Incremental and Decremental Proximal Support Vector Classification using Decay Coefficients

Amund Tveit, Magnus Lie Hetland and Håvard Engum

Department of Computer and Information Science,
Norwegian University of Science and Technology,
N-7491 Trondheim, Norway
{amundt,mlh,havare}@idi.ntnu.no

Abstract. This paper presents an efficient approach for supporting decremental learning for incremental proximal support vector machines (SVM). The presented decremental algorithm based on decay coefficients is compared with an existing window-based decremental algorithm, and is shown to perform at a similar level in accuracy, but providing significantly better computational performance.

1 Introduction

Support Vector Machines (SVMs) is an exceptionally efficient data mining approach for classification, clustering and time series analysis [5, 12, 4]. This is primarily due to SVMs highly accurate results that are competitive with other data mining approaches, e.g. artificial neural networks (ANNs) and evolutionary algorithms (EAs). In recent years tremendous growth in the amount of data gathered (e.g. user clickstreams on the web, in e-commerce and in intrusion detection systems), has changed the focus of SVM classifier algorithms to not only provide accurate results, but to also enable online learning, i.e. incremental and decremental learning, in order to handle concept drift of classes [2, 13].

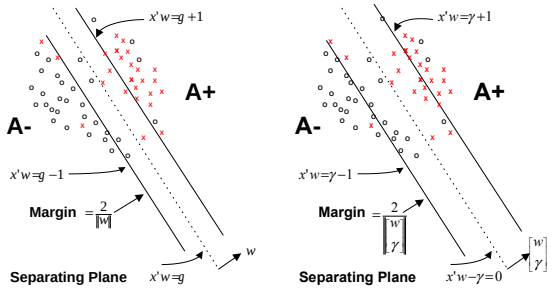
Fung and Mangasarian introduced the Incremental and Decremental Linear Proximal Support Vector Machine (PSVM) for binary classification [10], and showed that it was able to be trained extremely fast, i.e. with 1 billion examples (500 increments of 2 million) in 2 hours and 26 minutes on relatively low-end hardware (400 MHz Pentium II). This has later been extended to support efficient support of incremental multicategorical classification [16]. Proximal SVMs has also been shown to perform at a similar level of accuracy as regular SVMs and at the same time being significantly faster [9].

In this paper we propose a computationally efficient algorithm that enables decremental support for Incremental PSVMs using a weight decay coefficient. The suggested approach is compared the current time-window based approach proposed by Fung and Mangasarian [10].

2 Background Theory

The basic idea of Support Vector Machine classification is to find an optimal maximal margin separating hyperplane between two classes. Support Vector Machines uses an implicit nonlinear mapping from input-space to a higher dimensional feature-space using kernel-functions, in order to find a hyperplane of problems which are not linear separable in input-space [7, 18]. Classifying multiple classes is commonly performed by combining several binary SVM classifiers in a tournament manner, either one-against-all or one-against-one, the latter approach requiring substantial more computational effort [11].

The standard binary SVM classification problem with soft margin (allowing some errors) is shown visually in Fig. 1(a). Intuitively, the problem is to maximize the margin between the solid planes and at the same time permit as few errors as possible, errors being positive class points on the negative side (of the solid line) or vice versa.



(a) Standard SVM classifier

(b) Proximal SVM classifier

Fig. 1. SVM and PSVM

The standard SVM problem can be stated as a quadratic optimization problem with constraints, as shown in (1).

$$\begin{aligned} \min_{(w,\gamma,y)\in\mathbb{R}^{n+1+m}} \quad & \{ve'y + \tfrac{1}{2}w'w\} \\ \text{s.t. } & D(Aw - e\gamma) + y \geq e \\ & y \geq 0 \end{aligned} \tag{1}$$

$$A \in \mathbb{R}^{m\times n},\ D \in \{-1,+1\}^{m\times 1},\ e = 1^{m\times 1}$$

Fung and Mangasarian [8] replaced the inequality constraint in (1) with an equality constraint. This changed the binary classification problem, because the points in Fig. 1(b) are no longer bounded by the planes, but are clustered around

them. By solving the equation for y and inserting the result into the expression to be minimized, one gets the following unconstrained optimization problem:

$$\min_{(w,\gamma)\in\mathbb{R}^{n+1+m}} f(w,\gamma) = \frac{\nu}{2}\|D(Aw - e\gamma) - e\|^2 + \frac{1}{2}(w'w + \gamma^2) \tag{2}$$

