Optimization Methods and Software, , 1–14 (2001)

Semi-Supervised Support Vector Machines for Unlabeled Data Classification

GLENN FUNG AND O. L. MANGASARIAN gfung@cs.wisc.edu, olvi@cs.wisc.edu
Computer Sciences Department
Uinversity of Wisconsin
Madison, WI 53706

Received May 22, 2000; Revised November 29, 2000

Editor: Masao Fukushima

Abstract. A concave minimization approach is proposed for classifying unlabeled data based on the following ideas: (i) A small representative percentage (5% to 10%) of the unlabeled data is chosen by a clustering algorithm and given to an expert or oracle to label. (ii) A linear support vector machine is trained using the small labeled sample while simultaneously assigning the remaining bulk of the unlabeled dataset to one of two classes so as to maximize the margin (distance) between the two bounding planes that determine the separating plane midway between them. This latter problem is formulated as a concave minimization problem on a polyhedral set for which a stationary point is quickly obtained by solving a few (5 to 7) linear programs. Such stationary points turn out to be very effective as evidenced by our computational results which show that clustered concave minimization yields: (a) Test set improvement as high as 20.4% over a linear support vector machine trained on a correspondingly small but randomly chosen subset that is labeled by an expert. (b) Test set correctness averaged to within 5.1% when compared to that of a completely supervised linear support vector machine trained on the *entire* dataset which has been labeled by an expert.

Keywords: unlabeled data, classification, support vector machines

1. INTRODUCTION

Linear support vector machines (SVMs) [14, 5, 2] classify two-class labeled datasets by constructing two parallel bounding planes, with maximum distance (margin) apart and such that each plane bounds one class of the labeled points. The distance between these planes, which is a measure of the generalization capability of the SVM, is split midway by a plane parallel to the bounding planes and is used as the classifying plane.

In semi-supervised learning, where only part of the two-class data is labeled, the same procedure is utilized except that the algorithm assigns the unlabeled data to one of two classes in such a way as to achieve separation by two bounding planes and maximizing the margin between the planes.

Bennett and Demiriz [1], who treat datasets which are already partially labeled, formulate the semi-supervised support vector machine (S^3VM) as a mixed integer program (MIP). Their formulation requires the introduction of a binary variable for each unlabeled data point in the training set. This makes the problem difficult to solve for large unlabeled data. State-of-the-art software does not handle

easily problems with much more than 50 unlabeled data points. To overcome this difficulty we propose here a formulation that can handle large unlabeled datasets (with a thousand points) and solve the semi-supervised problem in a considerably shorter time. Our new approach consists of formulating the problem as a concave minimization problem which is solved by a successive linear approximation algorithm [7]. Such an approach has been successfully used on a number of machine learning, data mining and other problems [7, 8, 3, 2]. We term our approach a concave semi-supervised support vector machine (VS^{3}VM).

For classifying unlabeled data, which is our principal aim here, we will make use of the k-median clustering algorithm [3] in combination with the proposed VS^{3}VM as follows. The k-median algorithm is used to select a small representative percentage, 5% to 10%, to be labeled by an expert or an oracle in order to be used as labeled data, together with the remaining part of the data, that remains unlabeled, in VS^{3}VM. Such an approach which can accommodate large datasets produces an improvement as high as 20.4% over a randomly chosen set labeled by an expert and used as a training set in a linear support vector machine. In addition, even if the **entire** dataset is labeled by an expert and classified by a linear support vector machine, our clustering concave minimization approach, using only 5% to 10% of the data as labeled data, can come within an average of 5.1%, in test set correctness, to an SVM trained on the entire dataset labeled by an expert.

When a clustering procedure is combined with VS^{3}VM, as described above, we term the resulting algorithm as clustered VS^{3}VM (CVS^{3}VM).

We briefly outline the contents of the paper now. In Section 2 we formulate the semi-supervised support vector machine S^{3}VM for classifying the elements of a partially labeled two-class dataset as a concave minimization problem (2.1) and prescribe a finite successive linear approximation VS^{3}VM, Algorithm 1, for solving it. In Section 3 we state the k-median clustering Algorithm 1 and demonstrate its use in conjunction with the VS^{3}VM Algorithm on a small two dimensional set. Figures 1 to 3 below, show the superiority of CVS^{3}VM on this example over both clustered and random choices for a training set used in conjunction with a plain linear support vector machine. Section 4 contains our numerical tests on five publicly available datasets which show the following: (a) CVS^{3}VM, our clustered semi-supervised approach, gave the best test set correctness when compared to both a random and a clustered choice of the data used as a training set in a linear support vector machine. Also, CVS^{3}VM was much faster than a mixed integer programming (MIP) formulation S^{3}VM. (b) An improvement as high as 20.4% of test set correctness of CVS^{3}VM over a random choice for a training set used in a linear support vector machine. (c) Test set correctness of CVS^{3}VM averaged to within 5.1% when compared to a completely supervised linear SVM for which the entire dataset has been labeled.

