Can Irrelevant Data Help Semi-supervised Learning, Why and How?

Haiqin Yang[1], Shenghuo Zhu[2], Irwin King[1,3], Michael R. Lyu[1]

[1]Computer Sciences and Engineering
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong
{hqyang, king, lyu}@cse.cuhk.edu.hk

[2]NEC Laboratories America
10080 N. Wolfe Rd, SW3-350
Cupertino, CA 95014, USA
zsh@sv.nec-labs.com

[3]AT&T Labs Research
201 Mission Street
San Francisco, CA, USA
irwin@research.att.com

ABSTRACT

Previous semi-supervised learning (SSL) techniques usually assume unlabeled data are relevant to the target task. That is, they follow the same distribution as the targeted labeled data. In this paper, we address a different and very difficult scenario in SSL, where the unlabeled data may be a mixture of data relevant or irrelevant to the target binary classification task. In our framework, we do not require explicitly prior knowledge on the relatedness of the unlabeled data to the target data. In order to alleviate the effect of the irrelevant unlabeled data and utilize the implicit knowledge among all available data, we develop a novel maximum margin classifier, named the *tri-class support vector machine* (3C-SVM), to seek an inductive rule to separate the target binary classification task well while finding out the irrelevant data by-product. To attain this goal, we introduce a new $\min$ loss function, which can relieve the impact of the irrelevant data while relying more on the labeled data and the relevant unlabeled data. This loss function can therefore achieve the maximum entropy principle. The 3C-SVM can then generalize standard SVMs, Semi-supervised SVMs, and SVMs learned from the universum as its special cases. We further analyze the property of 3C-SVM on why the irrelevant data can help to improve the model performance. For implementation, we make relaxation and approximate the objective by the convex-concave procedure, which turns the original optimization from integral programming problem to a problem by just solving a finite number of quadratic programming problems. Empirical results are reported to demonstrate the advantages of our 3C-SVM model.

Categories and Subject Descriptors

I.2.6 [**Learning**]: Induction; G.1.6 [**Optimization**]: Integer programming, Quadratic programming methods

General Terms

Algorithms, Experimentation, Performance

CIKM'11, October 24–28, 2011, Glasgow, Scotland, UK.

Keywords

Semi-supervised learning, Concave-convex procedure

1. INTRODUCTION

Traditional supervised learning usually needs a large number of labeled training samples to learn the inductive rule. However, labeling data is usually expansive and time consuming since it needs experts' knowledge. Due to the limited amount of labeled training samples, researchers have proposed various methods, such as active learning [20], transfer learning [17], and semi-supervised learning (SSL) [33], to resolve this problem. Active learning requires users' additional interaction to label the data during the training procedure. Transfer learning transfers the knowledge learned from related tasks to the target task, where the related tasks may need sufficient labeled data. On the contrary, semi-supervised learning needs the least labeled data. It automatically learns a model based on both labeled and unlabeled data. Currently, a variety of SSL methods, including EM with generative mixture models, co-training, Transductive/Semi-Supervised Support Vector Machines, and graph-based, have been proposed in the literature [5, 13, 31, 32, 35].

Previously proposed semi-supervised learning models usually assume unlabeled data are relevant to the target task, i.e., they follow the same distribution as the target labeled data [1, 2]. This assumption implicitly indicates that the unlabeled data are well prepared [5, 33]. That is, the unlabeled data has excluded all irrelevant data, which follow distributions different from the target labeled data. However, in real world applications, without carefully preprocessing, irrelevant data are easy to be included as unlabeled data. For example, when crawling web pages as unlabeled data to help classifying corresponding categories, it is very easy to collect some irrelevant web pages for them. Similarly, as illustrated in Fig. 1(a) and Fig. 1(b), when we classify the digits "5" and "8" with the help of unlabeled digits. It is possible to include other digits into unlabeled data. In these cases, learning from the labeled and the mixed unlabeled data indeed do harm to the previously proposed semi-supervised learning models [15, 21].

Hence, it is important to resolve the effect of the irrelevant data in the mixed unlabeled data. To achieve this goal and to utilize the irrelevant data, in this paper, we propose a novel maximal margin semi-supervised classifier, named the *tri-class support vector machine* (3C-SVM). The 3C-SVM can find the inductive rule to separate the targeted binary data well while determining the irrelevant data as the 0-class. More specifically, we introduce a novel $\min$ loss function to measure the empirical risk on the unlabeled data. This loss function can take advantages of the symmetrical

(a) USPS (b) MNIST (c) Classifiers illustration

Figure 1: Illustration of two benchmark handwritten digit datasets and classifiers trained by different models on a mixed unlabeled synthetic dataset. In Fig. 1(a) and Fig. 1(b), block digits are labeled data on the target task, while black digits are unlabeled data mixed with other digits. The irrelevant digits affect the model in seeking optimal decision. Figure (c) illustrates that 3C-SVM (the thin solid line) attains the best result, which is closest to the Bayesian optimal classifier (the thick solid line), among all SVM-related classifiers and automatically distinguishes the irrelevant unlabeled data (black dots with red circles) well.

hinge loss function and the ε-insensitive loss function. More importantly, our model can achieve the maximum entropy principle, i.e., the decision function can rely more on the labeled data and the relevant data, while maximally ignore the irrelevant data.

We highlight the main contributions of our work as follows:

- First, our 3C-SVM generalizes several popular maximum margin classifiers, including standard SVMs, Semi-supervised SVMs (S^{3}VMs), and SVMs learned from universum data ($\mathcal{U}$-SVMs).
- Second, we provide theoretical analysis on 3C-SVM to indicate why the irrelevant data can help the SSL, i.e., the irrelevant data play the role of seeking a good subspace of the decision boundary.
- Third, the original formulation of 3C-SVM is an integer programming problem. We relax the objective and solve it by the concave-convex procedure (CCCP) [29]. This finally transforms it by solving a finite number of quadratic programming (QP) problems, which yields the same worst case time complexity as that of S^{3}VMs [6].
- Fourth, we conduct a series of empirical evaluation to demonstrate the advantages of the 3C-SVM.

The rest of the paper is organized as follows: In Section 2, we review the related work on learning from both labeled and universum data. In Section 3, the proposed 3C-SVM with its properties is presented. In section 4, we detail how to solve 3C-SVM algorithm through CCCP. We report the experimental comparison and results in section 5 and conclude the paper in section 6.

2. RELATED WORK

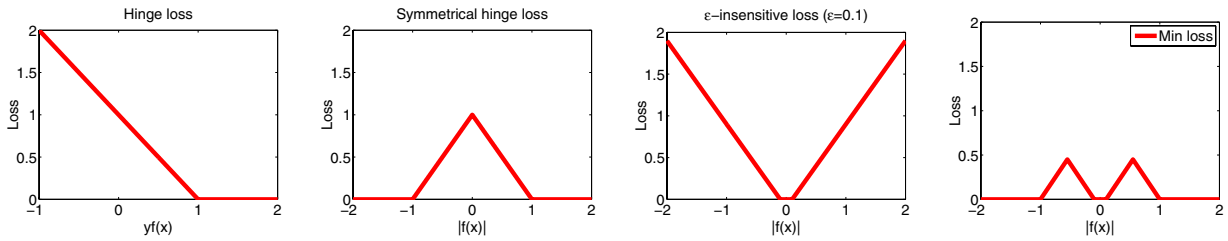
In literature, many models have been proposed to learning a binary classifier with the help from other auxiliary data. These models usually only work when the auxiliary data are "clean" and satisfy the models' assumption [5, 25].

A typical kind of methods is semi-supervised learning [5, 34], including generative methods for SSL [12, 16], graph-based SSL methods [2, 35], maximum margin classifiers [5, 6, 11], etc. These methods utilize the labeled data and the unlabeled data to improve model performance. Usually, they assume that the given auxiliary data follow the same distribution as the labeled data [5, 35]. However, when unlabeled data are mixed with irrelevant data, they will hurt the SSL models [21]. An illustration is shown as the dash line in Figure 1(c).

Another line of work is the $\mathcal{U}$-SVMs [27], which learns from labeled data and the universum data, a third kind of data whose distribution is different from neither of the positive class nor the negative class. The universum can play the role of seeking the subspace for the decision function [22], but they have to be specified explicitly and chosen carefully before the training. Hence, without prior knowledge on the label of the given auxiliary data, e.g., data may be mixed with universum data and relevant data, the relevant data will also disturb the $\mathcal{U}$-SVM eventually; see e.g., the dash-dot line in Figure 1(c).

