised SVM

Erstveröffentlichung in / First published in:

Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007). Omaha, 28.-31.10.2007. IEEE, S. 539-544. ISBN 978-0-7695-3019-2

DOI: https://doi.org/10.1109/ICDMW.2007.106

Final edited form was published in "Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007), Omaha, 2007". S. 539-544. ISBN 978-0-7695-3019-2.
https://doi.org/10.1109/ICDMW.2007.106

An Approach for Incremental Semi-supervised SVM

Wael Emara and Mehmed Kantardzic
Data Mining Laboratory
Department of Computer Science and Computer Engineering
University of Louisville, Louisville KY 40292
waemar01@louisville.edu, mmkant01@louisville.edu

Abstract

In this paper we propose an approach for incremental learning of semi-supervised SVM. The proposed approach makes use of the locality of radial basis function kernels to do local and incremental training of semi-supervised support vector machines. The algorithm introduces a sequential minimal optimization based implementation of the branch and bound technique for training semi-supervised SVM problems. The novelty of our approach lies in the introduction of incremental learning techniques to semi-supervised SVMs.

1. Introduction

The machine learning and data mining communities have recently had new research challenges imposed by the massive growth in information technology. Data is being generated continuously and knowledge is always needed to be extracted from these streaming data. Extracting knowledge from data streams requires incremental learning approaches which are concerned with constantly enhancing the available learning machines using the continuously collected data without retraining on all the available data. An important aspect of data streams is that the labeling process of training data is a costly and time consuming process. On the other hand, vast amounts of unlabeled data are collected all the time. Hence, developing incremental learning approaches that can make use of labeled and unlabeled samples (semi-supervised) during the learning process is of great benefit.

Support vector machines (SVMs) [19] are powerful and popular machine learning tools due to their good generalization performance. As a result, researchers have been trying to extend the paradigm of SVMs (originally supervised learning) to be able to handle semi-supervised and incremental learning problems.

The basic idea of SVMs is to find the maximum margin classifier for the labeled training samples. Formally, given a labeled training dataset $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$, where $\mathbf{x}_i \in \mathbb{R}^n$ and $y_i \in \{+1, -1\}$, the problem is to find the solution of the optimization problem

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{i=m} \ell_l((\mathbf{w}, b); (\mathbf{x}_i, y_i)) \qquad (1)$$

where b is a bias term and

$$\ell_l((\mathbf{w}, b); (\mathbf{x}_i, y_i)) = max\{0, 1 - y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b)\} \qquad (2)$$

is the loss function with subscript l denoting the use of labeled training data set. This is basically a convex optimization problem (quadratic programming problem) that can be solved using standard quadratic programming (QP) solvers.

In semi-supervised SVM (S^3VM), the training data consists of labeled and unlabeled samples. The goal is to find the maximum margin classifier using both labeled and unlabeled samples. This is formulated as follows: Given a mixed labeled/unlabeled dataset $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m} \bigcup \{\mathbf{x}_i\}_{i=m+1}^{m+k}$ where $\mathbf{x}_i \in \mathbb{R}^n$, $y_i \in \{+1, -1\}$, m and k are the number of labeled and unlabeled samples, respectively. The semi-supervised problem is to find the solution of

$$\begin{aligned} \min_{\mathbf{w}, \mathbf{y}_u} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \quad + \quad C\sum_{i=1}^{i=m} \ell_l((\mathbf{w}, b); (\mathbf{x}_i, y_i)) \\ & + \quad C^*\sum_{i=m+1}^{i=m+k} \ell_u((\mathbf{w}, b); (\mathbf{x}_i)) \end{aligned} \qquad (3)$$

where u denotes the unlabeled samples

$$\ell_u((\mathbf{w}, b); (\mathbf{x}_i)) = \max_{y_i \in \{-1, +1\}} \{0, 1 - y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b)\}. \qquad (4)$$

and $\mathbf{y}_u = [y_{m+1} \cdots y_{m+n}]$ is the vector of labels of all the unlabeled samples. The solution of (3) will result in finding the optimal separating hyperplane $\mathbf{w}$ and the labels assigned to the unlabeled samples in $\mathbf{y}_u$.

