SVMC: Single-Class Classification With Support Vector Machines

Hwanjo Yu
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana,IL 61801 USA
hwanjoyu@uiuc.edu

Abstract

Single-Class Classification (SCC) seeks to distinguish one class of data from the universal set of multiple classes. We present a new SCC algorithm that efficiently computes an accurate boundary of the target class from positive and unlabeled data (without labeled negative data).

1 Introduction

Single-Class Classification (SCC) seeks to distinguish one class of data from the universal set of multiple classes, (e.g., distinguishing apples from fruits, identifying "waterfall" pictures from image databases, or classifying personal homepages from the Web) (Throughout the paper, we call the target class *positive* and the complement set of samples *negative.)*

Since it is not natural to collect the "non-interesting" objects (i.e., negative data) to train the concept of the "interesting" objects (i.e., positive data), SCC problems are prevalent in real world where positive and unlabeled data are widely available but negative data are hard or expensive to acquire [Yu *et al.* 2002; Letouzey *et al. ,* 2000; DeComite *et a*l., 1999]. For example, in text or Web page classification (e.g., personal homepage classification), collecting negative training data (e.g., a sample of "non-homepages") is delicate and arduous because manually collected negative data could be easily biased because of a person's unintentional prejudice, which could be detrimental to classification accuracy. In an example of diagnosis of a disease, positive data are easy to access (e.g., all patients who have the disease) and unlabeled data are abundant (e.g., all patients), but negative data are expensive if detection tests for the disease are expensive since all patients in the database cannot be assumed to be negative samples if they have never been tested. Further applications can be also found in pattern recognition, image retrieval, classification for data mining, rare class classification, etc. In this paper, we focus on this SCC problem from positive and unlabeled data (without labeled negative data).

1.1 Previous Approaches for SCC

Traditional (semi-)supervised learning schemes are not suitable for SCC without labeled negative data because: (1) the portions of positive and negative spaces are seriously unbalanced without being known (i.e., *Pr(P) « Pr(P)),* and (2) the absence of negative samples in the labeled data set makes unfair the initial parameters of the model and thus it leads to unfair guesses for the unlabeled data.

Active learning methods also try to minimize the labeling labors to construct an accurate classification function by a different approach that involves an interactive process between the learning system and users [Tong and Koller, 2000].

Valiant in 1984 [Valiant, 1984] pioneered *learning theory from positive examples* based on rule learning. In 1998, F. Denis defined the Probably Approximately Correct (PAC) learning model for positive and unlabeled examples, and showed that *k-DNF* (Disjunctive Normal Form) is learnable from positive and unlabeled examples [Denis, 1998]. After that, some experimental attempts to learn using positive and unlabeled data have been tried using k:-DNF or C4.5 [Letouzey *et al.,* 2000; DeComite *et al,* 1999]. Rule learning methods are simple and efficient for learning nominal features but tricky to use for the problems of continuous features, high dimensions, or sparse instance spaces.

Positive Example-Based Learning (PEBL) framework was proposed for Web page classification [Yu *et ai,* 2002]. Their method is limited to the Web domain with binary features, and its training efficiency is poor because of using SVM iteratively whose training time is already at least quadratic to the size of training data set. This problem becomes critical when the size of unlabeled data set is large.

A probabilistic method for the SCC problem has been recently proposed for the text domain [Liu *et ai,* 2002]. As they specified in the paper, their method - a revision of the EM algorithm - performs badly on "hard" problems due to the fundamental limitations of the generative model assumption, the attribute independence assumption which results in linear separation, and the requirement of good estimation of prior probabilities.

OSVM (One-Class SVM) also distinguishes one class of data from the rest of the feature space given only a positive data set [Tax and Duin, 2001; Manevitz and Yousef, 2001]. Based on a strong mathematical foundation, OSVM draws a nonlinear boundary of the positive data set in the feature space using two parameters - v (to control the noise in the training data) and γ (to control the "smoothness" of the boundary). They have the same advantages as SVM, such as efficient handling of high dimensional spaces and systematic nonlinear classification using advanced kernel functions.