The third line of work is similar to what we consider in this paper, where the unlabeled data is noisy. In [30], a graph-based semi-supervised learning model is proposed to learn from both labeled and unlabeled data, where the unlabeled data is assumed following the same distribution of the targeted binary classification task and the labeled data contain the universum data. This model needs to explicitly indicate the label of the universum data. In [10], a semi-supervised support vector machine is extended to learn from both labeled and mixed unlabeled data. The proposed model is solved by a Semi-Definite Programming (SDP) problem, whose time complexity scales to $\mathcal{O}((L+U^2)^2(L+U)^{2.5})$ (L and U denote the number of labeled and unlabeled data, respectively.), the same as that of the relaxed transductive SVM by SDP [4]. In [14], the safe semi-supervised support vector machine method is proposed to alleviate the effect of the noise in the unlabeled data. The proposed method consists of two steps: 1) to train a SVM and a S^{3}VM simultaneously; 2) to determine the label of a data point by the confidence of the SVM or the S^{3}VM. This method needs a postprocess on the results and do not consider the case of mixture unlabeled data.

In summary, previously proposed methods contain insufficiency in solving the problem of semi-supervised learning with mixed unlabeled data. Hence, in this paper, we try to alleviate the effect of those unspecified irrelevant data and utilize them in determining the decision function.



(a) Hinge loss (b) Symmetric hinge loss (c) ε-insensitive loss (d) Min loss

Figure 2: Illustration of different loss functions, including hinge loss, symmetrical hinge loss, ε-insensitive loss, and our proposed $\min$ loss.

3. LEARNING WITH IRRELEVANT DATA

In this section, we first define the problem setup of learning with irrelevant data. Next, we formulate the problem and propose the tri-class support vector machine, namely 3C-SVM. After that, we study the properties of 3C-SVM.

3.1 Problem Statement and Formulation

Suppose we are given two sets of data, labeled data $\mathcal{L}$ and unlabeled data $\mathcal{U}$, where the set of labeled data consists of L labeled samples, $\mathcal{L} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{L}$, and the set of unlabeled data consists of U unlabeled samples, $\mathcal{U} = \{\mathbf{x}_i\}_{i=L+1}^{L+U}$. Here, $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^d$, and the label is triple, i.e., $y_i \in \{-1, 0, 1\}$. Hence, the labeled data consist of two sets of data, $\mathcal{L}_{\pm 1}$ and $\mathcal{L}_0$, where data in $\mathcal{L}_{\pm 1}$ follows the same distribution in the target task and they are labeled by -1 or $+1$; while data in $\mathcal{L}_0$ are irrelevant to the target task. That is, data with distributions different from the labeled target data are all cast into 0-class to construct the $\mathcal{L}_0$ dataset. Similarly, unlabeled data are a mixture of these data. We denote them as $\mathcal{U} = \mathcal{U}_{\mathcal{L}} \cup \mathcal{U}_0$, where data in $\mathcal{U}_{\mathcal{L}}$ follow the same distribution of ± 1 data, and data in $\mathcal{U}_0$ follow distributions different from the ± 1 data. Normally, the number of unlabeled data is much larger than the number of the labeled target data, i.e., $|\mathcal{L}_{\pm 1}| \ll U$, and given an unlabeled data point, one does not know whether it comes from $\mathcal{U}_{\mathcal{L}}$ or from $\mathcal{U}_0$.

Here, the goal is to seek a decision boundary, $f_{\boldsymbol{\vartheta}}(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b$, to classify the ± 1 data well with the help of given labeled and mixed unlabeled data, where $\boldsymbol{\vartheta} = (\mathbf{w}, b)$ and $\phi : \mathbb{R}^d \to \mathbb{R}^f$, is a feature mapping function often implicitly defined by a Mercer kernel [19, 24]. Hence, we formulate the objective as follows:

$$\min_{\boldsymbol{\vartheta}} \ \frac{\lambda}{2}\|\mathbf{w}\|^2 + \sum_{\mathbf{x}_i \in \mathcal{L}} r_i \ell_L(f_{\boldsymbol{\vartheta}}(\mathbf{x}_i), y_i) + \sum_{\mathbf{x}_i \in \mathcal{U}} r_i \ell_U(f_{\boldsymbol{\vartheta}}(\mathbf{x}_i)) \,, \quad (1)$$

where minimizing $\|\mathbf{w}\|^2$ corresponds to maximizing the margin width [24] and avoids the overfitting. The parameter λ is a trade-off constant for the regularization term. $\ell_L(\cdot, \cdot)$ is a loss function to measure the empirical risk of the labeled data and $\ell_U(\cdot)$ is a loss function to measure the empirical risk of the unlabeled data. $r_i, i = 1, \ldots, L + U$, is a ratio penalty to balance the loss on the point $\mathbf{x}_i$ and the regularization term.

Typically, one may choose different loss functions to measure the empirical risk on the given data. These loss functions include

- **Hinge loss:** $H_1(u) = \max\{0, 1 - u\}$, a loss function has been used to measure the empirical risk of labeled data in standard SVMs [24]; see Figure 2(a).
- **Symmetric hinge loss:** $H_1(|\cdot|)$, a loss function has been applied to measure the empirical risk on unlabeled data for S^{3}VMs [6]; see Figure 2(b).
- **ε-insensitive loss:** $I_\varepsilon(u) = \max\{0, |u| - \varepsilon\}$, a loss function has been adopted to measure the empirical risk in Support Vector Regression [24] and the Universum data in $\mathcal{U}$-SVMs [27]; see Figure 2(c).

In our problem setup, the unlabeled data may be a mixture of data relevant or irrelevant to the target task. Actually, this assumption matches natural to normal scenarios when the unlabeled data do not well prepared. However, without prior knowledge, how to distinguish the relevant and irrelevant data correctly is a very difficult task.

Here, in order to relieve the effect of irrelevant unlabeled data, we try to utilize them based on the following two principles. First, from *logistic regression* perspective [9, 18], when a data point lies farther away from the decision boundary, the data is more likely to be classified as data from ± 1-class; while data points lie near the decision boundary, they are less confident to be classified correctly. Hence, ideally, data from ± 1-class should lie on or outside of the margin gap; while other irrelevant data are close to the decision boundary. Second, the *maximum entropy principle* indicates that a classifier should rely more on the labeled and relevant data, while maximally ignore the irrelevant data. These two principles indicate that irrelevant data should lie around the sought decision boundary.

In order to fulfill the above principles, we adopt a $\min$ loss function to measure the risk of unlabeled data, so as to separate the unlabeled data into relevant and irrelevant data. This loss function determines and measures the error of an unlabeled data point by the min value of the symmetric hinge loss and the ε-insensitive loss (see Figure 2(d)):

$$\ell_{\min}(\mathbf{x}) = \min\left\{H_1(|f_{\boldsymbol{\vartheta}}(\mathbf{x}_i)|), I_\varepsilon(|f_{\boldsymbol{\vartheta}}(\mathbf{x}_i)|)\right\}. \quad (2)$$

Hence, for an unlabeled data point, when the error measured by the ε-insensitive loss is smaller than the error measured by the symmetric hinge loss, we can deem it as irrelevant data; otherwise, we set it as relevant data.

With this loss function, we can develop a novel maximum margin classifier, named the *tri-class support vector machine* (3C-SVM), as follows:

$$\min_{\boldsymbol{\vartheta}} \ \frac{\lambda}{2}\|\mathbf{w}\|^2 + \sum_{\mathbf{x}_i \in \mathcal{L}_{\pm 1}} r_i H_1(y_i f_{\boldsymbol{\vartheta}}(\mathbf{x}_i)) + \sum_{\mathbf{x}_i \in \mathcal{L}_0} r_i I_\varepsilon(f_{\boldsymbol{\vartheta}}(\mathbf{x}_i)) + \sum_{\mathbf{x}_i \in \mathcal{U}} r_i \min\left\{H_1(|f_{\boldsymbol{\vartheta}}(\mathbf{x}_i)|), I_\varepsilon(|f_{\boldsymbol{\vartheta}}(\mathbf{x}_i)|)\right\}. \quad (3)$$

In the above, the first two terms correspond to the formulation of a standard SVM [24]. The third term measures the empirical risk of $\mathcal{L}_0$ data, the same in $\mathcal{U}$-SVMs [27]. The last term measures the loss of unlabeled data. Hence, we can determine the class of a data

Table 1: Relation between different models and the training data.

3C-SVM			
$\mathcal{L}$	-1	0	1
$\mathcal{U}$	-1	0	1

SVM		
$\mathcal{L}$	-1	1
$\mathcal{U}$		

S^3VM		
$\mathcal{L}$	-1	1
$\mathcal{U}$	-1	1

$\mathcal{U}$-SVM			
$\mathcal{L}$	-1	0	1
$\mathcal{U}$			

point $\mathbf{x}$ by the following criterion:

$$c(\mathbf{x}) = \begin{cases} +1 & \text{if } f_{\vartheta}(\mathbf{x}) > \frac{1+\varepsilon}{2} \\ -1 & \text{if } f_{\vartheta}(\mathbf{x}) < -\frac{1+\varepsilon}{2} \\ 0 & \text{otherwise} \end{cases}. \quad (4)$$

The above criterion separate the data into three classes. The 0-class data corresponds to the irrelevant data.