Final edited form was published in "Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007), Omaha, 2007". S. 539-544. ISBN 978-0-7695-3019-2.
https://doi.org/10.1109/ICDMW.2007.106

The S^3VM optimization problem in (3) is non-convex due to the non-convexity of ℓ_u, which is the reason that makes the S^3VM optimization a challenging problem. Bennett and Demiriz [1] were the first to address this problem. They used 1-norm of w and reformulated the problem so as to be operable by mixed-integer programming techniques. They showed results for data sets with no more than 50 unlabeled points. Joachims [11] introduced another approach to solve S^3VM with the text classification problem in mind. The novelty of that work lies in the capability to handle large number of unlabeled samples and the introduction of another constraint, *balancing constraint*, which makes sure that the algorithm avoids degenerate solutions. This condition has been considered in many of the subsequent works on S^3VM [6, 17]. Other approaches attempted to use off the shelf optimization techniques. ∇S^3VM introduced in [7] uses gradient descent to solve the problem. However, due to the non-differentiablity of the loss functions they proposed using smooth versions of both ℓ_l and ℓ_u. Also global optimization techniques such as continuation methods [5], deterministic annealing methods [17], and branch and bound methods [6] have also been used recently. Other techniques tried to overcome the main difficulty of optimizing S^3VM, non-convexity of ℓ_u in (4). In CCCP [8], a symmetric-ramp loss function that is then analyzed into convex and concave components and then a convex-concave optimization is applied. Fung and Mangasarian [9] used the same concept of resolving the non-convex optimization function, however their work is restricted to linear S^3VM. Transforming the non-convex S^3VM problem into a convex semi-definite programming problem [2, 20] is another way to overcome the non-convexity issue.

Incremental learning of SVMs has attracted considerable attention due to the need to use SVMs with very large data sets. Syed [18] proposed one of the earliest attempts to solve SVM incrementally. This approach works by retraining the SVM using a new batch of data along with the support vectors from the past SVM. An improvement was proposed in [16] so as to make the algorithm capable of dealing with the problem of concept drifting basically by making the cost of error on the old support vectors more costly than the cost of error on the new data. Fung and Mangasarian [10] proposed a fast and simple approach for incremental SVMs that is based on modifying the current linear SVM by removing old data and adding new data. Kivinen [12] considered incremental learning in a Reproducing Kernel Hilbert Space using gradient descent within a feature space. LISVM introduced in [15] is an approach that exploits the locality of the RBF kernels to update the current SVM by only considering a subset of the support candidates in the neighborhood of the new sample.

In this work, we propose an approach for incremental learning of S^3VM. The novelty of our work is that it addresses two situations encountered in real life applications; It is always desirable to enhance the currently available classifiers using newly observed data without going through the tedious and impractical process of retraining over all the available data. On the other hand, the newly observed data can be a mixture of labeled and unlabeled samples. The unlabeled samples constitutes the majority of the newly observed data due to the high cost and long time needed to manually label training data. Hence, the benefit of the proposed approach is quite obvious.

The rest of the paper will be organized as follows: In Section 2, we review some related works which includes Sequential Minimal Optimization (SMO) training of SVMs [14] and branch and bound for S^3VM training [6]. Section 3 will discuss the proposed approach. Experimental results and conclusion will be presented in Section 4 and Section 5, respectively.

2. Related work

The basic formulation of SVMs in (1) and (2) is a compact one that is used to show a closed form for the objective function being optimized. A more conventional form is to solve

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i \tag{5}$$

subject to the constraints

$$y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geqslant 1-\xi_i, \quad \xi_i \geqslant 0, \quad i = 1, 2, \cdots m. \tag{6}$$

Introducing the Lagrange multipliers α_i for each of the constraints in (6) the dual problem is obtained;

$$\min_{\alpha_i} \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j\rangle \alpha_i \alpha_j - \sum_{i=1}^{m}\alpha_i \tag{7}$$

subject to

$$\sum_{i=1}^{m} y_i \alpha_i = 0, \quad 0 \leqslant \alpha_i \leqslant C, \quad i = 1, 2, \cdots m. \tag{8}$$