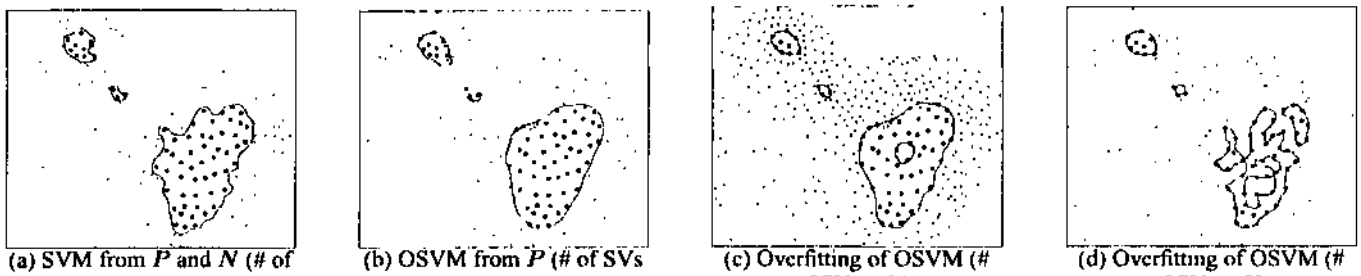


Figure 1: Boundaries of SVM and OSVM on a synthetic data set. *big dots: positive data, small dots: negative data*

However, OSVM requires a much larger amount of positive training data to induce an accurate class boundary because its support vectors (SVs) of the boundary only comes from the positive data set and thus the small number of positive SVs can hardly cover the major directions of the boundary especially in high dimensional spaces. Due to the SVs coming only from positive data, OSVM tends to ovcrfit and undcrfit easily. Tax proposed a sophisticated method which uses artifically generated unlabeled data to optimize the OSVM's parameters that "balance" between ovcrfitting and undcrfitting [Tax and Duin, 2001]. However, their optimization method is infeasibly inefficient in high dimensional spaces, and even with the best parameter setting, its performance still lags far behind the original SVM with negative data due to the shortage of SVs which makes "incomplete" the boundary description. Figure 1(a) and (b) show the boundaries of SVM and OSVM on a synthetic data set in a two-dimensional space. (We used L1BSVM version 2.33' for SVM implementation.) In this low-dimensional space with "enough" data, the obstensibly "smooth" boundary of OSVM is not the result of the good generalization but instead is from the poor expressibility caused by the "incomplete" SVs, which will become much worse in high-dimensional spaces where more SVs around the boundary are needed to cover major directions in the high-dimensional spaces. When we increase the number of SVs in OSVM, it ovcrfits rather than being more accurate as shown in Figure I(c)and(d).

1.2 Contributions and Paper Layout

We first discuss the "optimal" SCC boundary, which motivates our new SCC framework *Mapping-Convergence (MC),* where the algorithms under the MC framework generate the boundary close to the optimum (Section 2). In Section 3, we present an efficient SCC algorithm *Support Vector Mapping Convergence (SVMC)* under the MC framework. We prove that although SVMC iterates under the MC framework for the "near-optimal" result, its training time is independent of the number of iterations, which is asymptotically equal to that of a SVM. We empirically verify our analysis of SVMC by extensive experiments on various domains of real data sets such as text classification (e.g., Web page classification), pattern recognition (e.g., letter recognition), and bioinformatics (e.g., diagnosis of breast cancer), which shows the outstanding performance of SVMC in a wide spectrum of SCC problems (with nominal or continuous attributes, linear or nonlinear separation, and low or high dimensions) (Section 4).

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvm

1.3 Notation

We use the following notation throughout this paper.

- x is a data instance such that $x \in U$.
- V is a subspace for positive class within U, from which positive data set P is sampled.
- U (unlabeled data set) is a uniform sample of the universal set.
- U is the feature space for the universal set such that $U \subset R^m$ where m is the number of dimensions.

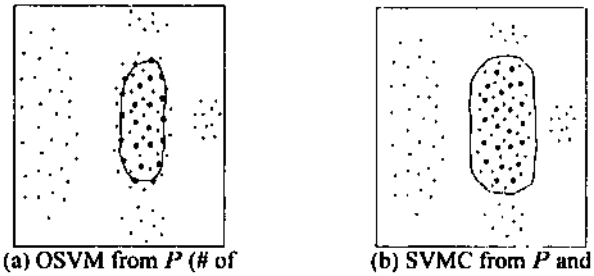
For an example of Web page classification, the universal set is the entire Web, U is a uniform sample of the Web, P is a collection of Web pages of interest, and $x \in R^m$ is an instance of a Web page.