3.2 Properties of 3C-SVMs

We first list the generalization property of our 3C-SVM, then outline the intuition behind 3C-SVMs through a specific case with $r_i = \infty$ for unlabeled data and $\varepsilon = 0$.

Our 3C-SVM provides a framework for the following popular maximum margin classifiers:

1. A standard SVM formulation [24] is a special case of the 3C-SVM. This can be attained by setting r_i to zero for the third and fourth terms in (3). When only labeled data are given in the training set, we can adopt this formulation.

2. An S^3VM formulation [5] is a special case of the 3C-SVM. This can be achieved by setting r_i to zero in the third term and using only symmetrical hinge loss to measure the empirical risk of unlabeled data in the fourth term in (3). When only labeled data and relevant unlabeled data are given, we can use this formulation.

3. The 3C-SVM also includes a $\mathcal{U}$-SVM [27]. It can be easily obtained by setting r_i to zero in the fourth term of (3). This formulation works when only labeled data and universum data are given.

Hence, our 3C-SVM, a general maximum margin semi-supervised learning formulation, includes standard SVMs, S^3VMs, and $\mathcal{U}$-SVMs as its special cases. A summarization is shown in Table 1.

Intuitively, our 3C-SVM may not work for all the cases since it seems that it requires the irrelevant data falls close to the decision function. However, we claim that we can tackle this problem by mapping the original data to a suitable space through the kernel trick. We then study why the 3C-SVM possibly works and give an insight of the model in the following theorem:

THEOREM 1. *A 3C-SVM with $r_i = \infty$ for unlabeled data and $\varepsilon = 0$ is equivalent to one of the following two cases: 1) training a general S^3VM to keep the unlabeled data falling on or out of the margin gap with only one or non of the unlabeled data in the decision boundary; or 2) separating the unlabeled data into two sets, $\mathcal{U}_{\mathcal{L}}$ and $\mathcal{U}_0$ with $|\mathcal{U}_0| \geq 2$, and training a general S^3VM on the training data projected onto the orthogonal complement of span $\{\phi(\mathbf{x}_j) - \phi(\mathbf{x}_0),\ \mathbf{x}_j \in \mathcal{U}_0\}$, where $\mathbf{x}_0$ is an arbitrary data point from $\mathcal{U}_0$, while keeping the unlabeled data in the set of $\mathcal{U}_{\mathcal{L}}$ falling on or out of the margin gap.*

Proof: $r_i = \infty$ for $\mathcal{U}$ data and $\varepsilon = 0$ imply that the min term in the fourth term of (3) vanish and the optimal solution of $\mathbf{w}$ and b in (3) is attained when one of the following conditions is fulfilled: (a) $|\mathbf{w}^\top \phi(\mathbf{x}_j) + b| \geq 1$, or (b) $\mathbf{w}^\top \phi(\mathbf{x}_j) + b = 0$. Hence, the above conditions set up the criterion of separating the unlabeled data into two sets, $\mathcal{U}_{\mathcal{L}}$ and $\mathcal{U}_0$, where data in $\mathcal{U}_{\mathcal{L}}$ satisfy the condition of (a) and data in $\mathcal{U}_0$ satisfy the condition of (b).

First, if $|\mathcal{U}_0| = 0$, or 1, it leads to the result of case 1) in the above theorem. Here, a general S^3VM means that it is a generalization of the S^3VM and the $\mathcal{U}$-SVM.

Next, if $\mathcal{U}_0$ contains at least two samples. For the data $\mathbf{x}_j$ from $\mathcal{U}_0$, we have $\mathbf{w}^\top \phi(\mathbf{x}_j) + b = 0$. Hence, picking arbitrary data $\mathbf{x}_0$ from $\mathcal{U}_0$, we obtain $\mathbf{w}^\top (\phi(\mathbf{x}_j) - \phi(\mathbf{x}_0)) = 0$. That is, $\mathbf{w}$ is orthogonal to span $\{\phi(\mathbf{x}_j) - \phi(\mathbf{x}_0),\ \mathbf{x}_j \in \mathcal{U}_0\}$. Now, let $P_{\mathcal{U}_0^\perp}$ denote an orthogonal project on the orthogonal complement of the mapped set $\mathcal{U}_0$, we have $\mathbf{w} = P_{\mathcal{U}_0^\perp}\mathbf{w}$, $\mathbf{w}^\top \mathbf{w} = \mathbf{w}^\top P_{\mathcal{U}_0^\perp}^\top P_{\mathcal{U}_0^\perp}\mathbf{w} = \mathbf{w}^\top \mathbf{w}$, and $\mathbf{w}^\top \mathbf{x}_i = \mathbf{w}^\top P_{\mathcal{U}_0^\perp}^\top \mathbf{x}_i = \mathbf{w}^\top P_{\mathcal{U}_0^\perp}\mathbf{x}_i$. This means that the optimal $\mathbf{w}$ is sought by training a general S^3VM on the projected labeled data and $\mathcal{U}_{\mathcal{L}}$ data with projection by $P_{\mathcal{U}_0^\perp}$ while keeping the condition (a) valid, or other unlabeled data falling on or out the margin gap. ■

Theorem 1 clearly shows that the optimization of our proposed model eventually is to find the most suitable subspace in which the margin is maximized while the overall empirical risk is minimized. The irrelevant data play the role of finding the subspace.

4. SOLUTION AND COMPUTATION

Due to the non-convexity of the min loss function, the formulation of the 3C-SVM in (3) is non-convex in general. Moreover, there are two difficulties to be solved in the formulation, the min term and the *absolute operation* on the unlabeled data. In the following, we show how to solve these two difficult problems.

4.1 Elimination of Min Terms and Absolute Values

First, we introduce decision variables, $d_k \in \{0, 1\}$, to remove the min term. This trick is similar to the L_1-norm S^3VM in [3]. We then transform the optimization into a mixed integer programming problem as follows:

$$\begin{aligned} \min_{\vartheta, \mathbf{d}} \ & \frac{\lambda}{2}\|\mathbf{w}\|^2 + \sum_{\mathbf{x}_i \in \mathcal{L}_{\pm 1}} r_i H_1(y_i f_{\vartheta}(\mathbf{x}_i)) + \sum_{\mathbf{x}_i \in \mathcal{L}_0} r_i I_{\varepsilon}(f_{\vartheta}(\mathbf{x}_i)) \\ & + \sum_{\mathbf{x}_{kL} \in \mathcal{U}} r_{kL} \underbrace{H_1(|f_{\vartheta}(\mathbf{x}_{kL})| + D(1 - d_k))}_{Q_1} \\ & + \sum_{\mathbf{x}_{kL} \in \mathcal{U}} r_{kL} \underbrace{I_{\varepsilon}(|f_{\vartheta}(\mathbf{x}_{kL})| - Dd_k)}_{Q_2}, \end{aligned} \quad (5)$$

where $kL = k + L$, $D > 0$ is a suitable constant making $Q_1 = 0$ when $d_k = 0$ and $Q_2 = 0$ when $d_k = 1$. That means, when $d_k = 0$, the error is counted from Q_2 and the unlabeled data are classified as 0-class and its error is measured by the ε-insensitive loss function; when $d_k = 1$, the error is incurred by Q_1 and the unlabeled data are classified as one of the ± 1-class, where its error is measured by the symmetrical hinge loss function.

Next, we deal with the absolute terms in the loss function by considering the properties of the loss functions. The shifted symmetrical hinge loss function in Q_1 can be abstracted as $H_1(|u|+a)$, or $H_1(|u| + a) = \max\{0, 1 - |u| - a\} = H_{1-a}(|u|)$. It can be

approximated by a symmetrical loss, which is similar to the ramp loss used in [6, 26], as follows:

$$H_1(|u|+a) \approx H_{1-a}(u) - H_\kappa(u) + H_{1-a}(-u) - H_\kappa(-u)\,.$$

The shifted ε-insensitive loss function in Q_2 can be transformed to another symmetrical loss as follows:

$$I_\varepsilon(|u|-a) = H_{-\varepsilon-a}(-u) + H_{-\varepsilon-a}(u)\,.$$

Due to the symmetry of both loss functions, we introduce new pair-data for the unlabeled data to simplify the expression as [6]. The new pair-data are

$$\begin{aligned} \mathbf{x}_{kL} &= \mathbf{x}_{k+L}, \quad y_{kL} = 1, \\ \mathbf{x}_{kLU} &= \mathbf{x}_{k+L}, \quad y_{kLU} = -1, \;\; k = 1, \ldots, U, \end{aligned}$$

where kL means $k+L$ and kLU means $k+L+U$.