Setting $\nabla f = \left(\frac{\partial f}{\partial w}, \frac{\partial f}{\partial \gamma}\right) = \mathbf{0}$ one gets:

$$\underbrace{\begin{pmatrix} w \\ \gamma \end{pmatrix}}_{\mathcal{X}} = \begin{pmatrix} A'A + \frac{I}{\nu} & -A'e \\ -e'A & \frac{1}{\nu} + m \end{pmatrix}^{-1} \begin{pmatrix} A'De \\ -e'De \end{pmatrix} = \underbrace{\left(\frac{I}{\nu} + E'E\right)^{-1}}_{\mathcal{A}^{-1}} \underbrace{E'De}_{\mathcal{B}} \tag{3}$$

$$E = [A - e],\ E \in \mathbb{R}^{m\times(n+1)}$$

Agarwal has showed that the Proximal SVM is directly transferable to a ridge regression expression [1]. Fung and Mangasarian [10] later showed that (3) can be rewritten to handle increments (E^i, d^i) and decrements (E^d, d^d), as shown in (4). This decremental approach is based on time windows.

$$\begin{aligned} \mathcal{X} &= \begin{pmatrix} w \\ \gamma \end{pmatrix} \\ &= \left(\frac{I}{\nu} + E'E + (E^i)'E^i - (E^d)'E^d\right)^{-1} \left(E'd + (E^i)'d^i - (E^d)'d^d\right) \ , \end{aligned} \tag{4}$$

where $d = De$

.

3 PSVM Decremental Learning using Weight Decay Coefficient

The basic idea is to reduce the effect of the existing (old) accumulated training knowledge $E'E$ with an exponential weight decay coefficient α.

$$\begin{pmatrix} w \\ \gamma \end{pmatrix} = \left(\frac{I}{\nu} + \alpha \cdot E^{'}E + E^{i^{'}}E^i\right)^{-1} \left(\alpha \cdot E^{'}d + E^{i^{'}}d^i\right) \ ;\ \alpha \in \langle 0, 1] \tag{5}$$

As opposed to the decremental approach in expression (4), the presented weight decay approach *does not require storage of increments* $(E^{i^{'}}E^i\,,\, E^{i^{'}}d^i)$ later to be retrieved as decrements $(E^{d^{'}}E^d\,,\, E^{d^{'}}d^d)$.

A hybrid approach is shown in expression (6), where one has both a soft decremental effect using the weight decay coefficient α as well as a hard decremental effect using a fixed window of size W increments.

$$\begin{pmatrix} w \\ \gamma \end{pmatrix} = \left(\frac{I}{\nu} + \alpha \cdot E^{'} E + E^{i^{'}} E^{i} - \alpha^{W} \cdot E^{d^{'}} E^{d} \right)^{-1} \cdot \left(\alpha \cdot E^{'} D + E^{i^{'}} D^{i} - \alpha^{W} \cdot E^{d^{'}} D^{d} \right) ; \alpha \in \langle 0, 1]$$ (6)

4 Related Work

Syed et al. presented an approach for handling concept drift with SVM [2]. Their approach trains on data, and keeps only the support vectors representing the data before (exact) training with new data *and* the previous support vectors. Klinkenberg and Joachims presented a window adjustment based SVM method for *detecting* and handling concept drift [13]. Cauwenberghs and Poggio proposed an incremental and decremental SVM method based on a different approximation than used by us [6].

5 Empirical results

In order to test and compare our suggested decremental PSVM learning approach with the existing window-based approach we created synthetic binary classification data sets with simulated concept drift. This was created by sampling feature values from a multivariate normal distribution where the covariance matrix $\Omega = I$ (identity matrix) and the mean vector μ was sliding linearly from only $+1$ values to -1 values for the positive class case, and vice versa for the negative class [14], as shown in algorithm 1.

Algorithm 1 *simConceptDrift*$(nFeat, nSteps, nExPerStep, start)$

Require: $nFeat, nSteps, nExPerStep \in \mathbb{N}$ and $start \in \mathbb{R}$
Ensure: Linear stochastic drift in $nSteps$ from $start$ to $-start$
1: $center = [start, \ldots, start]$ {vector of length $nFeat$}
2: $origcenter = center$
3: **for all** *step* in $\{0, \ldots, nSteps - 1\}$ **do**
4: **for all** *synthExampleCount* in $\{0, \ldots, nExPerStep - 1\}$ **do**
5: sample example from multivar.gauss.dist with $\mu = center$ and σ^2's $= 1$
6: **end for**
7: $center = origcenter \cdot (1 - 2 \cdot \frac{step+1}{nStep-1})$ {concept drift}
8: **end for**