2 Mapping Convergence (MC) Framework

2.1 Motivation

In machine learning theory, the "optimal" class boundary function (or hypothesis) $h(x)$ given a limited number of training data set $\{(x,l)\}$ (l is the label of x) is considered the one that gives the best *generalization* performance which denotes the performance on "unseen" examples rather than on the training data. The performance on the training data is not regarded as a good evaluation measure for a hypothesis because the hypothesis ends up *overfitting* when it tries to fit the training data too hard. (When a problem is easy (to classify) and the boundary function is complicated more than it needs to be, the boundary is likely overfitted. When a problem is hard and the classifier is not powerful enough, the boundary becomes undcrfit.) SVM is an excellent example of supervised learning that tries to maximize the generalization by maximizing the *margin* and also supports nonlinear separation using advanced kernels, by which SVM tries to avoid overfitting and underfitting [Burges, 1998].

The "optimal" SCC classifier without labeled negative data also needs to maximize the generalization somehow with highly expressive power to avoid ovcrfitting and undcrfitting. To illustrate an example of the "near-optimal" SCC boundary without labeled negative data, consider the synthetic data set (Figure 2) simulating a real situation where within U, (1) the universal set is composed of multiple groups of data, (2)



(a) OSVM from P (# of SVs = 7)

(b) SVMC from P and U (# of SVs = 75)

Figure 2: Synthetic data set simulating a real situation. *P: big dots, U: all dots (big and small dots)*

the positive class *V* is one of them (supposing *V* is the data group in the center), and (3) the positive data set *P* is a sample from *V* (assuming that the big dots are the sample *P).* OSVM draws *V,* a tight boundary around *P,* as shown in Figure 2(a), which overfits the true class area *V* due to the absence of the knowledge of the distribution of *U.* However, the "near-optimal" SCC classifiers must locate the boundary between *V* and *U* outside *V* (Figure 2(b)) and thus maximize the generalization. The MC framework using *U* systematically draws the boundary of Figure 2(b).

2.2 Negative Strength

Let *h(x)* be the boundary function of the positive class in *U,* which outputs the distance from the boundary to the instance *x* in *U* such that

$$h(x) > 0 \quad \text{if } x \text{ is a positive instance,}$$
$$h(x) < 0 \quad \text{if } x \text{ is a negative instance,}$$
$$|h(x)| > |h(x')| \quad \text{if } x \text{ is located farther than } x' \text{ from the boundary in } U.$$

Definition 1 (Strength of negative instances). *For two negative instances x and x' such that h(x) < 0 and h(x') < 0, if\h(x)\ > \h(x')\, then x is stronger than x'.*

Example 1. *Consider a resume page classification function h(x) from the Web (U). Suppose there are two negative data objects x and x' (non-resumepages) in U such that h(x) < 0 and h(x') < 0: x is "how to write a resume" page, and x' is "how to write an article " page. In U, x' is considered more distant from the boundary of the resume class because x has more relevant features to the resume class (e.g., the word "resume " in text) though it is not a true resume page.*

2.3 MC Framework

The MC framework is composed of two stages: the *mapping stage* and the *convergence stage.* In the mapping stage, the algorithm uses a weak classifier Ψ_1, which draws an initial approximation of "strong negatives" - the negative data located far from the boundary of the positive class in *U* (steps 1 and 2 in Figure 4). Based on the initial approximation, the convergence stage runs in iteration using a second base classifier $2, which maximizes the margin to make a progressively better approximation of negative data (steps 3 through 5 in Figure 4). Thus the class boundary eventually converges to

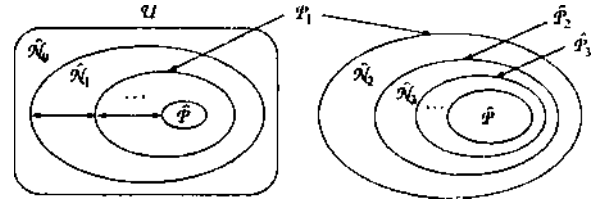


Figure 3: Example of the spaces of the MC framework in *U*

the boundary around the positive data set in the feature space which also maximizes the margin.