A word about our notation. All vectors will be column vectors unless transposed to a row vector by a prime superscript $'$. The scalar (inner) product of two vectors x and y in the n-dimensional real space R^n will be denoted by $x'y$. For an $\ell \times d$ matrix A, A_i will denote the ith row of A. The identity matrix in a real space of arbitrary dimension will be denoted by I, while a column vector of ones of arbitrary

dimension will be denoted by e. The component-by-component minimum of two p-dimensional vectors r and s is denoted by $\min\{r, s\}$, with component j being: $\min\{r_j, s_j\}$, $j = 1, \ldots, p$.

2. CONCAVE SEMI-SUPERVISED SVM (VS^{3}VM)

We consider here the dataset consisting of m labeled points and p unlabeled points all in R^n. The m labeled points are represented by the matrix $A \in R^{m\times n}$ and p unlabeled points in R^n represented by the matrix $B \in R^{p\times n}$. The labels for A are given by an $m \times m$ diagonal matrix D of ± 1. Bennett and Demiriz [1] formulate the semi-supervised linear support vector machine for generating the separating plane $x'w = \gamma$ for this problem as follows:

$$\begin{array}{rcl} & \min\limits_{w,\gamma,y,z,r,s} \nu e^{'} y + e^{'} z + \mu e^{'} \min\{r, s\} & \\ \text{s.t.} & D(Aw - e\gamma) + y & \geq e \\ -z \leq & w & \leq z \\ & Bw - e\gamma + r & \geq e \\ & -Bw + e\gamma + s & \geq e \\ & y \geq 0, r \geq 0, s \geq 0, & \end{array} \tag{2.1}$$

where ν and μ are positive parameters. The first two terms of the objective function together with the first two constrains and $y \geq 0$, correspond to a linear SVM (3.6, below) [2, Equation (13)] which attempts to classify the labeled part of the dataset represented by the matrix A. The last term in the objective function together with the remaining constraints assign each row of the matrix B, representing unlabeled data, to class $+1$ or -1, whichever generates a lower misclassification error: $\min\{r, s\}$. The parameters μ, ν are positive numbers that weight the different terms of the objective function and are chosen as described in Section 4. Bennett and Demiriz [1] formulate this problem as a mixed integer program (MIP) by assigning a binary decision variable to each row of the unlabeled matrix B. However only relatively small unlabeled datasets (e.g. 50 points [1]) can be handled by this MIP formulation which sometimes fails due to excessive branching [1]. However, if some local search procedure is combined with the MIP formulation together with the clustering techniques proposed in this paper, the MIP approach can conceivably be considerably improved.

We propose here instead a concave minimization procedure, consisting of solving a finite number (typically 5 to 7) of linear programs, which terminate at a point satisfying a necessary optimality for problem (2.1). The approach is based on the finite successive linear approximation algorithm for minimizing a concave function on a polyhedral set [8, Algorithm 1] and is justified by the fact that nonlinear term $\min\{r, s\}$ in the objective function of (2.1) is concave because it is the minimum of two linear functions. The algorithm consists of linearizing the nonlinear term $\min\{r, s\}$ around the current iterate (r^i, s^i) by taking a supporting plane (generalization of a tangent plane for non-differentiable concave functions) approximation of the function at that point and solving the resulting linear program. This leads to

the following finitely terminating successive linear approximation algorithm based on [8, Algorithm 1].

Algorithm 1 VS^{3}VM *Successive Linear Approximation for S^3VM (2.1) Choose positive values for the parameters μ, ν. Start with a random $(r^0, s^0) \geq 0$. Having (r^i, s^i) determine $(w^{i+1}, \gamma^{i+1}, y^{i+1}, z^{i+1}, r^{i+1}, s^{i+1})$ by solving the linear program:*

$$\min_{w,\gamma,y,z,r,s} \nu e^{'} y + e^{'} z + \mu \partial (e^{'} \min \{r^i, s^i\}) \begin{bmatrix} r - r^i \\ s - s^i \end{bmatrix}$$