4.2 Concave-Convex Procedure (CCCP)

Hence, we can transform the problem in (5) into $Q^\kappa(\boldsymbol{\vartheta}, \mathbf{d})$, which is the summation of two terms, $Q_{vex}(\boldsymbol{\vartheta}, \mathbf{d})$ and $Q^\kappa_{cav}(\boldsymbol{\vartheta})$. They are defined as follows

$$\begin{aligned} &Q_{vex}(\boldsymbol{\vartheta}, \mathbf{d}) \\ &= \frac{\lambda}{2}\|\mathbf{w}\|^2 + \sum_{\mathbf{x}_i \in \mathcal{L}_{\pm 1}} r_i H_1(y_i f_{\boldsymbol{\vartheta}}(\mathbf{x}_i)) + \sum_{\mathbf{x}_i \in \mathcal{L}_0} r_i I_\varepsilon(f_{\boldsymbol{\vartheta}}(\mathbf{x}_i)) \\ &+ \sum_{k=1}^{U} r_{kL} H_{\tilde{1}}(y_{kL} f_{\boldsymbol{\vartheta}}(\mathbf{x}_{kL})) + \sum_{k=1}^{U} r_{kL} H_{\tilde{1}}(y_{kLU} f_{\boldsymbol{\vartheta}}(\mathbf{x}_{kLU})) \\ &+ \sum_{k=1}^{U} r_{kL} H_{\tilde{\varepsilon}}(y_{kL} f_{\boldsymbol{\vartheta}}(\mathbf{x}_{kL})) + \sum_{k=1}^{U} r_{kL} H_{\tilde{\varepsilon}}(y_{kLU} f_{\boldsymbol{\vartheta}}(\mathbf{x}_{kLU}))\big)\,, \\ &Q^\kappa_{cav}(\boldsymbol{\vartheta}) = -\sum_{j=L+1}^{L+2U} r_j H_\kappa(y_j f_{\boldsymbol{\vartheta}}(\mathbf{x}_j))\,. \end{aligned}$$

where $\tilde{1} = 1 - D(1-d_k)$ and $\tilde{\varepsilon} = -\varepsilon - Dd_k$.

Note that the above concave term, Q^κ_{cav}, keeps the non-convexity of the model following from the ramp loss in approximating the Q_1. The optimization in $Q^\kappa(\boldsymbol{\vartheta}, \mathbf{d})$ is a summation of a convex term and a concave term, or difference of convex programming. Hence, it can be solved by the concave-convex procedure (CCCP) [29], a technique has been adopted in large scale transductive SVMs [6] and SVMs on data with missing values [23].

In the CCCP, we need to use the first order Taylor expansion to approximate the concave term of Q^κ_{cav}. Since the variable $\mathbf{d}$ does not appear in the concave term, we only need to apply the first order Taylor expansion of Q^κ_{cav} at $\boldsymbol{\vartheta}^t$. Hence, we can seek the optimal variables by solving a sequence of the following optimization problem:

$$\min_{\boldsymbol{\vartheta}, \mathbf{d}} \left(Q_{vex}(\boldsymbol{\vartheta}, \mathbf{d}) + \frac{\partial Q^\kappa_{cav}(\boldsymbol{\vartheta}^t)}{\partial \boldsymbol{\vartheta}} \cdot \boldsymbol{\vartheta} \right), \tag{6}$$

The above optimization is a mixed integer optimization problem since $\mathbf{d}$ is an integer vector. Here, we adopt a standard routine to solve the integer programming problem [28]: 1) relaxing the integer variable to a real variable, then solve the whole optimization together; 2) rounding the corresponding variable to get its integer solution.

For 3C-SVM in (6), we relax the decision variable d_k from $\{0, 1\}$ to $[0, 1]$ and solve the optimization problem in (6) first. We then determine the value of d_k by its definition, the error incurred is less when the data is assigned to the associated class, as follows

$$d_k = \begin{cases} 1 & \text{if } \; \xi_k \le \xi^*_k \\ 0 & \text{otherwise} \end{cases}, \tag{7}$$

where $\xi_k = H_1(|f_{\boldsymbol{\vartheta}}(\mathbf{x}_{kL})|)$ and $\xi^*_k = I_\varepsilon(|f_{\boldsymbol{\vartheta}}(\mathbf{x}_{kL})|)$, $k = 1, \ldots, U$.

To simplify the first order approximation of the concave term in (6), we define

$$\mu_{k+\mathfrak{s}} = y_{k+\mathfrak{s}} \frac{\partial Q^\kappa_{cav}(\boldsymbol{\vartheta})}{\partial f_{\boldsymbol{\vartheta}}(\mathbf{x}_{k+\mathfrak{s}})} = \begin{cases} r_{k+\mathfrak{s}} & \text{if } y_{k+\mathfrak{s}} f_{\boldsymbol{\vartheta}}(\mathbf{x}_{k+\mathfrak{s}}) < \kappa \\ 0 & \text{otherwise} \end{cases}, \tag{8}$$

for those unlabeled samples $\mathbf{x}_{k+\mathfrak{s}}$ with $d_k = 1$, where $k = 1, \ldots, U$, and $\mathfrak{s}$ is L or $L+U$. Hence, the first order Taylor expansion of the concave term is then expressed as

$$\frac{\partial Q^\kappa_{cav}(\boldsymbol{\vartheta}^t)}{\partial \boldsymbol{\vartheta}} \cdot \boldsymbol{\vartheta} = \sum_{j=L+1}^{L+2U} \mu_j y_j f_{\boldsymbol{\vartheta}}(\mathbf{x}_j).$$

Now we turn to solve the relaxed optimization in (6) and summarize the result in the following theorem:

THEOREM 2. *The dual problem of the relaxed optimization in (6) is a Quadratic Programming (QP) problem as follows:*

$$\begin{aligned} \max_{\boldsymbol{\alpha}, \boldsymbol{\alpha}^*} \quad & -\frac{1}{2\lambda}[\boldsymbol{\alpha}; \boldsymbol{\alpha}^*]^\top \boldsymbol{\Omega} [\boldsymbol{\alpha}; \boldsymbol{\alpha}^*] + \boldsymbol{\varrho}^\top [\boldsymbol{\alpha}; \boldsymbol{\alpha}^*] \\ \textit{s.t.} \quad & \mathbf{0} \le \boldsymbol{\alpha}, \boldsymbol{\alpha}^* \le \mathbf{r}, \\ & \mathbf{A}_e\,[\boldsymbol{\alpha}; \boldsymbol{\alpha}^*] = \boldsymbol{\mu}^\top \mathbf{Y}_{\bullet 2U}, \\ & \mathbf{A}[\boldsymbol{\alpha}; \boldsymbol{\alpha}^*] \le \mathbf{0}\,, \end{aligned} \tag{9}$$

where the Lagrangian multipliers $[\boldsymbol{\alpha}; \boldsymbol{\alpha}^*]$ consists of an $|\mathcal{L}_0| + L + 4U$-dimensional vector. The matrix $\boldsymbol{\Omega}$ on the quadratic term is defined as $\boldsymbol{\Omega} = \begin{bmatrix} Q_{|\mathcal{L}_0|+L+2U, |\mathcal{L}_0|+L+2U} & Q_{|\mathcal{L}_0|+L+2U, 2U} \\ Q_{2U, |\mathcal{L}_0|+L+2U} & Q_{2U,2U} \end{bmatrix}$, *and the coefficient for the linear term is*

$$\boldsymbol{\varrho} = \frac{1}{\lambda} \begin{bmatrix} Q_{2U, |\mathcal{L}_0|+L+2U} \\ Q_{2U,2U} \end{bmatrix} \boldsymbol{\mu} + \begin{bmatrix} -\varepsilon \mathbf{1}_{2|\mathcal{L}_0|} \\ \mathbf{1}_{L-|\mathcal{L}_0|} \\ (1-D)\mathbf{1}_{2U} \\ -\varepsilon \mathbf{1}_{2U} \end{bmatrix},$$

$\mathbf{A}_e = [\mathbf{Y}; \mathbf{Y}_{\bullet 2U}]$ *and* $\mathbf{A} = [\mathbf{0}_{U,L}, \; -\mathbf{I}_U, \; -\mathbf{I}_U, \; \mathbf{I}_U, \; \mathbf{I}_U]$, $\mathbf{Y}$ *is a vector containing the label value of all training data including the expanding auxiliary labels, and* $\mathbf{Y}_{\bullet 2U}$ *denotes the last* $2U$*-element in* $\mathbf{Y}$.