Input: - positive data set *P,* unlabeled data set *U*
Output: - a boundary function h,

Ψ_1: an algorithm identifying "strong negatives" from *U*
Ψ_2: a supervised learning algorithm that maximizes the margin

Algorithm:
1. Use Ψ_1 to construct a classifier *ho* from *P* and *U* which classifies only "strong negatives" as negative and the others as positive
2. Classify *U* by h_0
   * $\hat{N}_0$ examples from *U* classified as negative by *ho*
   * $\hat{P}_0$:= examples from *U* classified as positive by *ho*
3. Set TV := $\emptyset$ and *i* := 0
4. Do loop
   4.1. *N* := *NUN,*
   4.2. Use Ψ_2 to construct h_i 1 from *P* and TV
   4.3. Classify $\hat{P}_i$ by h_{i+1}
      * N_{i+1} := examples from P_i classified as negative by h_{i+1}
      * $\hat{P}_{i+1}$: examples from P_x classified as positive by h_{i+1}
   4.4. *i* := *i* + 1
   4.5. Repeat until $\hat{N}_i = \emptyset$
5. return h_i

Figure 4: MC framework

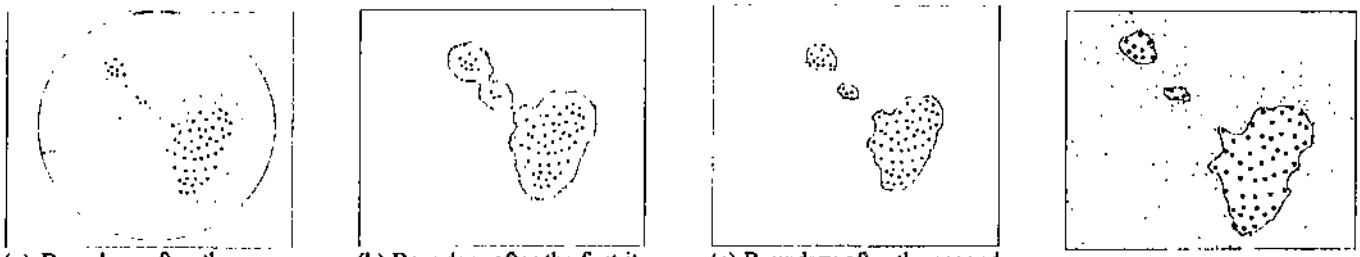
Assume that *V* is a subspace tightly subsuming *P* within *U* where the class of the boundary function for *V* is from the algorithm Ψ_2 (e.g., SVM). In Figure 4, let N_o be the negative space and $\hat{P}_0$ be the positive space within *U* divided by h_0 (a boundary drawn by Ψ_1), and let N*i* be the negative space and $\hat{P}_i$ be the positive space within $\hat{P}_{i-1}$ divided by h_i (a boundary drawn by Ψ_2). Then, we can induce the following formulae from the MC framework of Figure 4. (Figure 3 illustrates an example of the spaces of the framework in *U.)*

$$U = \hat{P}_i + \bigcup_{k=0}^{i} \hat{N}_k \qquad (1)$$

$$\hat{P}_i = \hat{P} + \bigcup_{k=i+1}^{I} \hat{N}_k \qquad (2)$$

where I is the number of iterations in the MC framework.

Theorem 1 (Boundary Convergence). *Suppose U is uniformly distributed in U. If algorithm does not generate*



(a) Boundary after the mapping stage

(b) Boundary after the first iteration

(c) Boundary after the second iteration

(d) Final boundary

Figure 5: Intermediate results of SVMC

false negatives, and algorithm Ψ_2 maximizes margin, then (I) the class boundary of the MC framework converges into the boundary that maximally separates P and U outside $\hat{\mathcal{P}}$, and (2) I (the number of iterations) is logarithmic in the margin betweer $\hat{\mathcal{N}}_0$ and $\hat{\mathcal{P}}$.