The solution is reached when all Karush-Kuhn-Tucker (KKT) conditions are being satisfied over all the training samples. KKT conditions are described as follows [14]:

$$\begin{aligned} \alpha_i = 0 &\Longrightarrow y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geqslant 1 \\ 0 < \alpha_i < C &\Longrightarrow y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) = 1 \\ \alpha_i = C &\Longrightarrow y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \leqslant 1 \end{aligned} \tag{9}$$

2.1. Sequential minimal optimization

Sequential Minimal Optimization (SMO) [14] is a simple algorithm to solve the QP problem arising in training

Final edited form was published in "Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007), Omaha, 2007". S. 539-544. ISBN 978-0-7695-3019-2.
https://doi.org/10.1109/ICDMW.2007.106

SVMs. It belongs to a family of algorithms that addresses training SVMs on large data sets by breaking the QP problem into smaller manageable ones [13]. SMO takes the idea of breaking the QP problem to its extreme by choosing to solve the smallest possible QP problem; At each iteration, two Lagrange multipliers are jointly optimized and the SVM is updated accordingly. The advantage of optimizing two Lagrange multipliers lies in the possibility to do it analytically which means fast.

Starting with $(\mathbf{x}_1, y_1, \alpha_1^{old})$ and $(\mathbf{x}_2, y_2, \alpha_2^{old})$ and considering the constraints in (8), it is clear that the space of possible values for α_1 and α_2 is actually, due to the inequality constraints, a square with side length C. Moreover, we can see that the summation constraint forces the values of α_1 and α_2 to lie on a diagonal line. These insights narrow down the space of the solution. The algorithm starts by finding the bounds on α_2^{new} which depends on the value $y_1 y_2$; If $y_1 y_2 = -1$, then the bounds on α_2^{new} are:

$$\begin{aligned} L &= max(0, \alpha_2^{old} - \alpha_1^{old}) \\ H &= min(C, C + \alpha_2^{old} - \alpha_1^{old}). \end{aligned} \tag{10}$$

If $y_1 y_2 = 1$, then the bounds on α_2^{new} are:

$$\begin{aligned} L &= max(0, \alpha_1^{old} + \alpha_2^{old} - C) \\ H &= min(C, \alpha_1^{old} + \alpha_2^{old}). \end{aligned} \tag{11}$$

Next, α_2^{new} is obtained by:

$$\alpha_2^{new} = \alpha_2^{old} - \frac{y_2(E_1 - E_2)}{\eta} \tag{12}$$

where η is the second derivative of the objective function (7) along the diagonal line:

$$\eta = 2\langle \mathbf{x}_1, \mathbf{x}_2 \rangle - \langle \mathbf{x}_1, \mathbf{x}_1 \rangle - \langle \mathbf{x}_2, \mathbf{x}_2 \rangle \tag{13}$$

and E_i is the error of the old SVM on the ith training sample. Then, α_2^{new} is clipped according to its bounds in (10) or (11).

$$\alpha_2^{new,clipped} = \begin{cases} H & \text{if } H \leqslant \alpha_2^{new} \\ \alpha_2^{new} & \text{if } L < \alpha_2^{new} < H \\ L & \text{if } \alpha_2^{new} \leqslant L \end{cases} \tag{14}$$

Finally, α_1^{new} is computed by

$$\alpha_1^{new} = \alpha_1^{old} + y_1 y_2 (\alpha_2^{old} - \alpha_2^{new,clipped}) \tag{15}$$

SMO provides two heuristics to select the Lagrange multipliers to be optimized. The first choice heuristic selects the first Lagrange multiplier by iterating over the whole training dataset until a multiplier violating the KKT conditions is found then the search start for the second Lagrange multiplier by initializing the second choice heuristic. When the first choice heuristic completes one pass through the entire training dataset, it starts iterating over the multipliers that are neither 0 or C (non-bound multipliers). Again after it finishes iterating over the non-bound multipliers, it makes another pass through the whole training dataset and so on. The second choice heuristic aims at choosing the second Lagrange multiplier that maximizes the step taken during the optimization. SMO proposes using $|E_1 - E_2|$ as an estimate for the step size. Therefore, a cached error E_i is stored for every non-bound multiplier which is then used by the second choice heuristic to choose the second Lagrange multiplier.

