Neural Networks 70 (2015) 90–102

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Maximum margin semi-supervised learning with irrelevant data

Haiqin Yang [a,b,*], Kaizhu Huang [c,**], Irwin King [a,b], Michael R. Lyu [a,b]

[a] Shenzhen Key Laboratory of Rich Media Big Data Analytics and Application, Shenzhen Research Institute, The Chinese University of Hong Kong, Hong Kong

[b] Computer Science & Engineering, The Chinese University of Hong Kong, Hong Kong

[c] Xi'an Jiaotong-Liverpool University, Suzhou, Jiangsu, China

ARTICLE INFO

Article history:
Received 12 November 2014
Received in revised form 4 May 2015
Accepted 30 June 2015
Available online 15 July 2015

Keywords:
Maximum margin classifier
Irrelevant data
Semi-supervised learning
Concave–convex procedure

ABSTRACT

Semi-supervised learning (SSL) is a typical learning paradigms training a model from both labeled and unlabeled data. The traditional SSL models usually assume unlabeled data are relevant to the labeled data, i.e., following the same distributions of the targeted labeled data. In this paper, we address a different, yet formidable scenario in semi-supervised classification, where the unlabeled data may contain irrelevant data to the labeled data. To tackle this problem, we develop a maximum margin model, named *tri-class support vector machine* (3C-SVM), to utilize the available training data, while seeking a hyperplane for separating the targeted data well. Our 3C-SVM exhibits several characteristics and advantages. First, it does not need any prior knowledge and explicit assumption on the data relatedness. On the contrary, it can relieve the effect of irrelevant unlabeled data based on the logistic principle and maximum entropy principle. That is, 3C-SVM approaches an ideal classifier. This classifier relies heavily on labeled data and is confident on the relevant data lying far away from the decision hyperplane, while maximally ignoring the irrelevant data, which are hardly distinguished. Second, theoretical analysis is provided to prove that in what condition, the irrelevant data can help to seek the hyperplane. Third, 3C-SVM is a generalized model that unifies several popular maximum margin models, including standard SVMs, Semi-supervised SVMs (S^3VMs), and SVMs learned from the universum ($\mathcal{U}$-SVMs) as its special cases. More importantly, we deploy a concave–convex produce to solve the proposed 3C-SVM, transforming the original mixed integer programming, to a semi-definite programming relaxation, and finally to a sequence of quadratic programming subproblems, which yields the same worst case time complexity as that of S^3VMs. Finally, we demonstrate the effectiveness and efficiency of our proposed 3C-SVM through systematical experimental comparisons.

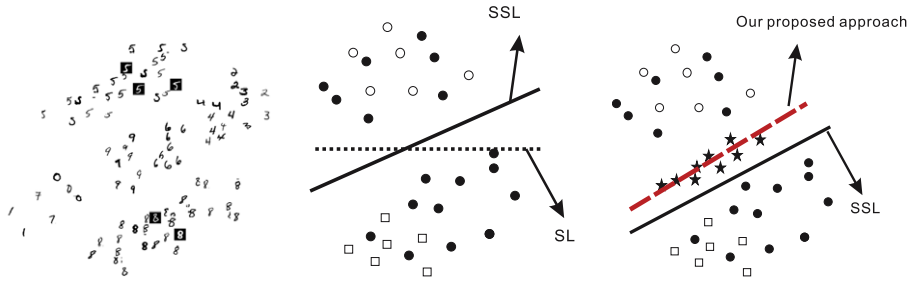
1. Introduction

Supervised learning is an effective tool to learn from a set of labeled data in solving classification and regression problems, which widely occurred in various application domains such as handwriting recognition (Schölkopf & Smola, 2002) and bioinformatics (Hastie, Tibshirani, & Friedman, 2009; Vapnik, 1999; Yang, King, & Lyu, 2011). However, supervised learning methods usually need a sufficiently large number of labeled samples in the training procedure to learn good decision functions. Essentially, labeling data is an expensive and time consuming task due to the request of experts' knowledge. To tackle the problem of insufficient amount of labeled training samples, researchers have proposed various methods in the literature. They include

- Active learning: A learning paradigm requires users' (or some other information source) interaction to provide the responses of new data points (Schohn & Cohn, 2000; Settles, 2010).
- Transfer learning: These methods focus on applying the knowledge learned from related, but different tasks to solve the target task (Pan & Yang, 2010; Yang, King, & Lyu, 2010; Yang, Lyu, & King, 2013). They usually require sufficient labeled data to learn accurate knowledge.
- Semi-supervised learning (SSL) or Transductive learning: These techniques aim at learning an inductive rule or try to accurately determine the label of the data from a small amount of labeled data with the help of a large amount of unlabeled data (Zhou & Li, 2010; Zhu & Goldberg, 2009).

* Corresponding author. Tel.: +852 31634251.
** Corresponding author. Tel.: +86 0512 8816 1404.
E-mail addresses: hqyang@ieee.org (H. Yang), Kaizhu.Huang@xjtlu.edu.cn (K. Huang).

http://dx.doi.org/10.1016/j.neunet.2015.06.004
0893-6080/



(a) MNIST. (b) SL vs. SSL. (c) SSL vs. 3C-SVM.

Fig. 1. The left figure illustrates the task of classifying digits "5" and "8" with mixed unlabeled digits: block digits are labeled data on the target binary classification task, while black digits are mixed unlabeled digits, including other irrelevant digits from "0" to "9". Our proposed 3C-SVM utilizes the "irrelevant" data (⋆'s) to seek a more meaningful decision function, while the S^3VM is misled by the irrelevant data.

In the above learning paradigms, active learning requires users' additional interaction and transfer learning techniques need sufficient labeled data to learn necessary knowledge. On the contrary, SSL needs the least training samples. Therefore, we consider semi-supervised learning paradigm in this paper.

Recently, various SSL methods have been proposed in the literature; see Chapelle, Schölkopf, and Zien (2006) and Zhu and Goldberg (2009) and references therein. However, previously proposed SSL methods mainly assume unlabeled data are "clean". Obviously, in real-world applications, without carefully preprocessing, unlabeled data are easily incorporated with other irrelevant data. That is, the unlabeled data may follow different distributions of the target labeled data. For example, when crawling unlabeled web pages to classify two categories, say "finance" and "sports", it is easy to collect some irrelevant web pages and include them as unlabeled data. The same scenario occurs in classifying handwritten digits, see Fig. 1(a) as an example of classifying the digits "5" and "8" with irrelevant data. Hence, the unlabeled data may consist of relevant and irrelevant data. Learning from the labeled data and the mixed unlabeled data may hurt the training of classification models. Fig. 1 gives a motivating example on S^3VM can learn a good classifier when the unlabeled data is clean, while being misled when the unlabeled data is mixed.

To tackle this difficult scenario, we propose a novel maximum margin semi-supervised model, named *tri-class support vector machine* (3C-SVM) to utilize all available data. We highlight the main contributions of this paper as follows:

1. First, we propose a novel 3C-SVM model to solve a very difficult scenario in semi-supervised learning with mixed unlabeled data. One main characteristic of 3C-SVM is that it generalizes several popular maximum margin models, including standard SVMs, Semi-supervised SVMs (S^3VMs), and SVMs learned from universum data ($\mathcal{U}$-SVMs). This paper summarizes our previously proposed two solutions in Huang, Xu, King, and Lyu (2008) and Yang, Zhu, King, and Lyu (2011).
2. Second, we not only provide the intuition of the model formulation, but also perform theoretical analysis on 3C-SVM, which shows why the irrelevant data can help the model. Based on logistic principle and the maximum entropy principle, we can rely more on the labeled and relevant data, while automatically ignoring the irrelevant data.
3. Third, we observe that the original formulation of 3C-SVM is a mixed integer programming problem. We derive the semi-definite programming relaxation when considering it as transductive learning. Furthermore and more importantly, we view the formulation as semi-supervised learning and deploy the concave–convex procedure (CCCP) (Yuille & Rangarajan, 2003) to efficiently seek the inductive rule. This yields solving a finite number of quadratic programming (QP) subproblems and achieves the same worst case time complexity as that of S^3VMs (Collobert, Sinz, Weston, & Bottou, 2006). The speedup is very competitive in terms of efficiency in many semi-supervised learning models.
4. Finally, we demonstrate the effectiveness and efficiency of 3C-SVM through a series of empirical evaluation on both synthetic and real-world datasets. Sensitivity analysis is also provided to exhibit the characteristics of 3C-SVM.