Proof $\hat{\mathcal{N}}_0 \cap \hat{\mathcal{P}} = \emptyset$ because a classifier h_0 constructed by the algorithm Ψ_1 does not generate false negative. A classifier h_1 constructed by the algorithm Ψ_2, trained from the separated space $\hat{\mathcal{N}}_0$ and $\hat{\mathcal{P}}$, divides the rest of the space $(\mathcal{U} - (\hat{\mathcal{N}}_0 + \hat{\mathcal{P}})$ which is equal to $\cup_{k=1}^{I}\hat{\mathcal{N}}_k)$ into two classes with a boundary that maximizes the margin between $\hat{\mathcal{N}}_0$ and $\hat{\mathcal{P}}$. The first part becomes $\hat{\mathcal{N}}_1$ and the other becomes $\cup_{k=2}^{I}\hat{\mathcal{N}}_k$. Repeatedly, a classifier h_{i+1} constructed by the same algorithm Ψ_2, trained from the separated space $\cup_{k=0}^{i}\hat{\mathcal{N}}_k$ and $\hat{\mathcal{P}}$, divides the rest of the space $\cup_{k=i+1}^{I}\hat{\mathcal{N}}_k$ into $\hat{\mathcal{N}}_{i+1}$ and $\cup_{k=i+2}^{I}\hat{\mathcal{N}}_k$ with equal margins. Thus, $\hat{\mathcal{N}}_{i+1}$ always has the margin of half of $\hat{\mathcal{N}}_i$ (for $i \geq 1$). Therefore, I will be logarithmic in the margin between $\hat{\mathcal{N}}_0$ and $\hat{\mathcal{P}}$.

The iteration stops when $\hat{\mathcal{N}}_i = \emptyset$, where there exists no sample of *U* outside $\hat{\mathcal{P}}$. Therefore, the final boundary will be located between *P* and *U* outside $\hat{\mathcal{P}}$ while maximizing the margin between them. □

Theorem 1 proves that under certain conditions, the final boundary will be located between *P* and *U* outside $\hat{\mathcal{P}}$. However, in the example of Figure 2(b), our framework generates the "better" boundary located between *P* and *U* outside *V* because in theorem 1, we made a somewhat strong assumption, i.e., *U* is uniformly distributed, to guarantee the boundary convergence. In a more realistic situation where there is some distance *S* between classes-Figure 2 shows some gaps between classes-if the *margin between h_{i+1} and $\hat{\mathcal{N}}_i$ becomes smaller than δ at some iteration, the convergence stops because $\hat{\mathcal{N}}_{i+1}$ becomes empty.* The margin between h_{i+1} and $\hat{\mathcal{N}}_i$ reduces by half at each iteration as the boundary h_{i+1} approaches to $\hat{\mathcal{P}}$ and thus the boundary is not likely to stop converging when it is far from $\hat{\mathcal{P}}$ unless *U* is severely sparse. Thus, we have the following claim:

Claim 1. *The boundary of MC is located between P and U outside V if U and P are not severely sparse and there exists visible gaps between V and U.*

Validity of the component algorithms Ψ_1 and Ψ_2

J. Ψ_1 must not generate false negatives.

Most classification methods have a threshold to control the trade-off between precision and recall. We can adjust the threshold of Ψ_1 so that it makes near 100% recall by sacrificing precision. (Some violations of this can be handled by the soft constraint of Ψ_2 (e.g., SVM).) Determining the threshold can be intuitive or automatic when not concerning the precision quality much. The precision quality of Ψ_1 does not affect the accuracy of the final boundary as far as it approximates a certain amount of negative data because the boundary will converge eventually. Figure 5 visualizes the boundary after each iteration of SVMC. The mapping stage only identifies very strong negatives by covering a wide area around the positive data (Figure 5(a)). (We used OSVM for the algorithm of the mapping stage. We intuitively set the parameters of OSVM such that it covers all the positive data without much concern for false positives.) Although the precision quality of mapping is poor, the boundary at each iteration converges (Figures 5(b) and (c)), and the final boundary is very close to the true boundary drawn by SVM with *P* and *N* (Figure 1(a) and 5(d)). Our experiments in Section 4 also show that the final boundary becomes very accurate although the initial boundary of the mapping stage is very rough by the "loose" setting of the threshold of .