$$\begin{array}{rcl} s.t. \quad D(Aw - e\gamma) + y & \geq & e \\ -z \leq \quad w & \leq & z \\ Bw - e\gamma + r & \geq & e \\ -Bw + e\gamma + s & \geq & e \\ y \geq 0, r \geq 0, s \geq 0. & & \end{array} \tag{2.2}$$

where the supergradient $\partial(e^{'} \min \{r^i, s^i\})$ of $e' \min \{r, s\}$ is defined below in (2.4). Stop when the following necessary optimality condition holds:

$$\begin{array}{l} \nu e^{'} (y^{i+1} - y^i) + e^{'} (z^{i+1} - z^i) \\ + \mu \partial (e^{'} \min \{r^i, s^i\})' \begin{bmatrix} r^{i+1} - r^i \\ s^{i+1} - s^i \end{bmatrix} = 0. \end{array} \tag{2.3}$$

For a concave function $f : R^n \longrightarrow R$ the supergradient $\partial(f(x))$ of f at x is a vector in R^n satisfying:

$$f(y) - f(x) \leq \partial f(x)(y - x),$$

for all $y \in R^n$. The supergradient reduces to the ordinary gradient $\nabla f(x)$, when f is differentiable at x [12, 13]. The set of all supergradients at a point x is called the superdifferential.

In our case $e^{'} \min \{r, s\} : R^{2p} \longrightarrow R$ is a non-differentiable concave function and its supergradient is given by:

$$\partial(e^{'} \min\{r,s\}) =$$

$$\sum_{j=1}^{p} \begin{cases} \begin{pmatrix} I_j \\ 0_p \end{pmatrix} & \text{if} \quad r_j < s_j \\ (1-\lambda)\begin{pmatrix} I_j \\ 0_p \end{pmatrix} + \lambda \begin{pmatrix} 0_p \\ I_j \end{pmatrix} & \text{if} \quad r_j = s_j \\ \begin{pmatrix} 0_p \\ I_j \end{pmatrix} & \text{if} \quad r_j > s_j \end{cases} \tag{2.4}$$

Here $0_p \in R^p$ is a column vector of zeros, $I_j \in R^p$ is the jth column of the identity matrix I, and $\lambda \in (0,1)$. In all our computations we set $\lambda = 0.5$.

By [8, Theorem 3] Algorithm 1 terminates after a finite number of linear programs at a point satisfying the necessary optimality condition (2.3) for problem (2.1).

Our numerical experiments showed that instead of a random starting point $(r^0, s^0) \geq 0$, a much better starting point for the Algorithm 2.1 can be obtained by solving the following linear program:

$$\begin{array}{rcl} \min\limits_{w,\gamma,y,z,r,s} \nu e'y + e'z + \dfrac{\mu}{2}(e'(r+s)) & & \\ \text{s.t.} \qquad D(Aw - e\gamma) + y & \geq & e \\ -z \leq \qquad w & \leq & z \\ Bw - e\gamma + r & \geq & e \\ -Bw + e\gamma + s & \geq & e \\ y \geq 0, r \geq 0, s \geq 0, & & \end{array} \tag{2.5}$$

which corresponds to the linear program (2.2) with a supergradient of $e' \min\{r,s\}$ evaluated at $r = s$.

We turn our attention now to a combination of the S^{3}VM with the k-median clustering algorithm that will enable us to handle unlabeled data.

3. CLUSTERING + VS^{3}VM (CVS^{3}VM) FOR UNLABELED DATA

To handle unlabeled data we make use of the k-median clustering algorithm [3], which de-emphasizes outliers, in order to form a small training set (5% to 10% of the data) by choosing among the unlabeled data a "representative" subset to be labeled by an expert. This also constitutes a means for handling large unlabeled datasets such as those that occur in data mining in which case relatively few points can be labeled through expensive or time consuming services of an expert. The clustering approach can also be used as part of an incremental algorithm where only a small percentage of incoming data is chosen by the k-median algorithm to be labeled.

Our approach here will consist of the following: For a given percentage of the data, select a "good" subset to label and give the resulting labeled-unlabeled dataset to the VS^{3}VM Algorithm 1. We describe now the "selection" procedure of the above

approach, which will be carried out by using the k-median clustering algorithm [3] described in the section below.