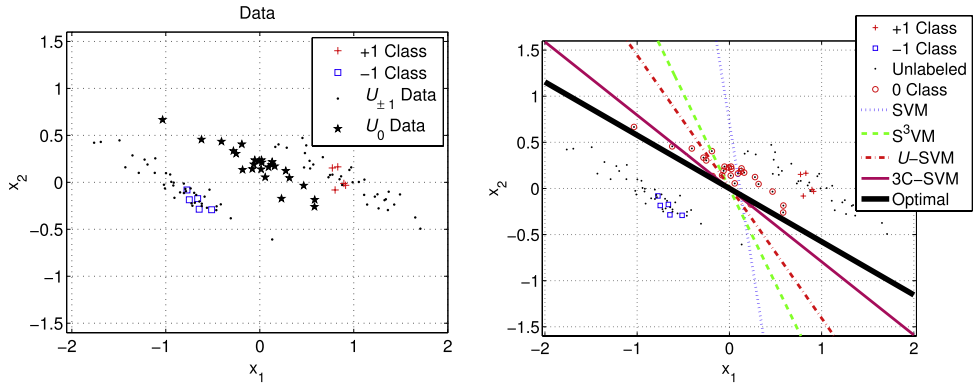
The rest of the paper is organized as follows. Section 2 reviews several typical related work, which motivates the proposed 3C-SVM. Section 3 presents the formulation of 3C-SVM and its properties. Section 4 details the solving procedure of 3C-SVM algorithm. Section 5 reports the experimental comparisons and results. Finally, Section 6 concludes the whole paper.

2. Related work

In the following, we review related methods that learn a binary classifier from both labeled and unlabeled data, or labeled targeted data with other auxiliary data.

Semi-supervised learning (SSL) is a learning paradigm that learns from both labeled and unlabeled data (Chapelle et al., 2006; Zhu & Goldberg, 2009). Typical SSL methods include generative methods for SSL (Lawrence & Jordan, 2005; Nigam, McCallum, Thrun, & Mitchell, 2000), graph-based SSL methods (Belkin, Niyogi, & Sindhwani, 2006; Huang, Song, Gupta, & Wu, 2014; Iosifidis, Tefas, & Pitas, 2014; Melacci & Belkin, 2011), maximum margin classifiers (Chapelle et al., 2006; Collobert et al., 2006; Joachims, 1999), etc. Usually, these models work when the number of label data is small and the number of unlabeled data is sufficiently large. Typically, the given unlabeled data are assumed following the same distribution as the labeled data (Chapelle et al., 2006; Dehdarbehbahani, Shakery, & Faili, 2014; Zhao, Zhang, Chow, & Li, 2014; Zhu & Goldberg, 2009). Hence, they utilize the unlabeled data to find the data distribution so as to improve the model performance. However, when unlabeled data are mixed with irrelevant data, where the data follow distributions different that of the target task, they usually do harm to the SSL models (Singh, Nowak, & Zhu, 2008). A motivating example is shown in Fig. 2(b) and an illustration is shown as the dash line in Fig. 2.

$\mathcal{U}$-SVM (Weston, Collobert, Sinz, Bottou, & Vapnik, 2006) is a special learning paradigm that learns from both labeled data and universum data. The universum data is a third kind of data yielding the most contradiction through the final hyperplane and obviously, the distribution of the universum data is different from neither the positive class nor the negative class in the target task. Since the universum data play the role of seeking the subspace for the decision function (Sinz, Chapelle, Agarwal, & Schölkopf, 2008), they can help to improve the model performance when the data are carefully chosen. However, the label of these data has to be explicitly specified in the training procedure. In real-world



(a) Data. (b) Classifiers illustration.

Fig. 2. Illustration of data and different classifiers on R^2. Data denoted by +'s and □'s are positive and negative data, respectively. Data denoted by ·'s come from the targeted binary classification task and those denoted by ⋆'s are irrelevant data, whose labels are unknown before the training. Fig. 2(b) shows that the 3C-SVM (the thin solid line) achieves the best result, which is closest to the Bayesian optimal classifier (the thick solid line), among all maximum margin based classifiers and automatically distinguishes the irrelevant unlabeled data (black dots with red circles) well. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Key notations used in this paper.

Notations	Description
$\mathbf{w}, \mathbf{K}$	Bold small and capital letters denote vectors and matrices, respectively.
$\mathcal{X}, \mathbb{R}$	Letters in calligraphic or blackboard bold fonts denote sets.
$\mathcal{L} = \{(\mathbf{x}_i,\ y_i)\}_{i=1}^{L}$	The set of labeled data consists of L labeled samples, where $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^d$, and the label is triple, $y_i \in \{-1,\ 0,\ 1\}$.
$\mathcal{L} = \mathcal{L}_{\pm 1} \cup \mathcal{L}_0$	$\mathcal{L}_{\pm 1}$ consists of the positive and negative data with labels being $+1$ or -1, while data in $\mathcal{L}_0$ follows distributions different from those in $\mathcal{L}_{\pm 1}$.
$\mathcal{U} = \{\mathbf{x}_j\}_{j=L+1}^{L+U}$	The set of unlabeled data consists of U samples, where $\mathbf{x}_j \in \mathbb{R}^d$.
$\mathcal{U} = \mathcal{U}_{\pm 1} \cup \mathcal{U}_0$	Data in $\mathcal{U}_{\pm 1}$ follows the same distribution of $\mathcal{L}_{\pm 1}$. Data in $\mathcal{U}_0$ follows the same distribution of $\mathcal{L}_0$. However, their labels are unknown in the training period.
$f_{\vartheta}(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b$	A hyperplane parameterized by $\vartheta = (\mathbf{w},\ b)$, where $\phi : \mathbb{R}^d \to \mathbb{R}^f$, is a feature mapping function often implicitly defined by a Mercer kernel.
kL, kLU	Simplified expressions: $kL \triangleq k + L$ and $kLU \triangleq k + L + U$.
$\tilde{1}_{d_k}$, and $\tilde{\varepsilon}_{d_k}$	Simplified expressions: $\tilde{1}_{d_k} \triangleq 1 - D(1 - d_k)$ and $\tilde{\varepsilon}_{d_k} \triangleq -\varepsilon - Dd_k$.
$\mathbf{0}_{U,L}$	A $U \times L$ matrix with all elements being 0.
$\mathbf{I}_U$	A $U \times U$ identity matrix.
$\mathbf{x}_{-i}, y_{-i}$	The index $-i$ indicates it shifting i advance from the 0-index.
$\mathbf{\Lambda}_{\mathcal{R},\mathcal{C}}$	$\Lambda_{ij} = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$: $\mathcal{R}$ and $\mathcal{C}$ indicate the corresponding row and column ranges of data indices, respectively. $\mathcal{R} = 2U$ denotes the row index ranges from $L + 1$ to $L + 2U$. $\mathcal{R} = \lvert\mathcal{L}_0\rvert + L + 2U$ denotes the row index ranges from $-\lvert\mathcal{L}_0\rvert$ to $L + 2U$ except 0.

applications, without prior knowledge, we cannot guarantee that the unlabeled data are "clean", or belong to the universum set. In this case, the data may be mixed with the universum data and the relevant data, where the relevant data may hurt the $\mathcal{U}$-SVMs; see the dash–dot line learned from mixed universum data in Fig. 2(b) as an example.

Some other models also consider the case of mixed auxiliary data, which is partially similar to the scenario we take into account in this paper. For example, Zhang, Wang, Wang, and Zhang (2008) proposed a graph-based semi-supervised learning model to learn from both labeled and unlabeled data. In the model, the unlabeled data are assumed following the same distribution of the targeted binary classification task, which is the same as previously proposed SSL models. A different assumption is that universum data are included in the labeled data and they need to explicitly indicate the label of the universum data. Li and Zhou (2010) proposed the safe semi-supervised support vector machine method to alleviate the effect of the noise in the unlabeled data. However, this method does not consider the case of mixture unlabeled data and needs to postprocess the results through two separate steps.