The above theorem can be derived based on the standard Lagrangian multiplier method, where Eq. (9) corresponds to the dual form of the optimization on (6).

After solving the QP problem in (9), we obtain $\mathbf{w}$ as a linear combination of the dual variables, $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^*$,

$$\mathbf{w} = \frac{1}{\lambda} \left(\sum_{i=-|\mathcal{L}_0|, i \ne 0}^{L+2U} \alpha_i y_i \phi(\mathbf{x}_i) + \sum_{i=L+1}^{L+2U} (\alpha^*_i - \mu_i) y_i \phi(\mathbf{x}_i) \right), \tag{10}$$

and the variable b corresponds to the dual variable of the equality constraint. The form of the weight we obtained is similar to that in [2]. We can also define the corresponding *support vectors*. They are those labeled data $\mathbf{x}_i$'s with non-zero α_i values and unlabeled data $\mathbf{x}_j$'s with non-zero $(\alpha_j + \alpha^*_j - \mu_j)$ values.

Hence, we obtain Algorithm 1 to solve the 3C-SVM algorithm. Recalling Theorem 1, we can know that, intuitively, the Algorithm 1 works in the following way: 1) first finding out those unlabeled data which are certainly outside the margin gap, removing

them from the training set; 2) then training a $\mathcal{U}$-SVM model on the labeled data with the rest unlabeled data.

Moreover, the convergence of Algorithm 1 is guaranteed by the following theorem:

THEOREM 3. *The algorithm 1 converges in a finite number of iterations.*

Proof: First, we prove that the objective Q^κ decreases in each iteration. From the CCCP, we have

$$Q_{vex}(\boldsymbol{\vartheta}^{t+1},\mathbf{d}) + \partial Q^\kappa_{cav}(\boldsymbol{\vartheta}^t)\cdot\boldsymbol{\vartheta}^{t+1} \leq Q_{vex}(\boldsymbol{\vartheta}^t,\mathbf{d}) + \partial Q^\kappa_{cav}(\boldsymbol{\vartheta}^t)\cdot\boldsymbol{\vartheta}^t \quad (11)$$

$$Q^\kappa_{cav}(\boldsymbol{\vartheta}^{t+1}) \leq Q^\kappa_{cav}(\boldsymbol{\vartheta}^t) + \partial Q^\kappa_{cav}(\boldsymbol{\vartheta}^t)\cdot(\boldsymbol{\vartheta}^{t+1}-\boldsymbol{\vartheta}^t), \quad (12)$$

where ∂Q^κ_{cav} defines the partial derivative of Q^κ_{cav} with respect to $\boldsymbol{\vartheta}$. Hence, summing (11) and (12) together, we get $Q^\kappa(\boldsymbol{\vartheta}^{t+1},\mathbf{d}) \leq Q^\kappa(\boldsymbol{\vartheta}^t,\mathbf{d})$ for the same $\mathbf{d}$.

After rounding, the objective value Q^κ is $Q^\kappa(\boldsymbol{\vartheta}^{t+1},\mathbf{d}^{t+1})$. It may be greater than $Q^\kappa(\boldsymbol{\vartheta}^t,\mathbf{d}^t)$. In order to avoid this case, we restore $\mathbf{d}^{t+1}$ to $\mathbf{d}^t$ and seek $\boldsymbol{\vartheta}^{t+1}$ again by minimizing Q^κ with fixed $\mathbf{d}$. This additional step guarantees to decrease the objective of Q^κ at each step.

Second, the variable $\boldsymbol{\mu}$ can only take a finite number of distinct values. The algorithm converges in several steps since Q^κ decreases in each iteration and the inequality (12) is strict unless $\boldsymbol{\mu}$ remains unchanged. ■

Algorithm 1 CCCP for 3C-SVMs

Initialization:
$t = 0$
Calculate $\boldsymbol{\vartheta}^0 = (\mathbf{w}^0, b^0)$ from a $\mathcal{U}$-SVM solution on the labeled/unlabeled data
Compute
$\mu_i^0 = \begin{cases} r_i \text{ if } y_i f_{\boldsymbol{\vartheta}^0}(\mathbf{x}_i) < \kappa \text{ and } i \geq L+1 \\ 0 \text{ otherwise} \end{cases}$
repeat
$t \leftarrow t+1$
Solve the optimization in (9) to obtain $\boldsymbol{\vartheta}^t$
Update $\mathbf{d}^t$ from (7)
Update $\boldsymbol{\mu}^t$ from (8)
while $Q^\kappa(\boldsymbol{\vartheta}^t,\mathbf{d}^t) > Q^\kappa(\boldsymbol{\vartheta}^{t-1},\mathbf{d}^{t-1})$ **do**
Restore $\mathbf{d}^t$ to $\mathbf{d}^{t-1}$
Update $\boldsymbol{\mu}^{t-1}$ from (8) with $\boldsymbol{\vartheta}^t$
Solve the optimization in (9) to obtain $\boldsymbol{\vartheta}^t$
Update $\mathbf{d}^t$ from (7)
Update $\boldsymbol{\mu}^t$ from (8)
end while
until $|\boldsymbol{\mu}^{t+1} - \boldsymbol{\mu}^t| \leq \epsilon$

Remark Note that the local optimal issue of the 3C-SVM has been alleviated by its initialization and the additional step to avoid increasing the rounded objective function is not needed usually. Our observation from the experimental results shows that our 3C-SVM works well using current initialization and the rounded objective function, $Q^\kappa(\boldsymbol{\vartheta}^t,\mathbf{d}^t)$, actually decreases in each step; see empirical study in Section 5.

Complexity Analysis Algorithm 1 has to solve a sequence of QPs in (9). In practice, we find that the number of iteration steps is a constant, usually less than 10; see Figure 4. Thus, training a 3C-SVM is equivalent to solving a constant number of QP problems with $|\mathcal{L}_0| + L + 4U$ variables. Therefore, the 3C-SVM algorithm has a worst case complexity of $\mathcal{O}((|\mathcal{L}_0| + L + 4U)^3)$ [8, 19]. Possible tricks may be applied to speed up the 3C-SVM algorithm in a quadratic scale [6, 19]. Furthermore, by exploring the sparsity structure among the dual variables, we can reduce the number of variables to the number of non-zero variables. This can reduce the computation cost of 3C-SVM largely.

4.3 Balance Constraint

In the formulation of (3), we do not consider the balance constraint for the unlabeled data. Actually, balance constraint can be easily incorporated in our formulation.

There are two observations: 1) Data from $\mathcal{U}_\mathcal{L}$ need the balance constraint [25]; 2) Data from $\mathcal{U}_{\mathcal{L}_0}$ do not need the balance constraint. By Theorem 1, ideally, the decision values of $\mathcal{U}_{\mathcal{L}_0}$ data approach to zero. Hence, summarizing the decision values of all unlabeled data, we can obtain the same balance constraint as that used in [7],

$$\frac{1}{U}\sum_{t=L+1}^{L+U} f_{\boldsymbol{\vartheta}}(\mathbf{x}_t) = \frac{1}{L}\sum_{i=1}^{L} y_i. \quad (13)$$

This constraint can be easily included in the optimization of (6) and rewrite into kernel form in (9) similar to the trick in [7].

It is noted that the balance constraint in (13) is affected by the summation of y_i. A possible better setting for the balance constraint is $\frac{1}{U}\sum_{t=L+1}^{L+U} f_{\boldsymbol{\vartheta}}(\mathbf{x}_t) = c$, where c is a user-specified constant related to the portion of the number of the unlabeled data assigning to the positive class [5]. However, it again introduces another hyperparameter. Actually, our empirical evaluation finds that balance constraint is insensitive to the model performance. One reason may be that the $\mathcal{U}_0$ data has played the role of balance constrant in the model.

5. EXPERIMENTS

In this section, we evaluate our proposed 3C-SVM algorithm on both synthetic and real-world datasets and compare it with SVM, S^{3}VM [6], and $\mathcal{U}$-SVM [27]. Our 3C-SVM algorithm is implemented in Matlab 7.3 and the QP problem is solved by a general optimization toolbox, MOSEK[1]. In the experiments, we try to investigate the following questions: (1) What is the performance of 3C-SVM comparing with other three maximum-margin based algorithms? (2) What is the trade-off on the parameters D and ε on 3C-SVM? (3) What is the convergence of 3C-SVM?