2. 2 must *maximize margin.*

SVM and Boosting are currently the most popular supervised learning algorithms that maximize the margin. With a strong mathematical foundation, SVM automatically finds the optimal boundary without a validation process and without many parameters to tune. The small numbers of theoretically motivated parameters also work well for an intuitive setting. For these reasons, we use SVM for Ψ_2 for our research. In practice, the soft constraint of SVM is necessary to cope with noise or outliers. The soft constraint of SVM can affect / and the accuracy of the final boundary. However, *P* is not likely to have a lot of noise in practice because it is usually carefully collected by users. In our experiments, a low setting (i.e., ν = 0.01) of ν (the parameter to control the rate of noise in the training data) performs well for all cases for this reason. (We used I/-SVM for the semantically meaningful parameter [Chang and Lin, 2001].)

3 Support Vector Mapping Convergence (SVMC)

3.1 Motivation

The classification time of the final boundary of SMC ("Simple" MC with Ψ_2 = *SVM)* is equal to that of SVM because the final boundary is a boundary function of Ψ_2. The training time of SMC can be very long if $|U|$ is very large because the training time of SVM highly depends on the size of data set n ($\approx |U|$), and SMC runs iteratively. $t_{SMC} = O(|U|^2 * log|U|)$ assuming *the number of iterations* $\approx log|U|$ and $t_{SVM} = O(|f/|^2)$ where t_Ψ is the training time of a classifier Ψ. *(tsvM* is known to be at least quadratic to n and linear to the number of dimensions). Refer to [Chang and Lin, 2001] for more discussion on the complexity of SVM. However, decreasing the sampling density of U to reduce the training time hurts the accuracy of the final boundary because the density of *U* will directly affect the quality of the SVs of the final boundary.

3.2 SVMC

SVMC prevents the training time from increasing dramatically as the sample size grows. We prove that although SVMC iterates under the MC framework for the "near-optimal" result, its training time is independent of the number of iterations, and thus its training time is asymptotically equal to that of a SVM.

The approach of SVMC is to use minimally required data set at each iteration such that the data set does not degrade the accuracy of the boundary while it saves the training time of each SVM maximally. To illustrate how SVMC achieves this, consider the point of starting the third iteration (when $i = 2$) in SMC. (See step 4.1 in Figure 4.) After we merge $\hat{N}_2$ into *N,* we may not need all the data from *N* in order to construct h_3 because the data far from h3 may not contribute to the SVs. The set of negative SVs of h2 is the representative data set for $\hat{N}_0$ and $\hat{N}_1$, so we only keep the negative SVs of h2 and the newly induced data set $\hat{N}_2$ to support the negative side of/13.

Claim 2 (Minimally required negative data). *Minimally required negative data at* $(i+1)$*th* $(\forall i \geq 1)$ *that makes h{*+1 *is as accurate as the boundary constructed from* $\cup_{j=0}^{i}\hat{N}_j$ *and P, is* $\hat{N}_i \cup$ *negative support vectors of h{.*

Rationale. The negative SVs of h_{i+1} will be found in $\hat{N}_i$ the negative SVs of *hi* because $\hat{N}_i$ is the closest data set to /ij+i and because the directions not supported by $\hat{N}_i$ in the feature space will be supported by the negative SVs of *h^* which are the representing data set for $\cup_{j=0}^{i-1}\hat{N}_j$. However, if any of the negative SVs of *hi* is excluded in constructing h_{i+1}, h_{i+1} might suffer because the negative SVs of *hi* need to support the direction that TVj does not support in the feature space. Thus, $\hat{N}_i \cup$ *negative support vectors of hi* are the minimally required negative data set at *(i* + l)th iteration. □

For the minimally required data set for the *positive* side, we cannot definitely exclude any data object from *P* at each iteration because positive SVs are determined depending on negative SVs, and it is hard to determine the positive data that cannot be SVs independent of negative SVs or SVM parameters.

Surprisingly, adding the following statement between step 4.4 and 4.5 of the original MC framework of Figure 4 completes the SVMC algorithm.

Reset *N* with negative SVs of Ψ_2

Theorem 2 (Training time of SVMC). *Suppose* $t_{SVM} = O(n^2)$, *and* $|\hat{N}_{i+1}| = \frac{|\hat{N}_i|}{2}$, $\forall i \geq 1$. *Then,* $t_{SVMC} = O(n^2)$.