In short, previously proposed methods cannot address well the scenario of learning from mixed unlabeled data, which consists of relevant and irrelevant data. However, without good preparation, the unlabeled data are easily mixed with irrelevant data. Hence, we target at this formidable task to learn an inductive rule or a hyperplane with the help of mixed unlabeled data.

3. Tri-class support vector machine (3C-SVM)

In this section, we first present the notations and the problem definition in this paper. Next, we formulate the problem and propose the tri-class support vector machine, namely 3C-SVM. After that, we present the important properties of 3C-SVM.

3.1. Notations and problem definition

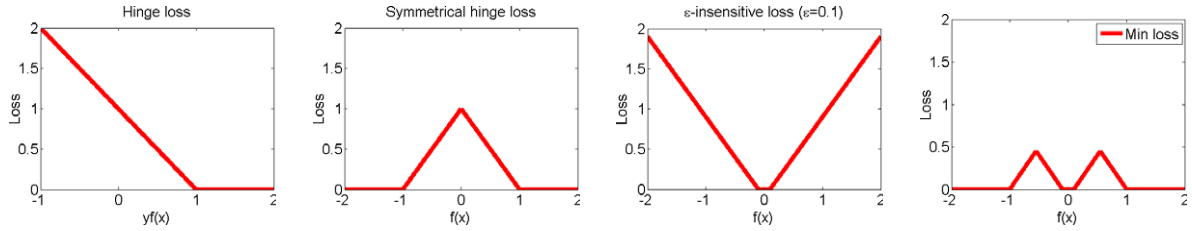
To make the notations consistent in the whole paper, we first present some important notations and describe their meaning in Table 1. Now, given labeled data $\mathcal{L}$ and unlabeled data $\mathcal{U}$ defined in Table 1, where the number of unlabeled data is much larger than that of the labeled target data, i.e., $|\mathcal{L}_{\pm 1}| \ll U$, our goal is to seek a hyperplane that classifies the ± 1 data well with the help of all available data. The decision hyperplane is defined as follows:

$$f_{\vartheta}(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b. \tag{1}$$

To attain this goal, we formulate the objective via the following maximum margin criterion:

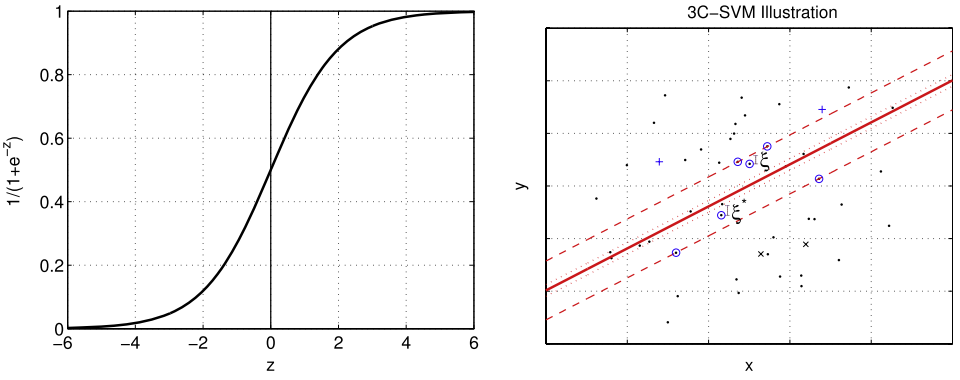
$$\min_{\vartheta} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \sum_{\mathbf{x}_i \in \mathcal{L}} r_i \ell_L(f_{\vartheta}(\mathbf{x}_i), y_i) + \sum_{\mathbf{x}_i \in \mathcal{U}} r_i \ell_U(f_{\vartheta}(\mathbf{x}_i)), \tag{2}$$

where minimizing $\|\mathbf{w}\|^2$ is equivalent to maximizing the margin width, which can also control the capacity of the function space



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

Fig. 3. Illustration of different loss functions, including hinge loss, symmetrical hinge loss, ε-insensitive loss, and our proposed min loss.



(a) Sigmoid function. (b) Errors.

Fig. 4. Illustration of a sigmoid function and the error measure by the min loss. In Fig. 4(b), points with circles are *support vectors*.

(Vapnik, 1999). The constant $\lambda > 0$ trades off the regularization term and the empirical risks. $\ell_L(\cdot,\cdot)$ defines the empirical risk of the labeled data and $\ell_U(\cdot)$ measures the empirical risk of the unlabeled data. $r_i > 0, i = 1,\ldots,L+U$, is the corresponding balance penalty ratio.

For different problems, we have to choose different loss functions to measure the empirical risk on the data. Typical loss functions include:

- **Hinge loss**: $H_1(u) = \max\{0, 1-u\}$. This loss function has been used to measure the empirical risk of labeled data in standard SVMs (Vapnik, 1999); see Fig. 3(a) as an illustration.
- **Symmetric hinge loss**: $\tilde{H}_s(u) = H_1(|u|)$. This loss function has been applied to measure the empirical risk on unlabeled data for S^{3}VMs (Collobert et al., 2006); see Fig. 3(b) as an illustration.
- ε-**insensitive loss**: $I_\varepsilon(u) = \max\{0, |u| - \varepsilon\}$. This loss function has been adopted to measure the empirical risk in Support Vector Regression (Vapnik, 1999) and the Universum data in $\mathcal{U}$-SVMs (Weston et al., 2006); see Fig. 3(c) as an illustration.

In Eq. (2), a difficult issue is how to measure the empirical risk on the unlabeled data, i.e., how to calculate the third term in Eq. (2). This is because without good preparation, the unlabeled data are possible to be mixed with data relevant or irrelevant to the target task. Without prior knowledge, it is very tough to differentiate them into the relevant and the irrelevant data.

Here, our intuition on tackling the unlabeled data is based on the following two principles:

1. **Logistic principle**: Data points lying farther away from the decision hyperplane are more likely to be classified as data from ±1-class. However, when data points lie near the decision hyperplane, they are difficult to be classified correctly. Hence, ideally, data from ±1-class should lie on or outside of the margin gap; while other the irrelevant data are close to the decision hyperplane.
2. **Maximum entropy principle**: An ideal classifier should believe in all labeled data, while relying on relevant data and maximally ignoring the irrelevant data. Since the labeled data are few and collected by experts' effort, we should take serious consideration on them. Relying on more confident data and ignoring uncertain data can achieve maximum entropy.

Hence, the above two principles imply that irrelevant data should lie around the sought decision hyperplane. In order to achieve the above principles, we design a min loss function to measure the risk on the mixed unlabeled 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 Fig. 3(d) as the definition and Fig. 4(b) as an illustration):

$$\ell_{\min}(\mathbf{x}) = \min\left\{H_1(|f_\vartheta(\mathbf{x}_i)|), I_\varepsilon(|f_\vartheta(\mathbf{x}_i)|)\right\}. \tag{3}$$

Hence, for an unlabeled data point, when its error is determined by the ε-insensitive loss, it is deemed as irrelevant data; otherwise, when its error is decided by the symmetric hinge loss, we can set it as relevant data.

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

$$\min_\vartheta \frac{\lambda}{2}\|\mathbf{w}\|^2 + \sum_{\mathbf{x}_i\in\mathcal{L}_{\pm1}} r_i H_1(y_i f_\vartheta(\mathbf{x}_i)) + \sum_{\mathbf{x}_i\in\mathcal{L}_0} r_i I_\varepsilon(f_\vartheta(\mathbf{x}_i)) + \sum_{\mathbf{x}_i\in\mathcal{U}} r_i \min\left\{H_1(|f_\vartheta(\mathbf{x}_i)|), I_\varepsilon(|f_\vartheta(\mathbf{x}_i)|)\right\}. \tag{4}$$

In the above, the first two terms correspond to the formulation of a standard SVM (Vapnik, 1999). The third term measures the empirical risk of $\mathcal{L}_0$ data, the same as that in $\mathcal{U}$-SVMs (Weston et al., 2006). The last term measures the loss of unlabeled data. Hence, when the decision function is learned, we can determine

Table 2
Relation between different models and the usage of training data.