3.1. THE K-MEDIAN CLUSTERING ALGORITHM

Consider a set of t data points in R^n represented by a general matrix $H \in R^{t\times n}$. We first find k cluster centers for the data such that the sum of distances between each point and the closest cluster center $C_l, l = 1, \ldots, k$ is minimized. The idea then is to treat points within a certain distance from these k cluster centers as representative points of that cluster, and hence of the overall dataset, and have them labeled by an expert. These points generate the matrix A of the semi-supervised Algorithm 1 S^{3}VM. The rest of the points remain unlabeled for use in S^{3}VM as the matrix B. In order to achieve this we use the simple and finite k-median clustering algorithm of [3] given below. When the k-median clustering algorithm is applied as described to select labeled data for the VS^{3}VM Algorithm 1, the algorithm is referred to as the **CVS^{3}VM Algorithm**.

Algorithm 1 k-Median Algorithm *Given $C_1^j, \ldots, C_k^j$ at iteration j, compute $C_1^{j+1}, \ldots, C_k^{j+1}$ by the following two steps:*

(a) Cluster Assignment: *For each H_i', $i = 1, \ldots t$, determine $\ell(i)$ such that $C_{\ell(i)}^j$ is closest to H_i' in the one norm.*

(b) Cluster Center Update: *For $\ell = 1, \ldots, k$ choose C_ℓ^{j+1} as a median of all H_i' assigned to C_ℓ^j.*

Stop when $C_\ell^{j+1} = C_\ell^j$.

The choice of k in the k-median algorithm above depends on the size of the original dataset and is typically chosen so that a certain desired total percentage, say 5% to 10%, of the dataset falls within a desired distance from a closest cluster center.

To show that the clustered choice of data labeled by an expert in combination with a semi-supervised SVM is the most effective way for handling unlabeled data, we compared CVS^{3}VM with other plausible approaches as follows.

1. **Total Set + SVM: Total Set labeled by expert + Linear SVM**
   We solve here the linear SVM:

$$\begin{array}{rc} & \min\limits_{w,\gamma,y,z} \nu e'y + e'z \\ \text{s.t.} & D(Aw - e\gamma) + y \geq e \\ -z \leq & w \leq z \\ & y \geq 0. \end{array} \tag{3.6}$$

   Thus training is done here on a completely labeled data set which is equivalent to (2.1) with an empty B. In contrast, although CS^{3}VM is trained on just a

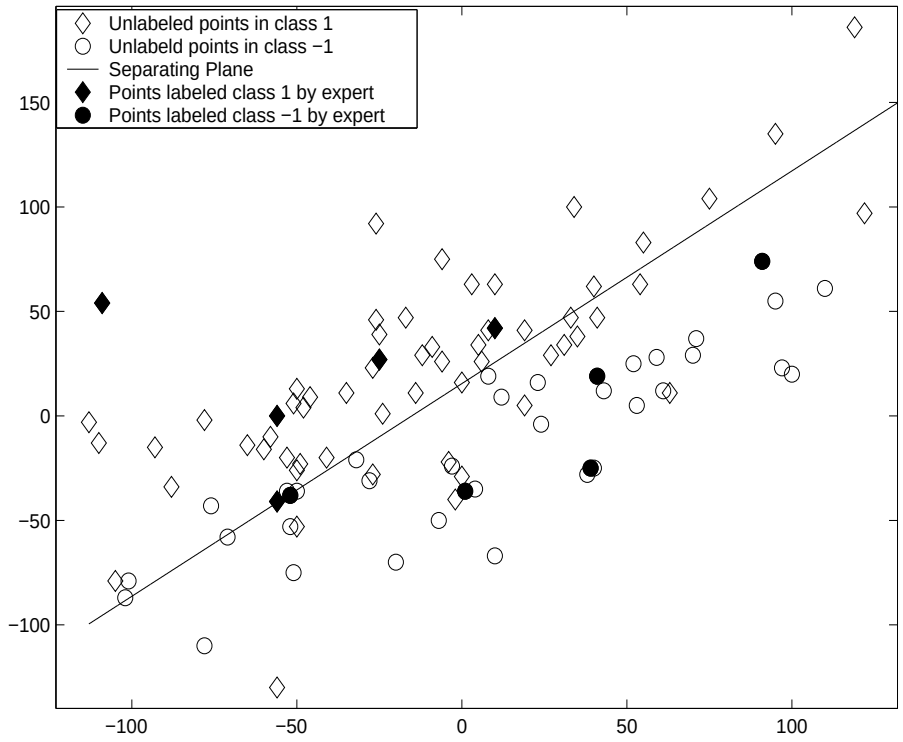

Figure 1. **CVS^{3}VM** for Unlabeled Data: Example showing 10% of a dataset, as solid points, whose labels, diamonds and circles, are unknown to the k-median algorithm which selects them to be labeled by an expert and then are used as labeled data in VS^{3}VM Algorithm 1. The remaining 90% points are used as unlabeled data by VS^{3}VM. **Resulting separating plane correctly classifies 81% of the data**

5% to 10% of the data that is labeled by an expert, its test correctness is close to that obtained by a linear SVM using **all** the dataset labeled by an expert as shown by our numerical tests.