2.2. Branch and bound for S^{3}VM

Starting with the S^{3}VM formulation in (3), let us denote the objective function to be minimized by

$$\begin{aligned} F(\mathbf{w}, \mathbf{y}_u) = \frac{1}{2}\|\mathbf{w}\|^2 \; + \; C \sum_{i=1}^{i=m} \ell_l((\mathbf{w}, b); (\mathbf{x}_i, y_i)) \\ + \; C^* \sum_{i=m+1}^{i=m+k} \ell_u((\mathbf{w}, b); (\mathbf{x}_i)) \end{aligned} \tag{16}$$

where ℓ_l and ℓ_u are defined in (2) and (4), respectively. Hence, the S^{3}VM is to solve

$$\min_{\mathbf{w}, \mathbf{y}_u} \quad F(\mathbf{w}, \mathbf{y}_u). \tag{17}$$

For a fixed $\mathbf{y}_u$, the problem in (17) will come down to a standard SVM training problem. Using this idea, (17) is reformulated to be

$$\begin{aligned} &\min_{\mathbf{y}_u} \quad J(\mathbf{y}_u) \\ \text{where} \quad &J(\mathbf{y}_u) = \min_{\mathbf{w}} \; F(\mathbf{w}, \mathbf{y}_u). \end{aligned} \tag{18}$$

Therefore, the goal now is to find the binary vector $\mathbf{y}_u$ that minimizes J in (18).

Branch and bound for S^{3}VM [6] is an approach to solve (18) on a tree structure. Basically, each node on the tree represents an SVM. The root has the SVM trained using only the labeled samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$ and the leaves have different solutions for the S^{3}VM in (18). Each node inside the tree represents an SVM trained on all the labeled samples and some of the unlabeled ones.

The basic components of any branch and bound algorithm are the branching criterion, bounds estimates, and exploration techniques adopted in the algorithm. The details of these components depend on the problem being solved. In [6], the branching criterion actually answers the question of which unlabeled sample should be considered for the next optimization step. The most intuitive way to do this is to optimize using the unlabeled sample that we are

Final edited form was published in "Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007), Omaha, 2007". S. 539-544. ISBN 978-0-7695-3019-2.
https://doi.org/10.1109/ICDMW.2007.106

most confident about, which is the nearest unlabeled sample to the so far labeled samples.

The algorithm proposed using the value of the of the objective function J in (18) as an upper bound for leaf nodes while non-leaf nodes do not have upper bounds (assigned to $+\infty$). On the other hand, the proposed lower bound is based on the observation that the value of $F(\mathbf{w}, \mathbf{y}_u)$ in (16) after training using only a labeled data set is always smaller than the its value when trained using labeled-unlabeled dataset. This is due to the addition of the third term (involving C^*) in (16). Therefore, at each node the lower bound is determined by optimizing a standard SVM using the so far labeled samples at that node.

Finally, the algorithm used the depth first search to explore the tree structure. This was motivated by the choice of the upper bound, as reaching the leaves as often as possible is necessary for obtaining a tight bound which in turns helps the pruning process throughout the tree search.

3. Proposed approach

We propose an approach to deal with the incremental training of S^3VM using labeled-unlabeld (partly labeled) data sets. The scenario attempted assumes the existence of an S^3VM. We use SMO [14] for all the SVM training involved in the approach. Due to the continuous availability of training samples overtime, we need to improve the existing S^3VM using these newly arriving training samples. It is almost always the case that the new training samples are unlabeled. Furthermore, labeled training samples can be available later in time due to the high cost and time requirements for the human labeling process. Our approach assumes the newly arriving training samples to be partly labeled.

The use of kernels that are based on the notion of neighborhood, makes the influence of a support vector greatly confined to a limited area in the feature space [15, 3]. Therefore, improving an existing SVM by using new training samples does not require repeating the training process all over the whole available training samples again. However, it suffices to solve the SVM QP problem only in the neighborhood of the new training samples. To make use of this concept, the proposed approach uses the radial basis function (RBF) kernel. Therefore, in the proposed approach only the neighborhood of the newly arriving training sample is considered for updating the S^3VM.