3C-SVM				SVM				S^3VM				$\mathcal{U}$-SVM			
$\mathcal{L}$	−1	0	1	$\mathcal{L}$	−1		1	$\mathcal{L}$	−1		1	$\mathcal{L}$	−1	0	1
$\mathcal{U}$	−1	0	1	$\mathcal{U}$				$\mathcal{U}$	−1		1	$\mathcal{U}$			

the class label of a data point $\mathbf{x}$ by the following criterion:

$$c(\mathbf{x}) = \begin{cases} +1 & \text{if } f_{\vartheta}(\mathbf{x}) > \dfrac{1+\varepsilon}{2} \\ -1 & \text{if } f_{\vartheta}(\mathbf{x}) < -\dfrac{1+\varepsilon}{2} \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

The above criterion separates the data into three classes, where the 0-class data corresponds to the irrelevant data.

3.2. Properties of 3C-SVMs

In the following, we present two favorite properties of our 3C-SVM. The first one is the generalization property. The second one is an insightful theorem on why the irrelevant data help the model.

First, as summarized in Table 2, our 3C-SVM framework generalizes several popular maximum margin models:

1. 3C-SVM includes a standard SVM formulation (Vapnik, 1999) as its special case. By setting r_i to zero in the third and fourth terms of Eq. (4) and only labeled data are given in the training set, we can reduce the formulation to a standard SVM problem.
2. An S^3VM formulation (Chapelle et al., 2006) 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 Eq. (4). 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 (Weston et al., 2006). It can easily be obtained by setting r_i to zero in the fourth term of Eq. (4). This formulation works when only labeled data and universum data are given.

Since our 3C-SVM enforces irrelevant data to fall close to the decision function, it appears that the model may not be suitable when data do not follow such scenarios. However, this problem can be resolved by selecting a suitable subspace through the kernel trick (Chen, Yang, King, & Lyu, 2015; Hu, Yang, King, Lyu, & So, 2015; Schölkopf & Smola, 2002; Yang, Xu, Ye, King, & Lyu, 2011). The following theorem provides an insightful result why the irrelevant data can help and how they help.

Theorem 1. *A 3C-SVM with $r_i = \infty$ for unlabeled data and $\varepsilon = 0$ is equivalent to separating the unlabeled data into two sets, $\mathcal{U}_{\pm\infty}$ and $\mathcal{U}_0$, where data in $\mathcal{U}_{\pm}$ fall on or outside of the margin gap and data in $\mathcal{U}_{\mathscr{A}}$ lie in the decision hyperplane, and corresponding to one of the following two cases:* (1) *When $|\mathcal{U}_0| \geq 2$, and it corresponds to training a general S^3VM on the training data projected onto the orthogonal complement of the span $\{\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j), \mathbf{x}_i, \mathbf{x}_j \in \mathcal{U}_0\}$; or* (2) *When $|\mathcal{U}_0| = 0$, or 1, it corresponds to training a general S^3VM, which guarantees at most one unlabeled data falling on the decision hyperplane, while keeping all other unlabeled data falling on or out of the margin gap.*

Proof. If we set $r_i = \infty$ for $\mathcal{U}$ data, then the min term, or the fourth term in Eq. (4) will vanish. By considering $\varepsilon = 0$, the optimal solutions of $\mathbf{w}$ and b will satisfy one of the following conditions:

$$|\mathbf{w}^\top \phi(\mathbf{x}_j) + b| \geq 1, \text{ or} \tag{6}$$

$$\mathbf{w}^\top \phi(\mathbf{x}_j) + b = 0. \tag{7}$$

The above two conditions indicate that the unlabeled data consist of two sets of data, $\mathcal{U}_{\pm}$ and $\mathcal{U}_0$. That is, those unlabeled data satisfying the condition of (6). Geometrically, they fall on or outside of the margin gap, are relevant data, denoted by $\mathcal{U}_{\pm}$; while those unlabeled data satisfying the condition of (7). Geometrically, they fall on the decision hyperplane, are irrelevant data, denoted by $\mathcal{U}_{\mathscr{A}}$.

It is noted that the condition of (7) can determine the subspace of the decision function when the number of data in $\mathcal{U}_0$ is greater than one. Hence, when $|\mathcal{U}_0| \geq 2$, we can select any two points $\mathbf{x}_i$ and $\mathbf{x}_j$ from $\mathcal{U}_0$. Subtracting the decision function values of these two points, we have $\mathbf{w}^\top(\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)) = 0$. Hence, we conclude that $\mathbf{w}$ is orthogonal to the span $\{\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j), \mathbf{x}_i, \mathbf{x}_j \in \mathcal{U}_0\}$. To see this, we define $P_{\mathcal{U}_0^\perp}$ as an orthogonal project on the orthogonal complement of the mapped set $\mathcal{U}_0$. Hence, we have $P_{\mathcal{U}_0^\perp}^\top = P_{\mathcal{U}_0^\perp}$. Further, we know the optimal $\mathbf{w}$ satisfies $\mathbf{w} = P_{\mathcal{U}_0^\perp}\mathbf{w}$ and $\mathbf{w}^\top\phi(\mathbf{x}_i) = \mathbf{w}^\top P_{\mathcal{U}_0^\perp}^\top \phi(\mathbf{x}_i) = \mathbf{w}^\top P_{\mathcal{U}_0^\perp}\phi(\mathbf{x}_i)$. This implies that the optimization problem in Eq. (4) is the same as an S^3VM on all labeled data and data from $\mathcal{U}_{\pm 1}$ transformed by $P_{\mathcal{U}_0^\perp}\phi(\mathbf{x}_i)$. This completes the proof of first case.

Second, if $|\mathcal{U}_0| = 0$, or 1, it leads to the result of case (2) in the above theorem. Here, a general S^3VM means that it is a generalization of the S^3VM and the $\mathcal{U}$-SVM. ■

The optimal 3C-SVM is to find the most suitable subspace to maximize the margin, while minimizing the overall empirical risk. Theorem 1 clearly shows that the irrelevant data can help 3C-SVM to find the subspace.

4. Solution and computation

In the following, we focus on how to solve 3C-SVM efficiently. A very difficult issue is how to tackle the min term.

4.1. Elimination of min terms

To remove the min term, we follow the idea of L_1-norm S^3VM (Bennett & Demiriz, 1998) and introduce decision variables, $d_k \in \{0, 1\}, k = 1, \ldots, U$, to separate the min term. That is, we obtain

$$\begin{aligned} &\ell_{\min}(\mathbf{x}) = Q_1 + Q_2, \\ &Q_1 := H_1(|f_{\vartheta}(\mathbf{x}_i)| + D(1 - d_k)), \\ &Q_2 := I_{\varepsilon}(|f_{\vartheta}(\mathbf{x}_i)| - Dd_k), \end{aligned}$$

where $D > 0$ 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 calculated from Q_2 and the unlabeled data are deemed 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.

Therefore, a mixed integer programming (MIP) problem can be obtained as follows:

$$\begin{aligned} \min_{\vartheta, \mathbf{d}} \ & \frac{\lambda}{2}\|\mathbf{w}\|^2 + \sum_{\mathbf{x}_i \in \mathcal{L}_{\pm 1}} r_i H_1(y_i f_{\vartheta}(\mathbf{x}_i)) + \sum_{\mathbf{x}_i \in \mathcal{L}_0} r_i I_{\varepsilon}(f_{\vartheta}(\mathbf{x}_i)) \\ & + \sum_{\mathbf{x}_{k+L} \in \mathcal{U}} r_{k+L} H_1(|f_{\vartheta}(\mathbf{x}_i)| + D(1 - d_k)) \\ & + \sum_{\mathbf{x}_{k+L} \in \mathcal{U}} r_{k+L} I_{\varepsilon}(|f_{\vartheta}(\mathbf{x}_i)| - Dd_k). \end{aligned} \tag{8}$$

Remark 1. In the literature, various software packages, e.g., CPLEX, have been built to solve standard MIP problems. One solution

to solve the MIP problem in Eq. (8) is to adopt these packages. However, the computational complexity is very high for solving MIP problems. It is even hard to handle an optimization problem with over 50 0/1 integer variables.

Remark 2. One straightforward relaxation is to relax the decision variables from {0, 1} to the range of [0, 1]. We know that the usage of the decision variables is to make the min loss function count by Q_1 or by Q_2. Hence, with a suitable constant D, relaxing the decision variables will not affect the performance significantly. Our sensitivity analysis in Fig. 7 verifies this claim.