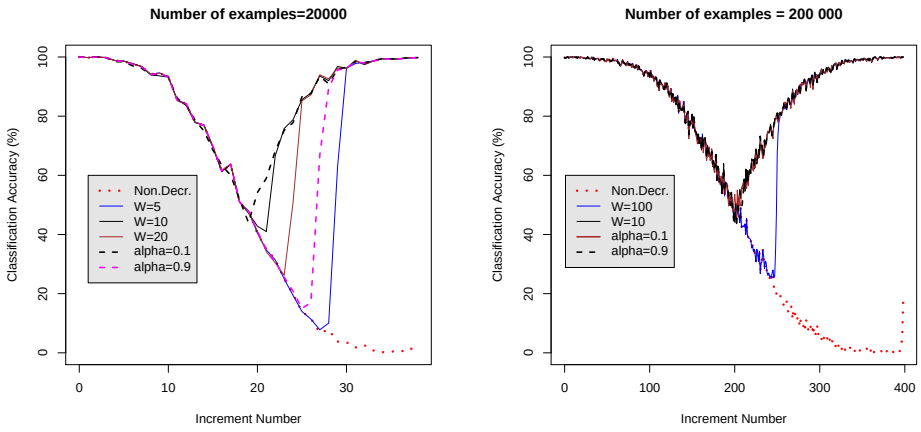
5.1 Classification Accuracy

For the small concept drift test (20000 examples with 10 features and 40 increments of 500 examples, figure 2(a)), the weight decay of $\alpha = 0.1$ performs slightly better in terms of unlearning than a window size of $W = 5$, and a weight

decay of $\alpha = 0.9$ performs between unlearning with $W = 10$ and $W = 20$, and the unlearning performance varies quite a bit with α.

For the medium concept drift test (200000 examples with 10 features and 400 increments of 500 examples, figure 2(b)), the value of α matters less, this due to more increments shown and faster exponential effect of the weight decay coefficient than in the small concept drift test.

As seen in both figure 2(a) and 2(b), there is "dip" in classification performance around their respective center points (increment number ≈ 20 and 200). This is caused by concept drift, i.e. the features of the positive and negative class are indiscernible.



(a) Short timespan

(b) Medium timespan

Fig. 2. Classification Accuracy under Concept Drift

5.2 Computational Performance

As shown in figure 5.2 the computational performance (measured in wallclock time) of the weight decay based approach is almost twice as fast as the window-based approach except for large windows (e.g. $W = 1000$). The performance difference seems to decrease with increasing increment size, this is supported by the P-values from T-test comparisons. 21 out of 27 T-tests (tables 1-3) showed significant difference in favor of the weight decay based approach over the window based approach. Performed T-tests were based on timing of ten repeated runs of each presented configuration of α, w and increment size.

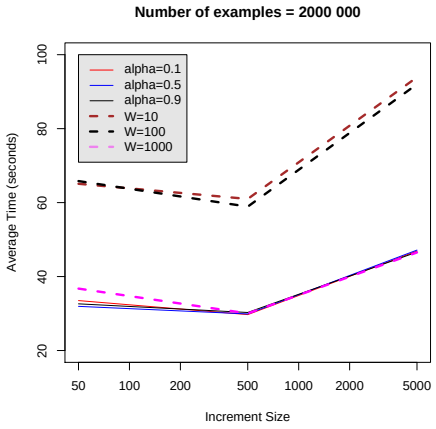


Fig. 3. Computational Performance (Long timespan)

	w=10	w=100	w=1000
α=0.1	0.00	0.00	0.01
α=0.5	0.00	0.00	0.00
α=0.9	0.00	0.00	0.00

Table 1. P-values for increment size 50 (Comp. Perf.)

5.3 Implementation and Test environment

The incremental and decremental proximal SVM has been implemented in C++ using the CLapack and ATLAS libraries [3, 19]. Support for Python and Java interfaces to the library is currently under development using the "Simplified Wrapper and Interface Generator"[15]. A Linux cluster (Athlon 1.4-1.66 GHz nodes, Sorceror Linux) has served as the test environment.

Acknowledgements

We would like to thank Professor Mihhail Matskin and Professor Arne Halaas. This work is supported by the Norwegian Research Council.