2. **Random + SVM: Random choice for labeling by expert + Linear SVM**

   Here, 5%-10% of the data to be labeled is chosen randomly and used as training data in the linear SVM algorithm. The information on the remaining unlabeled data is not considered since we are applying a supervised learning approach to the labeled data only. Because of the random choice of the training data, we performed this experiment 10 times in order to obtain a more consistent result. As was expected, this approach gave the worst performance.

3. **Clustering + SVM: Clustering choice of data + Linear SVM**

   In this case no information on the unlabeled data is used. However, since the k-median clustering algorithm is used to choose the data to be labeled, a

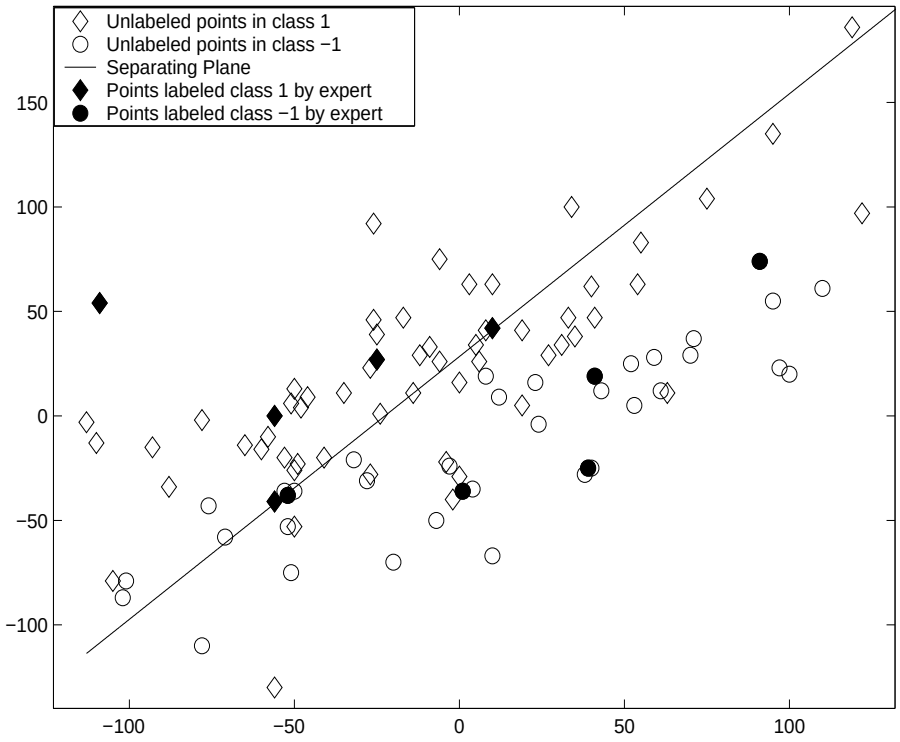

Figure 2. **Cluster + SVM** for Unlabeled Data: Example showing 10% of a dataset, as solid points, whose labels, diamonds and circles, are unknown to the k-median algorithm which selects them to be labeled by an expert and then are used as labeled data in a linear SVM (3.6). **The resulting separating plane correctly classifies 72% of the data.**

"smarter" choice of the data to be labeled is made. An improvement on test set correctness over Random + SVM is obtained.

4. **CVS^{3}VM: Clustered choice of the data + VS^{3}VM**

   This is the principal proposed algorithm of this paper. Pick a small percentage of the unlabeled data by clustering to be labeled, then use this labeled data with the the remaining unlabeled data in the Concave Semi-supervised SVM (VS^{3}VM).

5. **Cluster + S^{3}VM: Clustering choice of data + S^{3}VM (MIP)**

   This case is similar to CVS^{3}VM except that instead of solving a concave minimization problem we solve here a Mixed Integer problem (MIP) as proposed in [1].