4.2. Semi-definite programming transformation

One way to solve the minimization problem in Eq. (8) is to deem it as a transductive learning problem, which focuses on determining the label of the unlabeled data. As the labels in our 3C-SVM can be -1, 0, $+1$, this also introduces another difficult mixed integer programming problem. One typical solution is the SDP relaxation (Valizadegan & Jin, 2006). Following the standard SDP relaxation, we can derive the following theorem:

Theorem 2. *The optimization of (8) can be relaxed and transformed into the following semi-definite programming problem:*

$$\begin{aligned} &\min_{\mathbf{M},\mathbf{d},\boldsymbol{\nu},\boldsymbol{\delta},t} \quad t \qquad (9)\\ &s.t. \quad \begin{bmatrix} \mathbf{P} & \mathbf{a}+\boldsymbol{\nu}-\mathbf{B}^\top\boldsymbol{\delta} \\ (\mathbf{a}+\boldsymbol{\nu}-\mathbf{B}^\top\boldsymbol{\delta})^\top & t-2\boldsymbol{\delta}^\top\mathbf{C} \end{bmatrix} \succeq 0,\\ &\quad 0 \le d_j \le 1,\ j=1,\ldots,U,\\ &\quad \mathrm{rank}(\mathbf{M})=1,\ \mathbf{M}_{1:L,1:L}=\mathbf{y}_{1:L}\mathbf{y}_{1:L}^\top,\\ &\quad \mathbf{0}_N \le \mathrm{diag}(\mathbf{M}) \le \mathbf{1}_N, \end{aligned}$$

where a matrix $\mathbf{A} \succeq 0$ *means that* $\mathbf{A}$ *is a semi-definite matrix,* $\mathbf{P}$ *is defined as*

$$\mathbf{P}=\begin{bmatrix} \mathbf{K}\circ\mathbf{M} & \mathrm{diag}(\mathbf{y})\mathbf{K}_{1:N,L:N} & -\mathrm{diag}(\mathbf{y})\mathbf{K}_{1:N,L:N} \\ \mathbf{K}_{1:N,L:N}^\top\mathrm{diag}(\mathbf{y}) & \mathbf{K}_{L+1:N,L+1:N} & -\mathbf{K}_{L+1:N,L+1:N} \\ -\mathbf{K}_{1:N,L:N}^\top\mathrm{diag}(\mathbf{y}) & -\mathbf{K}_{L+1:N,L+1:N} & \mathbf{K}_{L+1:N,L+1:N} \end{bmatrix},$$

$\mathbf{B}=\begin{bmatrix} \mathbf{I}_{N\times N}, & \mathbf{0}_{N\times 2U} \\ \mathbf{0}_{U\times N}, & \mathbf{Q}_{U\times 2U} \end{bmatrix}$, $\mathbf{Q}_{U\times 2U}=[\mathbf{I}_{U\times U}, \mathbf{I}_{U\times U}]$, $\mathbf{C}=\mathbf{r}/\lambda \in \mathbb{R}^{L+2U}$, $\mathbf{a}=[\mathbf{1}_L; \mathbf{1}_U - D(\mathbf{1}-\mathbf{d}); -D\mathbf{d}; -D\mathbf{d}]$, *and* $N=L+U$.

The detailed derivation can be referred to Huang et al. (2008). The following are some remarks:

- The rank-one matrix $\mathbf{M}$ is an approximation of $(\mathbf{y}\mathbf{y}^\top)$, where we force the corresponding values for the labeled indices are known, i.e., $\mathbf{M}_{1:L,1:L}=\mathbf{y}_{1:L}\mathbf{y}_{1:L}^\top$. Moreover, due to the rank-one assumption, the diagonal matrix $\mathrm{diag}(\mathbf{y})$ can be represented by the elements of $\mathbf{M}$. That is, $\mathrm{diag}(\mathbf{y})=\mathrm{diag}(M_{11}, M_{12}, \ldots, M_{1N})$.
- We can follow the optimization methods in SSL (Valizadegan & Jin, 2006; Xu, Jin, Zhu, King, & Lyu, 2007) to further remove the rank-one constraint and obtain the above minimization problem as an exact SDP problem.
- This SDP problem in Eq. (9) can be solved in polynomial time by some packages, e.g., Sedumi (Sturm, 1999). The time complexity of the above SDP problem is bounded by $\mathcal{O}((L+U^2)^2(L+U)^{2.5})$, which yields the same time complexity as that of the relaxed transductive SVM by SDP implementation (Bie & Cristianini, 2003).

4.3. Concave–convex procedure (CCCP)

Considering 3C-SVM as transductive learning and solving it by SDP relaxation is still very time consuming. In the following, we will view 3C-SVM as semi-supervised learning and deploy the concave–convex procedure to seek the corresponding inductive rule in a more efficient way.

Absolute operator elimination. Another difficult issue on minimizing Eq. (8) is that it contains the absolute operators in the symmetrical hinge loss function and the ε-insensitive loss function. We first observe that the loss function in Q_1 is a shifted symmetrical hinge loss function and can be abstracted as follows:

$$Q_1 : H_1(|u|+a)=\max\{0, 1-|u|-a\}=H_{1-a}(|u|). \qquad (10)$$

To remove the above absolute operator, we adopt the ramp loss (Collobert et al., 2006; Wang, Shen, & Pan, 2009) to get the following approximation:

$$H_1(|u|+a) \approx H_{1-a}(u)-H_\kappa(u)+H_{1-a}(-u)-H_\kappa(-u). \qquad (11)$$

In the above, $H_s(u)=\max\{0, s-u\}$. Hence, we can calculate $H_{1-a}(u)=\max\{0, 1-a-u\}$ and $H_\kappa(u)=\max\{0, \kappa-u\}$. This transformation yields the computation of Q_1 to a summation of two symmetrical losses.

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

$$Q_2 : I_\varepsilon(|u|-a)=H_{-\varepsilon-a}(-u)+H_{-\varepsilon-a}(u). \qquad (12)$$

Since the losses calculated by Eqs. (11) and (12) are both symmetrical, we can introduce new paired-data for the unlabeled data as in Collobert et al. (2006) to simplify the calculation. The new paired-data are defined as follows:

$$\mathbf{x}_{kL}=\mathbf{x}_{k+L}, \qquad y_{kL}=1, \qquad (13)$$

$$\mathbf{x}_{kLU}=\mathbf{x}_{k+L}, \qquad y_{kLU}=-1, \quad k=1,\ldots,U. \qquad (14)$$

Similarly, if there is labeled data coming from 0-class, we have to use ε-insensitive loss function to calculate their risks. We introduce new paired data for these data as follows:

$$\mathbf{x}_{-i}=\mathbf{x}_i, \qquad y_i=-1, \quad i=1,\ldots,|\mathcal{L}_0|, \qquad (15)$$

$$\mathbf{x}_i=\mathbf{x}_i, \qquad y_i=1, \quad i=1,\ldots,|\mathcal{L}_0|. \qquad (16)$$

It is noted that for simplicity, we have overloaded the notations a little and extended them to negative indices, where the index $-i$ indicates the index shifting advance from the 0-index. When there is no $\mathcal{L}_0$ data, i.e., $|\mathcal{L}_0|=0$, we do not introduce new paired data.