6 Conclusion and Future Work

We have introduced a weigth decay based decremental approach for proximal SVMs and shown that it can replace the current window-based approach. The

	w=10	w=100	w=1000
α=0.1	0.00	0.00	0.25
α=0.5	0.00	0.00	0.39
α=0.9	0.00	0.00	0.67

Table 2. P-values for increment size 500 (Comp. Perf.)

	w=10	w=100	w=1000
α=0.1	0.00	0.00	0.67
α=0.5	0.00	0.00	0.79
α=0.9	0.00	0.00	0.57

Table 3. P-values for increment size 5000 (Comp. Perf.)

weight decay based approach is significantly faster than the window-based approach (due to less IO-requirements) for small-to-medium increment and window sizes, this is supported by simulation and p-values from T-Test.

Future work includes applying the approach on demanding incremental classification and prediction problems. e.g. game usage mining [17]. Algorithmic improvements that needs to be done include 1) develop incremental multiclass balancing mechanisms, 2) investigate the approriateness of parallellized incremental proximal SVMs, 3) strengthen implementation with support for tuning set and kernels.

References

1. Deepak K. Agarwal. Shrinkage Estimator Generalizations of Proximal Support Vector Machines. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 173–182. ACM Press, 2002.
2. Nadeem Ahmed, Huan Liu, and Kah Kay Sung. Handling Concept Drifts in Incremental Learning with Support Vector Machines. In *Proceedings of the fifth International Conference on Knowledge Discovery and Data Mining*, pages 317–321. ACM Press, 1999.
3. E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999.
4. Asa Ben-Hur, David Horn, Hava T. Siegelmann, and Vladimir Vapnik. Support Vector Clustering. *Journal of Machine Learning Research*, 2:125–137, 2001.
5. Robert Burbidge and Bernhard F. Buxton. An introduction to support vector machines for data mining. In M. Sheppee, editor, *Keynote Papers, Young OR12*, pages 3–15, University of Nottingham, March 2001. Operational Research Society, Operational Research Society.

6. Gert Cauwenberghs and Tomaso Poggio. Incremental and Decremental Support Vector Machine Learning. In *Advances in Neural Information Processing Systems (NIPS'2000)*, volume 13, pages 409–415. MIT Press, 2001.
7. Nello Christiani and John Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods*, chapter 6, pages 93–111. Cambridge University Press, 1st edition, 2000.
8. Glenn Fung and Olvi L. Mangasarian. Multicategory Proximal Support Vector Classifiers. *Submitted to Machine Learning Journal*, 2001.
9. Glenn Fung and Olvi L. Mangasarian. Proximal support vector machine classifiers. In *Proceedings of the 7th ACM Conference on Knowledge Discovery and Data Mining*, pages 77–86. ACM, 2001.
10. Glenn Fung and Olvi L. Mangasarian. Incremental Support Vector Machine Classification. In R. Grossman, H. Mannila, and R. Motwani, editors, *Proceedings of the Second SIAM International Conference on Data Mining*, pages 247–260. SIAM, April 2002.
11. Chih-Wei Hsu and Chih-Jen Lin. A Comparison of Methods for Multi-class Support Vector Machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, 2002.
12. Jeffrey Huang, Xuhui Shao, and Harry Wechsler. Face pose discrimination using support vector machines (svm). In *Proceedings of 14th Int'l Conf. on Pattern Recognition (ICPR'98)*, pages 154–156. IEEE, 1998.
13. Ralf Klinkenberg and Thorsten Joachims. Detecting Concept Drift with Support Vector Machines. In Pat Langley, editor, *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*. Morgan Kaufmann, 2000.
14. Kenneth Lange. *Numerical Analysis for Statisticians*, chapter 7.3, pages 80–81. Springer-Verlag, 1999.
15. Simplified wrapper and interface generator. Online, http://www.swig.org/, March 2003.
16. Amund Tveit and Magnus Lie Hetland. Multicategory Incremental Proximal Support Vector Classifiers. In *Proceedings of the 7th International Conference on Knowledge-Based Information & Engineering Systems (forthcoming)*, Lecture Notes in Artificial Intelligence. Springer-Verlag, 2003.
17. Amund Tveit and Gisle B. Tveit. Game Usage Mining: Information Gathering for Knowledge Discovery in Massive Multiplayer Games. In *Proceedings of the International Conference on Internet Computing (IC'2002), session on Web Mining*. CSREA Press, June 2002.
18. Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*, chapter 5, pages 138–146. Springer-Verlag, 2nd edition, 1999.
19. Richard C. Whaley, Antoine Petitet, and Jack J. Dongarra. Automated Empirical Optimization of Software and the ATLAS Project". *Parallel Computing*, 27(1-2):3–25, 2001.