We now illustrate how CVS^{3}VM works on a simple 2-dimensional example of 100 data points consisting of diamond and circular shapes depicted in Figure 1, created

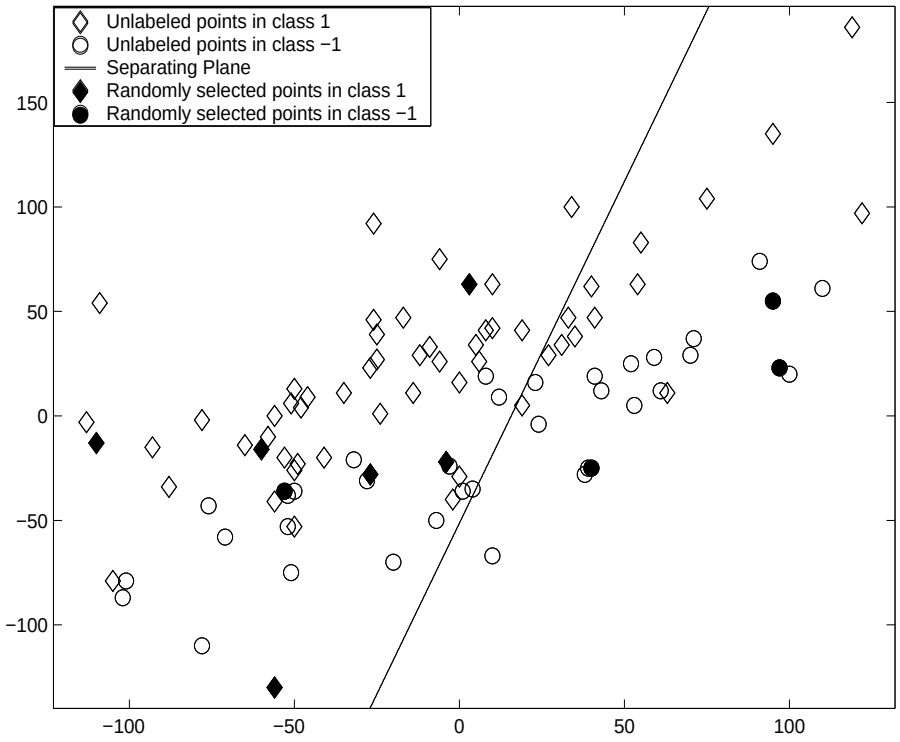

Figure 3. **Random + SVM** for Unlabeled Data: Example showing 10% of a dataset, as solid points, which are randomly selected and labeled by an expert, then used as training set in a linear SVM (3.6). **Resulting separating plane correctly classifies 69% of the data.**

by the NDC (normally distributed clusters) generator [11]. The labels (diamond and circular shapes) are made known to VS^{3}VM as follows. In Figure 1 the solid shapes are the 10% of the original unlabeled dataset that are selected by the k-median algorithm to be labeled data and given to VS^{3}VM while the remaining 90% of the original data shown as hollow shapes are treated as unlabeled data by VS^{3}VM. The resulting separating plane shown in Figure 1 correctly classifies 81% of the data. If we now drop the unlabeled data from the training part of the problem and revert to a linear SVM (3.6) trained on data chosen by a k-median algorithm, we obtain the separating plane shown in Figure 2 which correctly classifies a lower percentage of the data: 72%. Finally, if we use the SVM (3.6) on a randomly chosen set of points that are labeled and depicted as solid points, we obtain the separating plane shown in Figure 3 with a still lower correctness of 69%.

4. NUMERICAL TESTING

Our numerical testing was carried out on five publicly available labeled datasets. Unlabeled data was simulated by dropping labels from some or all the points in a given dataset. Four of the datasets are from the UCI Machine Learning Repository [10], and one of them was created by the NDC (normally distributed clusters) generator [11]. Table 1 shows the number of points of each dataset and the dimensionality of the space they are in. All the matrix manipulations were carried out using MATLAB [9]. Linear programs were solved by calling the state-of-the-art CPLEX solver [6] from MATLAB and GAMS [4]. All experiments were run on Locop1, one of the machines of the University of Wisconsin Computer Sciences Department Data Mining Institute. Locop1 is a Dell PowerEdge 6300 server powered with four 400 MHz Pentium II Xeon processors, four gigabytes of memory, 36 gigabytes of disk space, and the Windows NT Server 4.0 operating system.

Test 1 *The following five experiments were performed:*

(i) **Total Set Linear SVM**

Linear SVM (3.6) trained on a completely labeled dataset.