CCCP. Since we introduce the parameter κ for the loss in Q_1, we redefine the problem in Eq. (8) as $Q^\kappa(\boldsymbol{\vartheta}, \mathbf{d})$ and separate it into two terms, a convex term and a concave term. That is

$$Q(\boldsymbol{\vartheta}, \mathbf{d}) \triangleq Q^\kappa(\boldsymbol{\vartheta}, \mathbf{d})=Q_{vex}(\boldsymbol{\vartheta}, \mathbf{d})+Q_{cav}^\kappa(\boldsymbol{\vartheta}), \qquad (17)$$

where

$$\begin{aligned} Q_{vex}(\boldsymbol{\vartheta}, \mathbf{d}) = {} & \frac{\lambda}{2}\|\mathbf{w}\|^2+\sum_{\mathbf{x}_i\in\mathcal{L}_{\pm 1}} r_i H_1(y_i f_{\boldsymbol{\vartheta}}(\mathbf{x}_i))+\sum_{\mathbf{x}_i\in\mathcal{L}_0} r_i I_\varepsilon(f_{\boldsymbol{\vartheta}}(\mathbf{x}_i))\\ & +\sum_{k=1}^{U} r_{k+L} H_{1-D(1-d_k)}(y_{k+L} f_{\boldsymbol{\vartheta}}(\mathbf{x}_{k+L}))\\ & +\sum_{k=1}^{U} r_{k+L} H_{1-D(1-d_k)}(y_{k+LU} f_{\boldsymbol{\vartheta}}(\mathbf{x}_{k+LU}))\\ & +\sum_{k=1}^{U} r_{k+L} H_{-\varepsilon-Dd_k}(y_{k+L} f_{\boldsymbol{\vartheta}}(\mathbf{x}_{k+L}))\\ & +\sum_{k=1}^{U} r_{k+L} H_{-\varepsilon-Dd_k}(y_{k+LU} f_{\boldsymbol{\vartheta}}(\mathbf{x}_{k+LU})), \end{aligned}$$

and

$$Q_{cav}^\kappa(\boldsymbol{\vartheta})=-\sum_{j=L+1}^{L+2U} r_j H_\kappa(y_j f_{\boldsymbol{\vartheta}}(\mathbf{x}_j)).$$

It is easily verified that Q_{vex} is a convex function and Q^{κ}_{cav} is a concave function. Hence, the optimization of $Q^{\kappa}(\boldsymbol{\vartheta}, \mathbf{d})$ is the difference of convex functions. The concave–convex procedure (CCCP) (Yuille & Rangarajan, 2003) is an efficient tool to solve this kind of problems and has been applied in large scale transductive SVMs (Collobert et al., 2006) and SVMs on data with missing values (Smola, Vishwanathan, & Hofmann, 2005).

The idea of CCCP is to use the first order Taylor expansion to approximate the concave term and to solve a sequence of problems until it converges. In Eq. (17), the concave term is Q^{κ}_{cav}. Since there is only the variable $\boldsymbol{\vartheta}$ in Q^{κ}_{cav}, we only need to apply the first order Taylor expansion of Q^{κ}_{cav} on $\boldsymbol{\vartheta}^t$. Hence, we can seek the optimal variables by solving a sequence of the following optimization problems:

$$\min_{\boldsymbol{\vartheta}, \mathbf{d}} \left(Q_{vex}(\boldsymbol{\vartheta}, \mathbf{d}) + \frac{\partial Q^{\kappa}_{cav}(\boldsymbol{\vartheta}^t)}{\partial \boldsymbol{\vartheta}} \cdot \boldsymbol{\vartheta} \right). \tag{18}$$

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 (Wolsey, 1998): by (1) relaxing the integer variable to a real variable, then solving the whole optimization together; (2) rounding the corresponding variable to get its integer solution.

For 3C-SVM in Eq. (18), we relax the decision variable d_k from $\{0, 1\}$ to $[0, 1]$ and solve the optimization problem in Eq. (18) first. We then determine the value of d_k by its definition, the error incurred is less when the data are assigned to the associated class, as follows:

$$d_k = \begin{cases} 1 & \text{if } \xi_k \le \xi^*_k \\ 0 & \text{otherwise,} \end{cases} \tag{19}$$

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 Eq. (18), we define

$$\mu_{k+\mathfrak{s}} = y_{k+\mathfrak{s}} \frac{\partial Q^{\kappa}_{cav}(\boldsymbol{\vartheta})}{\partial f_{\boldsymbol{\vartheta}}(\mathbf{x}_{k+\mathfrak{s}})} = \begin{cases} r_{k+\mathfrak{s}} & \text{if } y_{k+\mathfrak{s}} f_{\boldsymbol{\vartheta}}(\mathbf{x}_{k+\mathfrak{s}}) < \kappa \\ 0 & \text{otherwise,} \end{cases} \tag{20}$$

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

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

The following theorem summarizes the final result of solving the relaxed optimization in Eq. (18):

Theorem 3. *The dual problem of the relaxed optimization problem in Eq. (18) is a Quadratic Programming (QP) problem as follows:*

$$\max_{\boldsymbol{\alpha}, \boldsymbol{\alpha}^*} \; -\frac{1}{2\lambda}[\boldsymbol{\alpha}; \boldsymbol{\alpha}^*]^{\top} \boldsymbol{\Omega} [\boldsymbol{\alpha}; \boldsymbol{\alpha}^*] + \boldsymbol{\varrho}^{\top} [\boldsymbol{\alpha}; \boldsymbol{\alpha}^*] \tag{21}$$

$$\begin{aligned} s.t. \quad & \mathbf{0} \le \boldsymbol{\alpha}, \boldsymbol{\alpha}^* \le \mathbf{r}, \\ & \mathbf{A}_e [\boldsymbol{\alpha}; \boldsymbol{\alpha}^*] = \boldsymbol{\mu}^{\top} \mathbf{Y}_{\bullet 2U}, \\ & \mathbf{A} [\boldsymbol{\alpha}; \boldsymbol{\alpha}^*] \le \mathbf{0}, \end{aligned}$$

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

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

Here, the index I *represents the indexes of all data from* $-|\mathcal{L}_0|$ *to* $L + 2U$ *and* $I_{\mathcal{U}}$ *represents all indexes for unlabeled data from* $L + 1$ *to* $L+2U$. $\mathbf{A}_e = [\mathbf{Y}; \mathbf{Y}_{\bullet 2U}]$ *and* $\mathbf{A} = \begin{bmatrix} \mathbf{0}_{U,L}, -\mathbf{I}_U, -\mathbf{I}_U, \mathbf{I}_U, \mathbf{I}_U \end{bmatrix}$, $\mathbf{Y}$ *is a vector consisting of the labels 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. (21) corresponds to the dual form of the optimization on Eq. (18). Detailed derivative is given in the Appendix.

After solving the QP problem in Eq. (21), we obtain $\mathbf{w}$ as a linear combination of the dual variables, $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^*$,

$$\mathbf{w} = \frac{1}{\lambda} \left(\sum_{i=-|\mathcal{L}_0|, i \ne 0}^{L+2U} \alpha_i y_i \phi(\mathbf{x}_i) + \sum_{i=L+1}^{L+2U} (\alpha^*_i - \mu_i) y_i \phi(\mathbf{x}_i) \right), \tag{22}$$

and the variable b corresponds to the dual variable of the equality constraint. It is noted that 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 are *support vectors*. An illustration is shown in Fig. 4(b). These support vectors play the role of controlling the decision function.

Algorithm 1 is guaranteed to be converged in a finite number of steps. The following theorem claims this statement:

Theorem 4. *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

$$\begin{aligned} & Q_{vex}(\boldsymbol{\vartheta}^{t+1}, \mathbf{d}) + \partial Q^{\kappa}_{cav}(\boldsymbol{\vartheta}^t) \cdot \boldsymbol{\vartheta}^{t+1} \\ & \quad \le Q_{vex}(\boldsymbol{\vartheta}^t, \mathbf{d}) + \partial Q^{\kappa}_{cav}(\boldsymbol{\vartheta}^t) \cdot \boldsymbol{\vartheta}^t \end{aligned} \tag{23}$$

$$Q^{\kappa}_{cav}(\boldsymbol{\vartheta}^{t+1}) \le Q^{\kappa}_{cav}(\boldsymbol{\vartheta}^t) + \partial Q^{\kappa}_{cav}(\boldsymbol{\vartheta}^t) \cdot (\boldsymbol{\vartheta}^{t+1} - \boldsymbol{\vartheta}^t), \tag{24}$$

where $\partial Q^{\kappa}_{cav}$ defines the partial derivative of Q^{κ}_{cav} with respect to $\boldsymbol{\vartheta}$. Hence, summing (23) and (24) together, we get $Q^{\kappa}(\boldsymbol{\vartheta}^{t+1}, \mathbf{d}) \le Q^{\kappa}(\boldsymbol{\vartheta}^t, \mathbf{d})$ 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 (24) is strict unless $\boldsymbol{\mu}$ remains unchanged. ■

Remark 3. Although the optimization of 3C-SVM is non-convex, which yields the local optimal problem, we have alleviated it by appropriate initialization and rounding recovery. Our empirical test shows that our 3C-SVM works well on this initialization and at each iteration, where $Q^{\kappa}(\boldsymbol{\vartheta}^t, \mathbf{d}^t)$ actually decreases in each step; see the convergence results in Fig. 5.