Proof. For simplicity of the proof, we approximate each value as follows.

$$n = |P| + |\cup_{i=0}^{I}\hat{N}_i| \approx |U|$$
$$|\hat{N}_0 \cup \hat{N}_1| \approx \frac{|U|}{2}$$

$$\begin{aligned} t_{SVMC} &= (\frac{|U|}{2})^2 + (\frac{|U|}{4})^2 + \ldots + (\frac{|U|}{2^I})^2 \\ &= \sum_{i=1}^{I}(\frac{|U|}{2^i})^2 = |U|^2\frac{1}{3}(1-\frac{1}{4^I}) \approx \frac{|U|^2}{3} \\ &= O(n^2) \end{aligned}$$

□

Theorem 2 states that the training complexity of SVMC is asymptotically equal to that of SVM. Our experiments in Section 4 also show that SVMC trains much faster than SMC while it remains the same accuracy. Figure 5 visualizes the boundary after each iteration of SVMC on the same data set of Figure 1.

4 Empirical Results

In this section, we show the empirical verification of our analysis on SVMC by extensive experiments on various domains of real data sets - Web page classification, letter recognition, and diagnosis of breast cancer - which show the outstanding performance of SVMC in a wide spectrum of SCC problems (with nominal or continuous attributes, linear or nonlinear separation, and low or high dimensions).

4.1 Datasets and Methodology

Due to space limitations, we reports only the main results. Our evaluation is based on the *F*\ measure ($F1 = 2pr/(p+r)$, *p* is precision and r is recall) as was used in [Liu *et al,* 2002] - one of the most recent works on SCC from positive and unlabeled data[2]. We also report the accuracy.

We used the letter recognition and breast cancer data sets from the UC1 machine learning repository[3] for direct comparisons with OSVM. (OSVM is often used for letter or digit

[2]Refer to [Liu *et al.,* 2002] for the justification of using the Fl measure for SCC.

[3] http://www.ics.uci.cdu/~mlcarn/M LRepository.html

Class	\P\	\u\	I'VI	F1, Accuracy (%)					T-Time (sec.)	
				TSVM	SMC	**SVMC**	OSVM	SVM.NN	**SMC**	SVMC
letter A	521	10007	384	0.9929, 99.96	**0.9840, 99.91**	**0.9840, 99.91**	0.8457, 99.22	0.0811,97.26	171.77	**45.37**
'B'	571	10004	377	0.9651,99.83	0.9046, 99.50	**0.9204, 99.59**	0.7207, 98.69	0.0834, 97.58	75.61	**14.34**
'C'	485	9996	371	0.9860, 99.23	**0.9641, 99.82**	**0.9641, 99.82**	0.7354, 98.82	0.0758, 97.55	155.70	**29.93**
'D'	525	10050	402	0.9820,99.91	**0.9300, 99.63**	**0.9300, 99.63**	0.6921,98.60	0.0902, 97.52	80.47	**16.06**
'E'	518	10026	392	0.9798, 99.90	**0.9419, 99.70**	0.9396, 99.69	0.7112,98.78	0.1333,97.64	98.13	**21.57**
b-canccr	135	259	77	0.9628, 98.87	**0.9585, 98.75**	**0.9585, 98.75**	0.6315,83.32	0.2434, 36.06	0.131	**0.025**
course	482	4179	448	0.8969, 97.82	0.8259, 96.65	**0.8434, 96.89**	0.2028, 36.59	0.0880, 89.30	636.70	**143.29**
faculty	532	4209	592	0.9032, 97.68	0.8420, 95.89	**0.8705, 96.58**	0.2621,50.11	0.0168,86.05	749.63	**217.85**
student	816	4154	825	0.9240, 97.27	0.8495, **94.29**	**0.8505,94.15**	0.3384, 32.52	0.0000,80.14	1181.59	**296.50**

Table 1: **Performance results.** \Pu\' # of positives in *U*\ T-Time: Training Time

recognition [Tax and Duin, 2001].) We also used Webkb[4] for Web page classification as used in [Liu *et ai,* 2002] for the indirect comparison with it. We set up the experiment environment in the same way as [Liu *et ai,* 2002], except our setup is more realistic: In [Liu *et ai,* 2002], *U* is composed of *b%* of positives (e.g., student) and samples from another class (e.g., course). Our *U* is *b%* of positives (e.g., student) and the remainder is from all other classes. (Refer to [Liu *et ai,* 2002] for the rest of the data set description.)