Table 2: Description of data

Dataset	d	L	U	U_0
Synthetic	50	20, 50 200, 500	500	Two designed cases
USPS	256	10	100, 1000	Except "5" and "8"
MNIST	784	10	100, 1000	Except "5" and "8"

5.1 Synthetic Datasets

The synthetic data is a 50-dimensional dataset. The $\pm$1-class are generated following the scheme of [22], where the means are $c_i^\pm = \pm 0.3$ for $i = 1,\ldots,50$ and variance values are $\sigma^2_{1,2} = 0.08$ and $\sigma^2_{3,\ldots,50} = 10$. In this setting, we can generate two Gaussians with the Bayes risk being approximately 5%. Two kinds of $\mathcal{U}_0$ data similar to those in [22] are generated:

- For the first kind, it is a zero mean with $\sigma^2_{1,2} = 0.1$ and $\sigma^2_{3,\ldots,50} = 10$. It contains a zero mean, where the optimal Bayesian decision boundary passes through it, but it contains larger variances on the first two dimensions of the data than those of the $\pm$1-class data.

[1] `http://www.mosek.com`

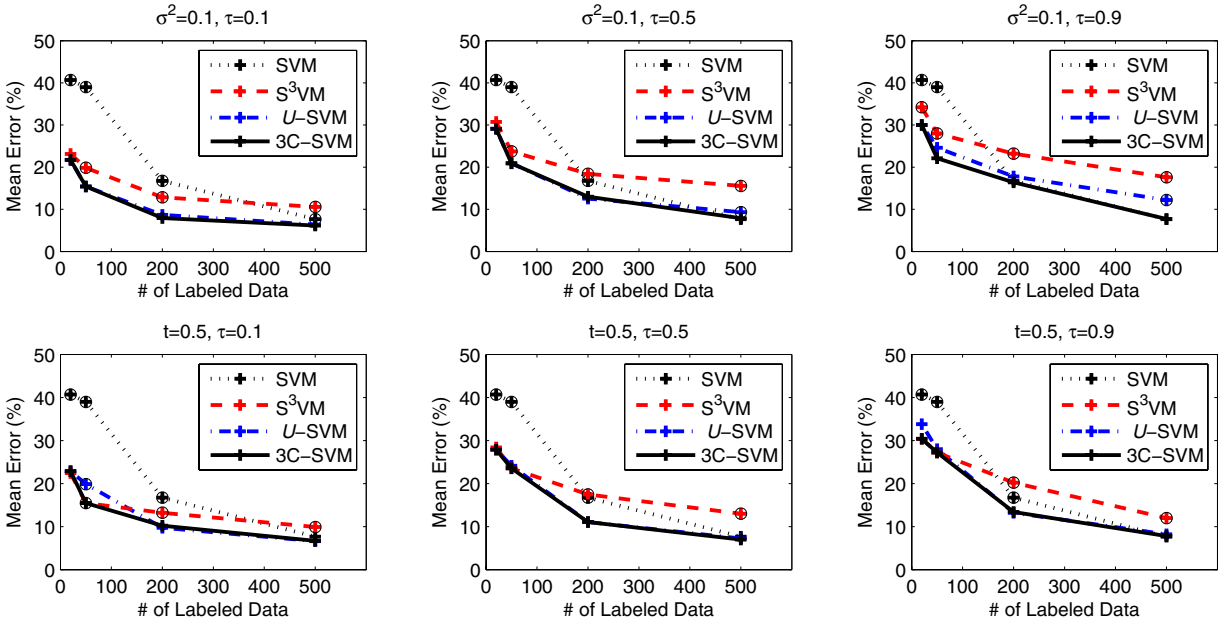


Figure 3: The performance of four algorithms on toy datasets with different combinations of mixed unlabeled data. The results of 3C-SVMs outperform the corresponding models with 95% significant level on paired t-test are marked by circles.

- For the second kind, the variance values are the same as the ± 1-class data, but the mean is $\frac{t}{2} \cdot (\mathbf{c}^+ - \mathbf{c}^-)$ $(t = 0.5)$, shifted a little bit from the origin, where the optimal Bayesian classifier passes through.

It is noted that the optimal decision boundary is a linear classifier for the synthetic datasets. Hence, we employ the linear kernel in fitting the data.

As reported in Table 2, we test the number of labeled data from $\{20, 50, 200, 500\}$ and vary the proportion of the mixed unlabeled data by $(\tau U, (1-\tau)U)$, where τU data are randomly chosen from ± 1-class and $(1-\tau)U$ data are randomly chosen from $\mathcal{U}_0$ data. τ is tested in $\{0.1, 0.5, 0.9\}$. We then evaluate the performance of the model on a separated test data with 500 data samples.

In the comparison, there are different parameters for different models need to be tuned. They include:

- the soft-margin hyperparameter C for SVM [24];
- the soft-margin hyperparameter, C, the trade-off constant for the loss of unlabeled data, $C_{\mathcal{U}}$, and the parameter for the ε-insensitive loss function, ε, in $\mathcal{U}$-SVM [22, 27];
- the soft-margin hyperparameter, C, the trade-off constant for the loss of unlabeled data $C_{\mathcal{U}}$, and the approximate parameter for ramp loss, κ, in S^{3}VM.

A problem of 3C-SVM is that its parameters are large and they will affect the model performance. These parameters in 3C-SVM not only include C in SVM, $C_{\mathcal{U}}$ in $\mathcal{U}$-SVM and S^{3}VM, ε in $\mathcal{U}$-SVM and κ in S^{3}VM, but also include the parameter D. It is terrible to tune all the parameters, e.g., by cross validation, together. To resolve this problem, we adopt a simple way to tune them. More specifically, we first tune the parameters in SVM, $\mathcal{U}$-SVM, and S^{3}VM on the test set, individually. Next, we set the parameters of our 3C-SVM based on the obtained optimal parameters from other models. That is, λ is set to $\frac{1}{C}$, $r_i = 1$ for labeled data and $r_i = \frac{C_{\mathcal{U}}}{C}$ for unlabeled data, where C and $C_{\mathcal{U}}$ are the optimal one corresponds to $\mathcal{U}$-SVM since this set of parameters obtains better performance than that of S^{3}VM. The parameters ε and κ are set the same as the optimal value from the $\mathcal{U}$-SVM and S^{3}VM, respectively. D is set to 2 for simplicity since the results have shown that our 3C-SVM can achieve very good performance.

Figure 3 reports the average performance (10 runs) of all four algorithms in the above all cases. It is shown that 3C-SVMs consistently attain the best results. $\mathcal{U}$-SVM achieves similar performance as 3C-SVM when the number of $\mathcal{U}_0$ data is large. However, its performance decreases when the number of $\mathcal{U}_0$ data decreases and cannot beat SVM when the size of the labeled training data is 500. Similar trend is obtained for S^{3}VM. On the contrary, our 3C-SVMs keep nearly the same accuracies and outperform $\mathcal{U}$-SVM and S^{3}VM when the number of labeled training data is large.

Convergence Study. We also study the convergence of 3C-SVM on the synthetic dataset with different settings ($L = 20/200$, $U = 500$). Figure 4 shows the one trial result on the objective function value and test errors at each CCCP iteration. The figures show that the 3C-SVM algorithm decreases the objective function value and the test error rates decrease correspondingly at each iteration. At the same time, 3C-SVM tends to converge in only a few iterations, less than 10.

5.2 Results on Real-World Handwritten Digit Datasets

The USPS dataset and the MNIST dataset are two popular benchmark handwritten digit datasets which have been used in literature to validate the corresponding classification models [6, 19]. As reported in Table 2, each image in USPS was normalized and centered with the size of 16×16, which forms 256-dimensional data (see examples in Figure 1(a)). This dataset contains 9,298 grayscale handwritten digit images, 7,291 of which are used as the training set, while the remaining 2,007 are used as the test set. The MNIST dataset consists of a training set of 60,000 digits and a test

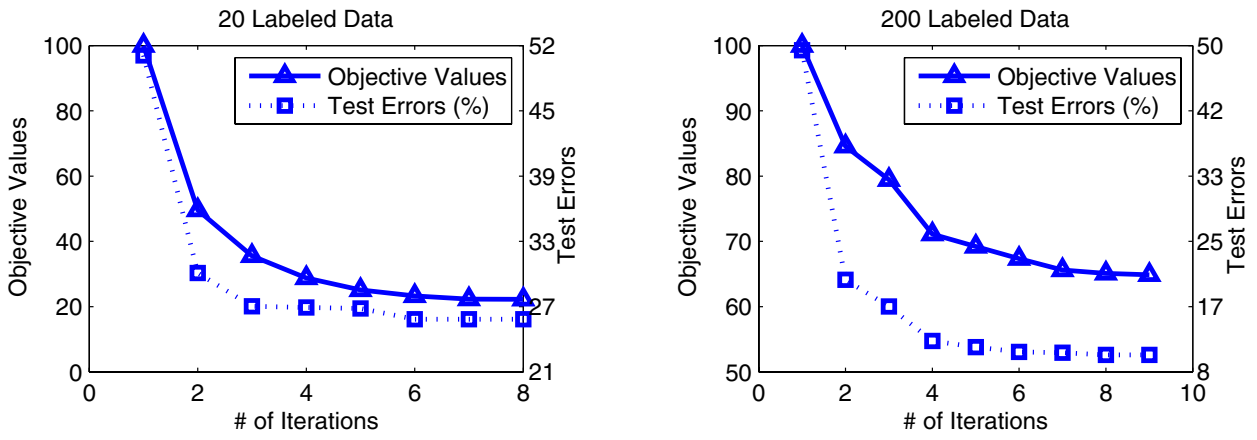


Figure 4: One trial result on the value of the objective function and test error during the CCCP iterations of training 3C-SVM on synthetic dataset with different number of labeled data. 3C-SVM converges only by a few iterations, less than 10.

set of 10,000 digits (see examples in Figure 1(b)). The digits are grayscale handwritten images normalized and centered in 28×28, which forms 784-dimensional data. We have normalized each pixel value in an image to the range of -1 and 1.