Complexity analysis. Practically, the number of QP sequences in Algorithm 1 is a constant, usually less than 10; see Fig. 5 for a trial result. Hence, 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 yields a worst case complexity of $\mathcal{O}((|\mathcal{L}_0| + L + 4U)^3)$ (Goldfarb & Liu, 1991; Schölkopf & Smola, 2002). Possible tricks may be applied to speed up the 3C-SVM algorithm in a quadratic scale (Collobert et al., 2006; Schölkopf & Smola, 2002). 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.

Algorithm 1 The Concave-Convex Procedure for 3C-SVMs

Initialization:
$t = 0$
Calculate $\boldsymbol{\vartheta}^0 = (\mathbf{w}^0, b^0)$ by solving SVM/$\mathcal{U}$-SVM on the labeled 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 (21) to obtain $\boldsymbol{\vartheta}^t$
Update $\mathbf{d}^t$ from Eq. (19)
Update $\boldsymbol{\mu}^t$ from Eq. (20)
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 Eq. (20) with $\boldsymbol{\vartheta}^t$
Solve the optimization in (21) to obtain $\boldsymbol{\vartheta}^t$
Update $\mathbf{d}^t$ from Eq. (19)
Update $\boldsymbol{\mu}^t$ from Eq. (20)
end while
until $|\boldsymbol{\mu}^{t+1} - \boldsymbol{\mu}^t| \leq \epsilon$

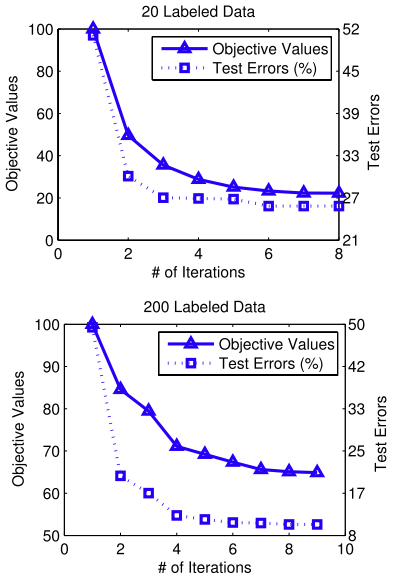


Fig. 5. The left part shows the concave–convex procedure for 3C-SVMs. The right part shows the convergence of algorithms (objective values/test errors vs. # of iterations) during the CCCP iterations of training 3C-SVM on two toy datasets (single trial).

4.4. Balance constraint

Usually, balance constraint is required in semi-supervised support vector machine (Joachims, 1999). Although our formulation of 3C-SVM in Eq. (4) does not include the balance constraint on the unlabeled data, in the following section, we show that balance constraint can be easily incorporated in our formulation.

The balance constraint is based on the following two observations: First, data from $\mathcal{U}_{\pm 1}$ require a balance constraint (Vapnik & Kotz, 2006). Second, data from $\mathcal{U}_0$ do not require a balance constraint since $\mathcal{U}_0$ data approach to the decision hyperplane. That is, their function values are close to zero. Hence, summarizing the function values of all unlabeled data, we can obtain the same balance constraint as (Collobert et al., 2006),

$$\frac{1}{U}\sum_{t=L+1}^{L+U} f_{\boldsymbol{\vartheta}}(\mathbf{x}_t) \approx \frac{1}{L}\sum_{i=1}^{L} y_i. \tag{25}$$

The balance constraint is a linear constraint. One dual variable can be introduced to the optimization problem of Eq. (18), which can easily be transformed into the kernel form of Eq. (21) as in Collobert et al. (2006).

It is noted that the summation of y_i will alter the effect of the balance constraint in Eq. (25). In practical, we find that the balance constraint is insensitive to the model performance since the $\mathcal{U}_0$ data may have played the role of controlling the optimal subspace of the decision function partially. To attain a better performance, one possible solution is to introduce a constant η and to transform the balance constraint as $\frac{1}{U}\sum_{t=L+1}^{L+U} f_{\boldsymbol{\vartheta}}(\mathbf{x}_t) = \eta$, which is related to the portion of the number of the unlabeled data assigning to the positive class (Chapelle et al., 2006). However, this will introduce a hyperparameter, which increases the difficulty of parameters tuning.

5. Experiments

In this section, we conduct empirical evaluation to show the performance of our proposed 3C-SVM algorithm on both synthetic and real-world datasets. Our 3C-SVM algorithm[1] is implemented in Matlab 7.3 and the QP problem is solved by a general optimization toolbox, MOSEK.[2] The codes are run on a PC with Intel Quad CPU Q9650@3.00 GHz and 8.00G RAM.

5.1. Synthetic datasets

We first test how 3C-SVM performs comparing with its three specific cases: SVM (Vapnik, 1999), S^{3}VM (Collobert et al., 2006), and $\mathcal{U}$-SVM (Weston et al., 2006). To control the setting, we evaluate on two synthetic datasets, which are generated similar to the setup in Sinz et al. (2008).

Data generation. The synthetic datasets consist of 50-dimensional data from ± 1-class and two different kinds of $\mathcal{U}_0$ data. The ± 1-class is the same for both synthetic datasets, following the generation scheme in Sinz et al. (2008), where the means are $c_i^{\pm} = \pm 0.3$ for $i = 1, \ldots, 50$ and variance values are $\sigma_{1,2}^2 = 0.08$ and $\sigma_{3,\ldots,50}^2 = 10$. This setting generates two Gaussians with the Bayes risk being approximately 5%. Two kinds of $\mathcal{U}_0$ data are generated as follows:

- The $\mathcal{U}_0$ data of the first synthetic data contain a zero mean with $\sigma_{1,2}^2 = 0.1$ and $\sigma_{3,\ldots,50}^2 = 10$. It is noted that $\mathcal{U}_0$ data contain larger variances on the first two dimensions of the data than those of the ± 1-class data, but the optimal Bayesian decision hyperplane passes through the origin, the center of the $\mathcal{U}_0$ data.
- In the second synthetic dataset, 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, i.e., a little bias from the Bayesian optimal classifier.

As reported in Table 3, 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] Our 3C-SVM toolbox can be downloaded in https://www.dropbox.com/s/9u8emz00a70b3ga/demo_3CSVM.rar?dl=0.
[2] http://www.mosek.com.

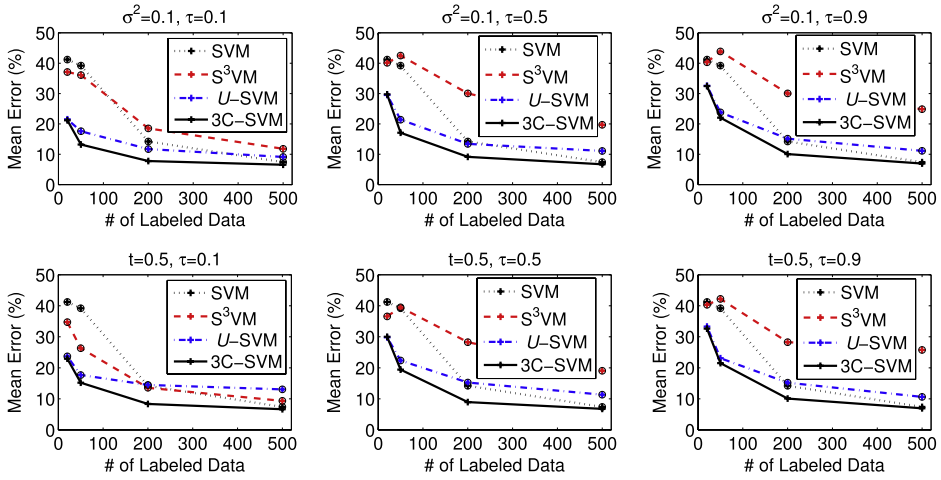


Fig. 6. The performance of four algorithms on toy datasets with different combinations of mixed unlabeled data. 3C-SVMs consistently achieve the best results over all other models. Results marked by circles indicate 3C-SVMs outperform the corresponding models with 95% significant level on paired t-test.