4.2 Results

Table 4 shows the performance results. TSVM (Traditional SVM) shows the ideal performance using SVM from *P* and manually classified *N* from *U.* SVMJMN (SVM with Noisy Negatives) is SVM from *P,* with *U* as a substitute for *N. (U* can be thought of as a good approximation of *N.)* Note that for TSVM, SVMJMN, and MC (SMC and SVMC), we used theoretically motivated fixed parameters without performing explicit optimization or validation process. For OSVM, we thoroughly searched for the best parameters based on the testing data set since optimizing parameters as specified in [Tax and Duin, 2001] is infeasibly inefficient especially in high dimensional spaces.

MC (SMC and SVMC) without labeled negative data show performance close to that of TSVM. SVMC trains much faster than SMC for most data sets. The performance of SMC and SVMC is comparable. (They differ a little because of the soft constraints of SVM and noise in the data.) OSVM performs fairly well on letter recognition and breast cancer (of low dimensionality with large amounts of data) but poor on Webkb (of high dimensionality). SVM.NN suffers from very low *F*\ scores because negative prediction dominates due to many false positives in the training data.

5 Conclusion

We present the MC framework and its instance algorithm SVMC, a new SCC method from positive and unlabeled data (without labeled negative data). SVMC without labeled negative data computes an accurate classification boundary around the positive data using the distribution of unlabeled data in a systematic way.

[4] http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/theo-11/www/wwkb/

References

[Burges, 1998] C. J. C. Burges. A tutorial on support vector machines **for pattern recognition.** ***Data Mining and Knowledge Discovery, 2:\2\-\67,*** **1998.**

[Chang and Lin, 2001] *C.-C.* Chang and C.-J. Lin. Training nu-support vector classifiers: Thoery and algorithms. *Neural Computation,* 13:2119-2147,2001.

[DeComitee/a/., 1999] F. DeComite, F. Denis, and R. Gillcron. Positive and unlabeled examples help learning. In *Pwc. 11th* ***Int. Conf. Algorithmic Learning Theory (ALT'99),*** **pages 219—** 230, Tokyo, Japan, 1999.

[Denis, 1998] F. Denis. PAC learning from positive statistical **queries. In** ***Proc. 10th Int. Conf. Algorithmic Learning Theory*** *(ALT'99),* pages 112 126, Otzenhausen, Germany, 1998.

[Letouzey *etal,* 2000] F. Letouzey, F. Denis, and R. Gillcron. Learning from positive and unlabeled examples. In *Proc. 11th* ***Int. Conf. Algorithmic Learning Theory (ALT'00),*** **pages 11-30,** Sydney, Australia, 2000.

[Liu *et ai,* 2002] B. Liu, W. S. Lee, P. S. Yu, and X. Li. Partially supervised classification of text documents. In *Proc. 19th Int.* ***Conf Machine Learning (ICML'02),*** **pages 387-394, Sydney,** Australia, 2002.

[Manevitz and Yousef, 2001] L. M. Manevitz and M. Yousef. One-class SVMs for document classification. *Journal of Machine* ***Learning Research,*** **2:139-154, 2001.**

[Tax and Duin, 2001] D. M. J. Tax and R. P. W. Duin. Uniform object generation for optimizing one-class classifiers. *Journal of* ***Machine Learning Research,*** **2:155-173, 2001.**

[Tong and Koller, 2000] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. In ***Proc. 17th Int. Conf Machine Learning (ICML'00),*** **pages 999** 1006, Stanford, CA, 2000.

[Valiant, 1984] L. G. Valiant. A theory of the learnable. *Communications of the ACM,* 27:1134-1142, 1984.

[Yu *et ai,* 2002] H. Yu, J. Han, and K. C. Chang. PEBL: Positive-example based learning for Web page classification using SVM. **In** ***Proc. 8th Int. Conf Knowledge Discovery and Data Mining*** *(KDD'02),* pages 239-248, Edmonton, Canada, 2002.