Similar to the setup in [22, 27], we employ digits "5" and "8" to construct the ± 1-class data, but differently, we utilize all other digits as 0-class. In the evaluation, we test the number of labeled data in 10 and the number of unlabeled data is 100 and 1000, while the proportion of the mixed unlabeled data is set as $(\tau U, (1-\tau)U)$, where where τU data are randomly chosen from digits "5" and "8" and $(1-\tau)U$ data are randomly chosen from other digits. τ is tested in $\{0.1, 0.5, 0.9\}$. The performance of the models is evaluated on the test set of digits "5" and "8".

Here, since the data are linearly nonseparable in the original feature space [19], we employ the RBF kernel on all the models. That is, $K(\mathbf{x}, \mathbf{y}) = \exp(-\gamma \|\mathbf{x} - \mathbf{y}\|^2)$, is adopted as the kernel, where γ is the width of the RBF kernel. Since we also need to tune an additional parameter, this makes it more difficult in tuning the parameters for 3C-SVM, and for other three models. Similar to the procedure in [5], we seek the optimal parameters on a separate validation set by maximizing the performance on the test set. More specifically, the parameters are sought as follows:

- For SVM, C is selected from $\{10^{-1}, 1, 10, 10^2, 10^3\}$. The width of the RBF kernel is set to $\gamma = \delta \times \frac{1}{d}$ as [19], where d is the number of data dimension, i.e., 256 for USPS and 784 for MNIST. δ is selected from $\{\frac{1}{32}, \frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1, 2, 4, 8\}$.
- For $\mathcal{U}$-SVM, C is tested in the same range of SVM and $C_{\mathcal{U}}$ is tested from $\{10^{-1}, 1, 10, 10^2\}$, where we find that when $C_{\mathcal{U}}$ is greater than 100, the model performance cannot be increased. The kernel parameter, γ, is tested as the same cases of SVM. ε is test in $\{0.01, 0.1, 0.2, 0.5, 0.8, 1\}$.
- For S^{3}VM, C, $C_{\mathcal{U}}$, and γ are tested the same as $\mathcal{U}$-SVM. κ is tested in $\{0.01, 0.1, 0.2, 0.5\}$.
- For 3C-SVM, we first fix ε as the optimal one obtained from $\mathcal{U}$-SVM and κ as the optimal one obtained from S^{3}VM. We then set $D = 2$ and test 3C-SVM by the optimal parameters from $\mathcal{U}$-SVM and by those from S^{3}VM. Our objective is to see which set of parameters can attain better performance. The empirical study indicates that adopting the parameters C, $C_{\mathcal{U}}$, and γ from $\mathcal{U}$-SVM, our 3C-SVM can achieve better results. Hence, we employ C, $C_{\mathcal{U}}$, and γ, which corresponds to the optimal ones from $\mathcal{U}$-SVM, in the experiment. In addition, we find that κ does not affect the result too much when it is around 0.1. So we do not specifically tune it in the experiment. After that, we tune ε in the same cases of $\mathcal{U}$-SVM and D in $\{1, 2, 10\}$ simultaneously.

Table 3 reports the average (10 runs) accuracies of four algorithms on the two handwritten digit datasets. 3C-SVM consistently attains better results in all cases. By examining the details of the results, we have the following observations:

- For SVM, since the model is optimized on the test datasets, it achieves satisfactory results. In practical, when we has limited number of labeled data, it is usually difficult to obtain good performance without the help of unlabeled data.
- For S^{3}VM, it is interesting to find that it is less sensitive to the proportion of the $\mathcal{U}_{\mathcal{L}}$ data and $\mathcal{U}_0$ data. For USPS, the performance of S^{3}VM does not change when the number of unlabeled data increases from 100 to 1000; while for MNIST, the performance of S^{3}VM is even worse in the case of $U = 1000$ than the case of $U = 100$. This indicates that the mixed unlabeled data actually hurts S^{3}VM.
- For $\mathcal{U}$-SVM, the performance decreases slightly as the number of $\mathcal{U}_0$ data decreases. This indicates that the $\mathcal{U}_0$ data actually plays the effect on helping $\mathcal{U}$-SVM. Similar to S^{3}VM on MNIST, $\mathcal{U}$-SVM also achieves worst performance when the number of unlabeled data increases in the USPS dataset. The decay may be due to the effect of the $\mathcal{U}_{\mathcal{L}}$ data.
- For 3C-SVM, it attains the best results and outperforms SVM for all cases, S^{3}VM for five cases on USPS and two cases on MNIST, and totally seven cases on $\mathcal{U}$-SVM for both datasets. It is observed that the performance of 3C-SVM also decreases slightly as the number of $\mathcal{U}_0$ data decreases. The reason lies that we employ the same regularization parameters of $\mathcal{U}$-SVM in the experiment.

5.3 Sensitivity Analysis

In the experiment, we also conduct sensitivity analysis on two parameters, ε and D, in 3C-SVM. The analysis of the hyperparameter, e.g., C, $C_{\mathcal{U}}$, γ, can be referred to [6, 19]. For κ, since it is insensitive when it is about 0.1, we do not study its effect in this section.

In the test, we change ε in $\{0, 0.01, 0.1, 0.2, 0.5, 0.8, 1.0\}$ and D in $\{1, 2, 10\}$ and test on the case of balance mixed unlabeled

Table 3: The average (10 runs) accuracies (%) of SVMs, S^3VMs, $\mathcal{U}$-SVMs, and 3C-SVMs on the USPS and the MNIST ("5" vs "8") datasets for different combinations of mixed unlabeled data. The p-values of paired t-test on 3C-SVMs against other algorithms are given in brackets. Significant improvement with 95% confidence level and the best accuracy are in bold.

Dataset	Setting	Algorithm	$\tau = 0.1$	$\tau = 0.5$	$\tau = 0.9$
USPS	$L = 10$ $U = 100$	SVM	87.1±1.7 (**0.5**)	87.1±1.7 (**1.2**)	87.1±1.7 (**0.8**)
		S^3VM	87.6±4.2 (**4.4**)	87.6±4.2 (5.1)	87.6±4.2 (**5.0**)
		$\mathcal{U}$-SVM	88.9±3.1 (17.1)	**88.8**±3.1 (25.1)	88.6±3.1 (10.2)
		3C-SVM	**89.2**±2.6	**88.8**±3.1	**88.8**±3.2
USPS	$L = 10$ $U = 1000$	SVM	87.1±1.7 (**0.4**)	87.1±1.7 (**0.9**)	87.1±1.7 (**0.9**)
		S^3VM	87.6±4.2 (**4.8**)	87.6±4.2 (**4.2**)	87.6±4.2 (**4.1**)
		$\mathcal{U}$-SVM	88.9±2.7 (6.7)	88.0±3.2 (**4.9**)	87.2±3.3 (**3.2**)
		3C-SVM	**89.3**±2.7	**89.1**±2.9	**89.1**±3.0
MNIST	$L = 10$ $U = 100$	SVM	71.0±9.3 (**0.5**)	71.0±9.3 (**0.8**)	71.0±9.3 (**0.9**)
		S^3VM	76.3±8.0 (17.8)	76.3±8.0 (22.1)	76.3±8.0 (30.2)
		$\mathcal{U}$-SVM	74.0±8.6 (**3.7**)	73.4±8.2 (**2.9**)	73.2±8.2 (**2.2**)
		3C-SVM	**76.8**±8.4	**76.7**±8.0	**76.3**±7.6
MNIST	$L = 10$ $U = 1000$	SVM	71.0±9.3 (**0.1**)	71.0±9.3 (**0.2**)	71.0±9.3 (**0.6**)
		S^3VM	75.9±7.9 (**4.8**)	75.9±7.9 (7.8)	75.9±7.9 (**0.6**)
		$\mathcal{U}$-SVM	74.0±8.4 (**1.7**)	73.9±8.1 (**1.5**)	73.3±7.9 (9.2)
		3C-SVM	**77.0**±7.4	**76.9**±7.9	**76.5**±8.0

data (i.e., $\tau = 0.5$). Figure 5 shows the changing trend with ε and D, respectively. It should be noted that the best results in Figure 5 also refer to the results in fifth column in Table 3. By examining these results, we have the following observations:

- 3C-SVM achieves the best performance on USPS when $\varepsilon = 1.0$, $D = 2$ and on MNIST when $\varepsilon = 0.5$, $D = 2$, respectively.
- For USPS dataset, when ε increases, the performance of 3C-SVM increases gradually and it achieves the best result when $\varepsilon = 1.0$, i.e., taking all unlabeled data as irrelevant data. For MNIST data, the performance increases gradually until $\varepsilon = 0.5$, then it decreases gradually.
- In terms of the effect of D, for both datasets, the performance of 3C-SVM increases gradually as D increases and decreases dramatically when $D = 10$. The best results are obtained when $D = 2$ for both datasets.