Starting from an existing S^3VM and a new training sample (labeled-unlabeld), the approach begins by constructing a neighborhood to the new sample. This neighborhood will include the current training samples that are expected to be mostly influenced by the new training sample. Then the approach attempts to update the current S^3VM by solving QP in the neighborhood. However, due to the fact the new sample may be unlabeled, we encounter a S^3VM problem. An SMO based implementation of the branch and bound for S^3VM has been developed and used to solve the S^3VM problem in hand.

Using the branch and bound technique to solve the S^3VM problem in our approach was motivated by the good results presented in [6] and the fact that it attempts to find the global solution for the S^3VM objective function. Despite its limited ability to deal with large S^3VM problems, the branch and bound was suitable to invoke in our approach due to the limited size of the neighborhood. The branch and bound for S^3VM algorithm involves solving SVM problem at each node visited by the search procedure. It is not practical to solve an SVM problem from scratch at each node. As a result, an incremental SVM technique should be used through out the branch and bound training process. For our approach, we chose the SMO as the core SVM solver inside the branch and bound technique. An important property of SMO is that it works by solving the SVM QP problem on two Lagrange multipliers at a time and then iterate until all the Lagrange multipliers satisfy the KKT conditions. This property makes the SMO inherently incremental; For a new sample to update an existing SVM, its Lagrange multiplier is initialized to zero and added to the Lagrange multipliers of the SVM. Then the SMO is asked to start checking the KKT conditions starting with the new Lagrange multiplier. The result is an improved SVM that have been updated using the a new training sample. This inherent incremental property was also one of the motivations to use SMO.

During the course of solving S^3VM, labels are assigned to the non-labeled training samples. These labels are the best solution the S^3VM algorithm could provide at the time of training. However, these labels are not real (binding). So, during the construction of the neighborhood we remove the labels assigned to the originally unlabeled samples. By doing this, we give the S^3VM solver the opportunity to find the best solution possible without being confined to labels that has been assigned before and sometimes they may be erroneous. In fact, using this technique enhances the performance by allowing the algorithm to recover from labeling mistakes made in the past.

The neighborhood in our approach is constructed by using the Euclidean distance as a measure of closeness and its size is the number of samples inside it. To achieve a good performance, the proposed approach uses varying size neighborhood. At first, the neighborhood size is initialized and the updated S^3VM is obtained. Then the approach estimates the performance of the updated S^3VM by using a validation dataset. If the performance is not satisfactory the approach changes the size and repeats until accepted performance is achieved. The initial size of the neighborhood is estimated experimentally using a validation dataset. Although the incremental construction of the neighborhood,

Final edited form was published in "Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007), Omaha, 2007". S. 539-544. ISBN 978-0-7695-3019-2.
https://doi.org/10.1109/ICDMW.2007.106

adding one sample at a time, seems more intuitive, the use of an initial estimate saves a lot of computation cost due to the smaller number of neighborhoods that would be examined. Starting from the initial neighborhood size, changes are performed by adding a new neighboring sample to the original neighborhood. If the size needs to be changed again, a sample is removed from the original neighborhood. This process of alternatively using increasing and decreasing neighborhoods is repeated until the proper size is found.

The proposed approach can be summarized as follows:

1. Starting with S^3VM^{old}.
2. New partly labeled (labeled-unlabeled) sample arrives.
3. Construct an initial neighborhood to the new sample.
4. Remove labels of the originally unlabeled samples inside the neighborhood.
5. SMO based Branch and Bound technique is used to locally solve the incremental S^3VM problem on the neighborhood and S^3VM^{new} is obtained.
6. S^3VM^{new} is evaluated using a validation dataset.
7. If the performance of S^3VM^{new} is not satisfactory (The correct classification rate is less than a predefined threshold), change the neighborhood size and go back to step 3.
8. Otherwise, wait for another new sample and get back to step 1.

4. Experimental results

This section will show sample results of the proposed algorithm on sample data sets. Two data sets have been considered. FourClass dataset is a two dimensional two-class dataset from [4]. It has a total of 802 training samples in both classes. Also, we used MNIST dataset of hand written digits. We used the a dataset with a total of 2000 samples from the digits 0 and 9.