(ii) **Random + Linear SVM** *A linear SVM for which a random 5% to 10% subset of the data is selected as the training set to be labeled by an expert.*

(iii) **Cluster + Linear SVM** *A linear SVM is used together with the k-median Clustering Algorithm 1 for selection of a 5%-10% subset of data points to be labeled and used as a training set.*

(vi) **CVS^{3}VM** *Algorithm VS^3 VM 1 with k-median Clustering Algorithm 1 for selection of a 5%-10% subset of data points to be labeled by an expert with the rest of the data remaining unlabeled in Algorithm VS^3 VM 1.*

(v) **Cluster + S^{3}VM (MIP)** *Use Algorithm S^3 VM [1] (MIP formulation) with k-median Clustering Algorithm 1 for selection of a 5%-10% subset of data points to be labeled by an expert with the rest of the data remaining unlabeled.*

When the k-median algorithm was used in order to choose the 5% (10%) subset of the data, the value for k was approximately set to 5% (10%) of the total number of points in the whole dataset. For example for the Ionosphere dataset of 351 points, we chose 10% of the data to be labeled, thus $k = 35$. The labeled training set was chosen as the set of points within a certain distance from the cluster centers of the k-median algorithm so as to total to 10% of the original unlabeled dataset. For CVS^{3}VM, we performed a variation of the standard *tenfold cross-validation.* Once we obtained our 10% labeled data to be used as training set, we divided the remaining 90% of data points into 10 folds, so that each fold contained 9% of the original dataset. We then used nine of these 10 folds (81 % of the original points) as unlabeled training data and the remaining 9% as a testing set. We repeated this procedure ten times choosing a different fold for testing and the remaining folds as

a unlabeled training data every time. The 10% of the labeled data was fixed for all the ten problems corresponding to each fold. See Figure 4 for a graphical depiction of this procedure. When comparison with a random choice of labeled data was made, we repeated the latter process ten times and reported the average.

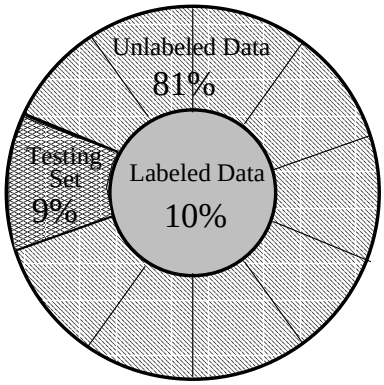

Figure 4. **CVS^{3}VM tenfold Cross Validation**

The parameter ν appearing in the formulation of the S^{3}VM problem (2.1) is determined by a formula proposed in [1]: $\nu = \frac{10^3}{m+p}$, where m is the number of the labeled points, and p the number of the remaining unlabeled points. The parameter μ in (2.1) was adjusted by tuning. When the SVM was used on the complete dataset as labeled data, the parameter ν was also adjusted by tuning.

Table 1 gives tenfold testing set correctness for 5 totally unlabeled datasets for the five algorithms described in Test 1: Total Set + Linear SVM, Random + Linear SVM, Cluster + Linear SVM, CVS^{3}VM and Cluster + S^{3}VM (MIP). The improvement and the p-values shown in the table are relative to the Random + Linear SVM approach.

CVS^{3}VM had the highest test set correctness and the relative improvement, over Random + Linear SVM, was as high as as 20.4% with a p-value of 0.02. Also, CVS^{3}M test set correctness using as little as 5% to 10% of the data as labeled data, was on average within 5.1% of that for a linear SVM using **all** the data as a labeled training dataset.

When S^{3}VM-MIP terminated before reaching the 10,000-seconds time limit, it was much slower than CVS^{3}M. For example, in the Heart dataset the total time spent by S^{3}VM-MIP on tenfold cross validation was 462.2 seconds, while the time used by VS^{3}VM was 14.9 seconds in total. The corresponding times for the Housing dataset were 1193.2 seconds for S^{3}VM-MIP and 32.9 for VS^{3}VM.

(i) **Total Set SVM :** Entire dataset labeled by an expert and used by a linear SVM (3.6).

(ii) **R + SVM:** Small randomly chosen subset labeled by an expert and used by a linear SVM (3.6).

(iii) **C + SVM:** Small subset of the data chosen by clustering, labeled by an expert and used by a linear SVM (3.6).

(iv) **CVS^{3}VM:** Small subset of the data chosen by clustering, labeled by an expert and VS^{3}VM (2.1) solved the concave minimization algorithm, Algorithm 1.

(v) **C + S^{3}VM:** Same as 4 except a Mixed Integer Program (MIP) is used instead of the concave minimization algorithm, Algorithm 1.

* The relative improvement and the the p-value are calculated with respect to **Random + SVM**.
† Failure was declared when total time exceeded 10,000 seconds.