6. CONCLUSION

In this paper, we have proposed a novel maximum margin semi-supervised classifier, named the *tri-class support vector machine*, to learn from mixed unlabeled data. More specifically, we introduce a new min loss function to distinguish the mixed unlabeled data into relevant and irrelevant data based on which error occurred is smaller when assigning the data to the associated class. The min loss function can therefore achieve the maximum entropy principle and force the irrelevant data close to the decision boundary. In generalization, 3C-SVM includes several popular maximum margin classifiers, such as SVMs, S^3VMs, and $\mathcal{U}$-SVMs, as its special cases. Furthermore, we provide detailed theoretical analysis to show the role of irrelevant data and why the model works. Moreover, in implementation, we transform 3C-SVM from an integer programming problem to a sequence of QP problems. The approximation by the concave-convex procedure has speeded up the model largely and finally yielded the same worst case time complexity as that of S^3VMs.

This work opens several interesting research issues. One worthy direction is to further speed up the model by warm-starting or by exploiting the sparsity structure of the solution. The other direction is to design an efficient way to tune the model parameters or to design a scheme to automatically learn the model parameters. We will also consider to extend the model to solve multi-class classification tasks and verify its performance.

Acknowledgment

This work is supported by two grants from the Research Grants Council of the Hong Kong SAR, China (Project No. CUHK 413210 and Project No. CUHK 415410) and a grant supported by a research funding from Google Focused Grant Project "Mobile 2014".

7. REFERENCES

[1] R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, 2005.

[2] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

[3] K. P. Bennett and A. Demiriz. Semi-supervised support vector machines. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1831–1850, 1998.

[4] T. D. Bie and N. Cristianini. Convex methods for transduction. In *Advances in Neural Information Processing Systems 16*, pages 73–80. MIT Press, 2003.

[5] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

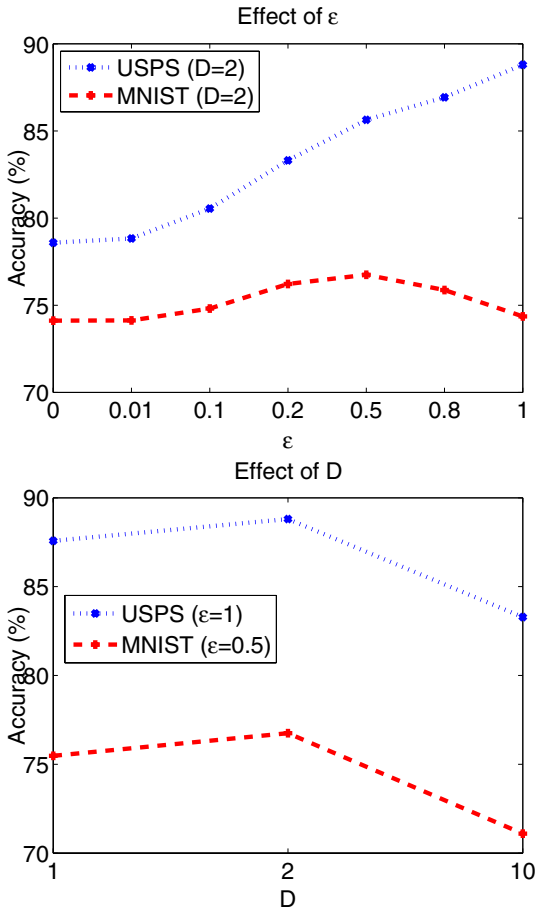


Figure 5: Effect on ε and D for two handwritten digit datasets. The best results of 3C-SVM are obtained when $\varepsilon = 1.0$, $D = 2$ for USPS and when $\varepsilon = 0.5$, $D = 2$ for MNIST, respectively.

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

[7] R. Collobert, F. H. Sinz, J. Weston, and L. Bottou. Trading convexity for scalability. In *ICML*, pages 201–208, 2006.

[8] D. Goldfarb and S. Liu. An o(n^3l) primal interior point algorithm for convex quadratic programming. *Math. Program.*, 49(3):325–340, 1991.

[9] D. W. Hosmer and S. Lemeshow. *Applied logistic regression*. Wiley-Interscience Publication, 2nd edition, 2000.

[10] K. Huang, Z. Xu, I. King, and M. R. Lyu. Semi-supervised learning from general unlabeled data. In *the IEEE International Conference on Data Mining, ICDM 2008*, pages 273–282, 2008.

[11] T. Joachims. Transductive inference for text classification using support vector machines. In *International Conference on Machine Learning (ICML)*, pages 200–209, Bled, Slowenien, 1999.

[12] N. D. Lawrence and M. I. Jordan. Semi-supervised learning via gaussian processes. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 753–760, Cambridge, MA, 2005. MIT Press.

[13] Y.-F. Li, J. T. Kwok, and Z.-H. Zhou. Cost-sensitive semi-supervised support vector machine. In *AAAI*, pages 500–505, 2010.

[14] Y.-F. Li and Z.-H. Zhou. S4vm: Safe semi-supervised support vector machine, 2010. arXiv:1005.1545.

[15] M. Mojdeh and G. V. Cormack. Semi-supervised spam filtering: does it work? In *SIGIR*, pages 745–746, 2008.

[16] K. Nigam, A. McCallum, S. Thrun, and T. M. Mitchell. Text classification from labeled and unlabeled documents using em. *Machine Learning*, 39(2/3):103–134, 2000.

[17] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359, 2010.

[18] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 1999.

[19] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

[20] B. Settles. Active learning literature survey. Technical Report 1648, Computer Sciences, University of Wisconsin-Madison, 2010.

[21] A. Singh, R. D. Nowak, and X. Zhu. Unlabeled data: Now it helps, now it doesn't. In *NIPS*, pages 1513–1520, 2008.

[22] F. H. Sinz, O. Chapelle, A. Agarwal, and B. Schölkopf. An analysis of inference with the universum. In *NIPS*, pages 1369–1376, 2008.

[23] A. J. Smola, S. V. N. Vishwanathan, and T. Hofmann. Kernel methods for missing variables. In *Proceedings of the tenth International Workshop on Artificial Intelligence and Statistics*, 2005.

[24] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 2nd edition, 1999.

[25] V. Vapnik and S. Kotz. *Estimation of Dependences Based on Empirical Data: Empirical Inference Science (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2nd edition, 2006.

[26] J. Wang, X. Shen, and W. Pan. On efficient large margin semisupervised learning: Method and theory. *Journal of the Royal Statistical Society, Series B*, 10(Mar):719–742, 2009.

[27] J. Weston, R. Collobert, F. H. Sinz, L. Bottou, and V. Vapnik. Inference with the universum. In *ICML*, pages 1009–1016, 2006.

[28] L. A. Wolsey. *Integer programming*. Wiley-Interscience, 1 edition, September 1998.

[29] A. L. Yuille and A. Rangarajan. The concave-convex procedure. *Neural Computation*, 15(4):915–936, 2003.

[30] D. Zhang, J. Wang, F. Wang, and C. Zhang. Semi-supervised classification with universum. In *SDM*, pages 323–333, 2008.

[31] Z.-H. Zhou and M. Li. Semi-supervised learning by disagreement. *Knowl. Inf. Syst.*, 24(3):415–439, 2010.

[32] Z.-H. Zhou, D.-C. Zhan, and Q. Yang. Semi-supervised learning with very few labeled training examples. In *AAAI*, pages 675–680, 2007.

[33] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Department of Computer Sciences, University of Wisconsin, Madison, 2005.

[34] X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, pages 912–919, 2003.

[35] X. Zhu and A. B. Goldberg. *Introduction to Semi-Supervised Learning*. Morgan & Claypool, 2009.