During the experiments each dataset is split into training dataset (8%), validation dataset for neighborhood size and other parameters choice (2%), testing dataset (2%), and the rest of the dataset is used to simulate the new training samples which are used to update the current S^3VM one at a time. 80% of the labels of the new training samples are removed to simulate a highly unlabeled dataset which is the case in the real world.

Figure 1 shows the results of the proposed approach on four successive new training samples located at different positions in the feature space. The square (yellow) and circular (pink) points represents the two classes. All the tan circled point are the training samples of the current S^3VM. The new training sample is the green one with the neighborhood region depicted around it. The average correct classification rate (CCR) for this dataset with and without varying neighborhood is 96.3% and 76.75% respectively.

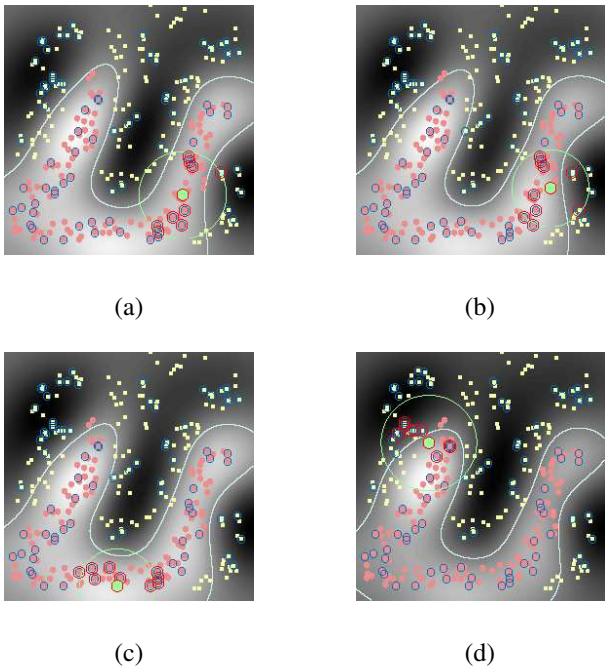


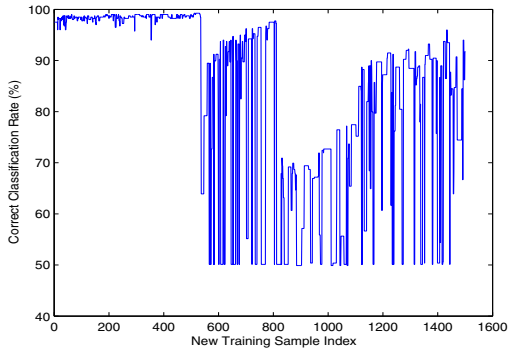
Figure 1. The performance of the algorithm on four different successive training samples

Figure 2 shows CCR for 1500 new training samples from the MNIST dataset. Figure 2 (b) depicts the good performance obtained by using varying neighborhood (average CCR is 98.68%). On the other hand, Figure 2 (a) shows much lower correct classification rate (average CCR is 83.26%) for the same dataset and same new training samples due to the use of a constant neighborhood size.

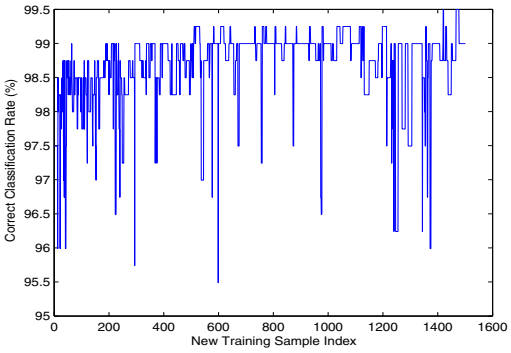
5. Conclusion and future work

The paper introduced a novel learning approach for incremental learning of semi-supervised support vector machines. The approach provided an implementation of the branch and bound technique to solve semi-supervised support vector machines that is based on sequential minimal optimization. This implementation makes use of the inherent incremental behavior of the sequential minimal optimization algorithm. Moreover, the proposed approach makes use of the locality of the radial basis function kernel which makes it possible to locally do incremental training of semi-supervised support vector machine problems. The preliminary experiments have shown promising results, yet

Final edited form was published in "Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007), Omaha, 2007". S. 539-544. ISBN 978-0-7695-3019-2.
https://doi.org/10.1109/ICDMW.2007.106



(a)



(b)

Figure 2. The correct classification rate for the MNIST 0 and 9 digits classification (a) Using constant neighborhood size and (b) Using the varying neighborhood size

we are currently working on extensive quantitative evaluation. For future work, we plan to investigate a formulation for a generalization error estimate that is based on semi-supervised dataset. This estimate should serve as a more efficient criterion to choose the neighborhood size, rather than using a validation dataset.