Table 1. Tenfold test set correctness of the experiments described in Test 4.1 on 5 public datasets. **Bold figures denote highest test set correctness.**

Data Set points×dim.	Total Set SVM Test	R + SVM Test	C + SVM Test Improvement* p-value*	CVS^{3}VM Test Improvement* p-value*	C + S^{3}VM(MIP) Test Improvement* p-value*
NDC Set 1000×32	72.6%	58.0%	60.0% 3.4% 0.38	**67.0%** 15.5% 0.01	Failed† - -
Cleveland Heart 297×13	83.2%	69.0%	73.3% 6.2% 0.01	**76.0%** 10.1% 0.01	**76.0%** 10.1% 0.01
Housing 506×13	86.2%	70.0%	73.3% 4.7% 0.18	**81.4%** 16.2% 0.05	81.0% 15.7% 0.08
Ionosphere 351×34	87.1%	78.3%	78.5% 0.25% 0.93	**84.2%** 7.5% 0.05	Failed† - -
Sonar 208×60	77.4%	64.0%	74.6% 16.6% 0.04	**77.1%** 20.4% 0.02	Failed† - -

5. CONCLUSION

We have proposed a concave formulation for the semi-supervised support vector machine problem and given a fast finite linear programming based formulation for its solution. Unlike a mixed integer formulation, our concave minimization Algorithm 1 can handle large datasets that are mostly unlabeled. Numerical tests show the potential of the concave semi-supervised support vector machine algorithm as an efficient and viable tool for handling large totally unlabeled datasets. This is carried out by selecting a small portion of the unlabeled dataset by clustering, labeling it by an expert and using the concave minimization algorithm VS^{3}VM. Future directions include application of VS^{3}VM to incremental data mining where a small portion of the dataset is labeled incrementally as new data becomes available, as well as multi-category unlabeled data classification.

Acknowledgements

We are grateful to Kristin Bennett for a number of helpful suggestions. The research described in this Data Mining Institute Report 99-05, October 1999, was supported by National Science Foundation Grants CCR-9729842 and CDA-9623632, by Air Force Office of Scientific Research Grant F49620-97-1-0326 and by the Microsoft Corporation.

References

1. K. P. Bennett and A. Demiriz. Semi-supervised support vector machines (1998). In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems -10-*, pages 368–374, (Cambridge, MA). MIT Press.
2. P. S. Bradley and O. L. Mangasarian (1998). Feature selection via concave minimization and support vector machines. In J. Shavlik, editor, *Machine Learning Proceedings of the Fifteenth International Conference(ICML '98)*, pages 82–90, (San Francisco, California). Morgan Kaufmann. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/98-03.ps.
3. P. S. Bradley, O. L. Mangasarian, and W. N. Street (1996). Clustering via concave minimization. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems -9-*, pages 368–374, (Cambridge, MA) . MIT Press. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/96-03.ps.
4. A. Brooke, D. Kendrick, and A. Meeraus (1998). *GAMS: A User's Guide*. The Scientific Press, (South San Francisco, CA).
5. V. Cherkassky and F. Mulier (1998). *Learning from Data - Concepts, Theory and Methods*. John Wiley & Sons, (New York).
6. CPLEX Optimization Inc., Incline Village, Nevada (1992). *Using the CPLEX(TM) Linear Optimizer and CPLEX(TM) Mixed Integer Optimizer (Version 2.0)*.
7. O. L. Mangasarian. Machine learning via polyhedral concave minimization (1996). In H. Fischer, B. Riedmueller, and S. Schaeffler, editors, *Applied Mathematics and Parallel Computing - Festschrift for Klaus Ritter*, pages 175–188. Physica-Verlag A Springer-Verlag Company, (Heidelberg). ftp://ftp.cs.wisc.edu/math-prog/tech-reports/95-20.ps.
8. O. L. Mangasarian. Solution of general linear complementarity problems via nondifferentiable concave minimization (1997). *Acta Mathematica Vietnamica*, 22(1):199–205. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/96-10.ps.
9. MATLAB (1992) . *User's Guide*. The MathWorks, Inc., Natick, MA 01760.

10. P. M. Murphy and D. W. Aha (1992). UCI repository of machine learning databases. www.ics.uci.edu/∼mlearn/MLRepository.html.
11. D. R. Musicant (1998). NDC: normally distributed clustered datasets. www.cs.wisc.edu/∼musicant/data/ndc/.
12. B. T. Polyak (1987). *Introduction to Optimization*. Optimization Software, Inc., Publications Division, (New York).
13. R. T. Rockafellar. *Convex Analysis*.
14. V. N. Vapnik (1995). *The Nature of Statistical Learning Theory*. Springer, (New York).