References

[1] K. P. Bennett and A. Demiriz. Semi-supervised support vector machines. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems 11*, Cambridge, MA, 1998. MIT Press.

[2] T. D. Bie and N. Cristianini. Convex methods for transduction. In S. Thrun, L. K. Saul, and B. Schölkopf, editors, *NIPS*. MIT Press, 2003.

[3] L. Bottou and V. Vapnik. Local learning algorithms. *Neural Comput.*, 4(6):888–900, 1992.

[4] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001.

[5] O. Chapelle, M. Chi, and A. Zien. A continuation method for semi-supervised svms. In W. W. Cohen and A. Moore, editors, *ICML*, pages 185–192. ACM, 2006.

[6] O. Chapelle, V. Sindhwani, and S. Keerthi. Branch and bound for semi-supervised support vector machines. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*. MIT Press, Cambridge, MA, 2007.

[7] O. Chapelle and A. Zien. Semi-supervised classification by low density separation. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AISTATS 2005)*, pages 57–64, 01 2005.

[8] R. Collobert, F. H. Sinz, J. Weston, and L. Bottou. Large scale transductive svms. *Journal of Machine Learning Research*, 7:1687–1712, 2006.

[9] G. Fung and O. Mangasarian. Semi-supervised support vector machines for unlabeled data classification, 2001.

[10] G. Fung and O. L. Mangasarian. Incremental support vector machine classification. In R. L. Grossman, J. Han, V. Kumar, H. Mannila, and R. Motwani, editors, *SDM*. SIAM, 2002.

[11] T. Joachims. Transductive inference for text classification using support vector machines. In I. Bratko and S. Dzeroski, editors, *Proceedings of ICML-99, 16th International Conference on Machine Learning*, pages 200–209, Bled, SL, 1999. Morgan Kaufmann Publishers, San Francisco, US.

[12] J. Kivinen, A. J. Smola, and R. C. Williamson. Online learning with kernels. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *NIPS*, pages 785–792. MIT Press, 2001.

[13] E. Osuna, R. Freund, and F. Girosi. An improved training algorithm for support vector machines. In J. Principe, L. Gile, N. Morgan, and E. Wilson, editors, *Neural Networks for Signal Processing VII — Proceedings of the 1997 IEEE Workshop*, pages 276 – 285, New York, 1997. IEEE.

[14] J. C. Platt. Fast training of support vector machines using sequential minimal optimization. *Advances in kernel methods: support vector learning*, pages 185–208, 1999.

[15] L. Ralaivola and F. d'Alché Buc. Incremental support vector machine learning: A local approach. In *ICANN '01: Proceedings of the International Conference on Artificial Neural Networks*, pages 322–330, London, UK, 2001. Springer-Verlag.

[16] S. Rüping. Incremental learning with support vector machines. In N. Cercone, T. Y. Lin, and X. Wu, editors, *ICDM*, pages 641–642. IEEE Computer Society, 2001.

[17] V. Sindhwani, S. S. Keerthi, and O. Chapelle. Deterministic annealing for semi-supervised kernel machines. In W. W. Cohen and A. Moore, editors, *Proceedings of the 23rd International Conference on Machine Learning*, pages 841–848. ACM, 2006.

[18] N. Syed, H. Liu, and K. Sung. Incremental learning with support vector machines. In *Proceedings of the Workshop on Support Vector Machines at the International Joint Conference on Articial Intelligence (IJCAI-99)*, 1999.

[19] V. N. Vapnik. *Statistical Learning Theory*. Wiley, September 1998.

[20] L. Xu and D. Schuurmans. Unsupervised and semi-supervised multi-class support vector machines. In M. M. Veloso and S. Kambhampati, editors, *AAAI*, pages 904–910. AAAI Press / The MIT Press, 2005.