Table 3
Data description.

Dataset	d	L	U	U_0
Synthetic	50	20, 50 200, 500	500	Two designed cases
USPS	256	10	$\{10^2, 10^3\}$	Except "5" and "8"
MNIST	784	10	$\{10^2, 10^3\}$	Except "5" and "8"

$\pm$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.

Comparison. Since the optimal decision hyperplane is linear for both synthetic datasets, we employ the linear kernel in fitting the data, and tune the hyperparameters of all the compared models on separated validation sets. The detailed tuning procedure is as follows:

- For SVM, we tune the soft-margin hyperparameter $C \in 10^{[-1:1:3]}$.
- For S^{3}VM, we apply the grid search to tune the soft-margin hyperparameter C for labeled data and the trade-off hyperparameter $C_{\mathcal{U}}$ for unlabeled data, where $C \in 10^{[-1:1:3]}$ and $C_{\mathcal{U}} \in 10^{[-1:1:2]}$. The approximate parameter κ for the ramp loss is searched in $\{0.01, 0.1, 0.2, 0.5\}$.
- For $\mathcal{U}$-SVM, C and $C_{\mathcal{U}}$ are tuned by the grid search in the same range of S^{3}VM, while the parameter ε for the ε-insensitive loss function is tuned in $\{0.01, 0.1, 0.2, 0.5, 0.8, 1\}$.
- For 3C-SVM, the corresponding hyperparameters include the regularized parameter λ, the trade-off parameter $\mathbf{r}$ for labeled and unlabeled data, and the parameters for the relaxed min loss function, D, ε, and κ. Practically, as $\lambda = 1/C$, we set $r_i = 1$ for the labeled data and $r_j = C_{\mathcal{U}}/C$ for the unlabeled data. Hence, we first fix D to 2, ε to 0.01, and κ to 0.01. We then tune C and $C_{\mathcal{U}}$ by the grid search, where the search range of $C_{\mathcal{U}}$ is the same as S^{3}VM and the search range is three successive values of the optimal C found in SVM. For example, if $C = 10$ is the best regularization parameter for SVM, we tune C from $\{1, 10, 100\}$. After finding the best C and C_U, we tune D, ε, and κ one by one, where D is tested in $\{1, 2, 5, 10\}$, ε is tested in the same range of $\mathcal{U}$-SVM, and κ is tested in the same range in S^{3}VM.

After obtaining the best hyperparameters for the corresponding compared models, we perform 10-fold cross validation on the generated datasets. Fig. 6 reports the average performance on the two synthetic datasets and three different settings for the unlabeled data. From the results, we have the following observations:

- 3C-SVM consistently attains the best results among four compared models. More importantly, 3C-SVM achieves 64 significantly better results among all 72 compared cases.
- For $\mathcal{U}$-SVM, the performance decreases as the number of $\mathcal{U}_0$ data decreases. This indicates that the $\mathcal{U}_0$ data actually can help to seek the optimal decision hyperplane. However, when the number of labeled data is large, $\mathcal{U}$-SVM cannot even beat SVM. This indicates that the "unclean" universum data really hurt the performance of $\mathcal{U}$-SVM.
- For S^{3}VM, the performance is much worse than that of $\mathcal{U}$-SVM and 3C-SVM. It is even worse than SVM when the number of labeled data is small, e.g., $L = 100$. These results clearly show that without properly selecting the unlabeled data, S^{3}VM is easily polluted by the "unclean" unlabeled data.

5.2. Real-world handwritten digit datasets

We conduct empirical evaluation on two popular benchmark handwritten digit datasets, the USPS dataset and the MNIST dataset, which have been frequently employed in evaluating the performance of semi-supervised classifiers (Collobert et al., 2006; Schölkopf & Smola, 2002). Here, the objective is to get a more complete comparison of our proposed 3C-SVM with the three related maximum margin based models and two more state-of-the-art semi-supervised classifiers: Laplacian Support Vector Machines (LapSVMs) (Belkin et al., 2006; Melacci & Belkin, 2011) and Semi-supervised Extreme Learning Machines (SS-ELMs) (Huang et al., 2014; Iosifidis et al., 2014).

Data description. As described in Table 3, each image in USPS was normalized and centered with the size of 16 $\times$ 16, which forms 256-dimensional data. This dataset contains 9298 grayscale handwritten digit images, 7291 of which are used as the training set, while the remaining 2007 are used as the test set. The MNIST dataset consists of a training set of 60,000 digits and a test set of 10,000 digits. The digits are grayscale handwritten images normalized and centered in 28 $\times$ 28, which form 784-dimensional data. We have normalized each pixel value in an image to the range of -1 and 1.

Similar to the setup in Sinz et al. (2008) and Weston et al. (2006), we employ digits "5" and "8" to construct the $\pm$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

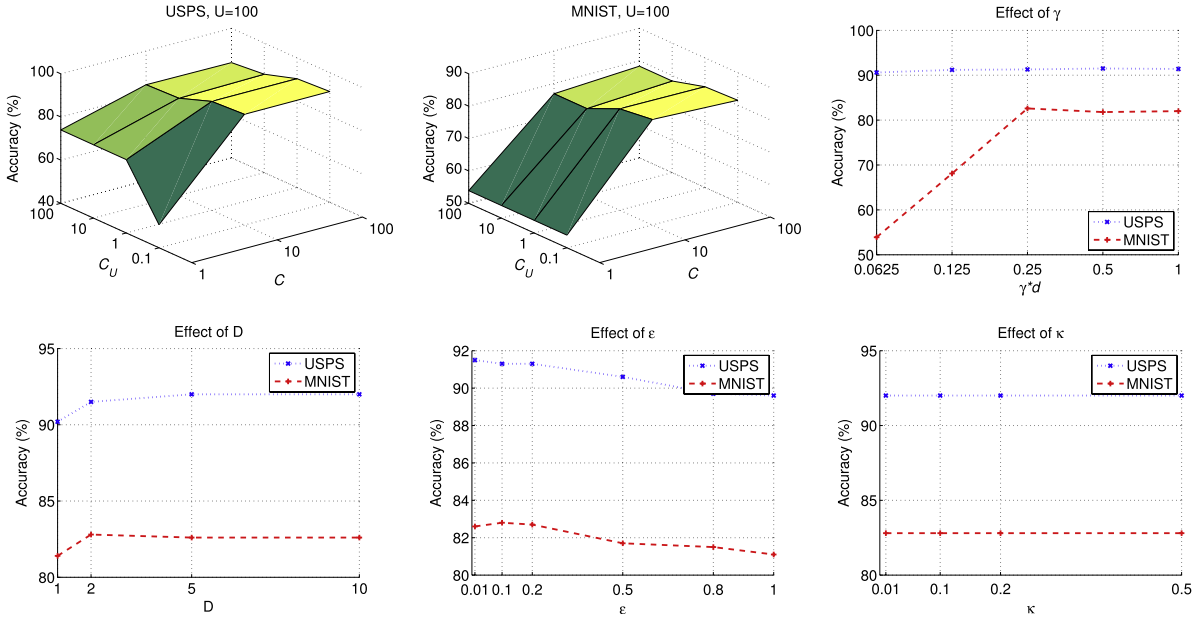


Fig. 7. Effect of hyperparameters for 3C-SVM on the validation sets of two handwritten digit datasets, see text for more details.

unlabeled data is 100 and 1000, respectively, while the proportion of the mixed unlabeled data is set as $(\tau U, (1-\tau)U)$, 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".

Comparison. Since the data are linearly nonseparable in the original feature space and the RBF kernel is adopted (Schölkopf & Smola, 2002), we employ it in all models except SS-ELMs. The RBF kernel function, $K(\mathbf{x}, \mathbf{y}) = \exp(-\gamma\|\mathbf{x}-\mathbf{y}\|^2)$ introduces a new parameter, γ, the width of the RBF kernel. We follow the same tuning procedure as outlined in the synthetic datasets to tune the hyperparameters for all models on separate validation sets. By adopting the RBF kernel, we have to tune the kernel width. The tuning procedure is as follows:

- For SVM, C and γ are tuned by grid search. Here C is selected from the same set as that in synthetic datasets. Following suggestion in Schölkopf and Smola (2002), γ is set to $\delta \times \frac{1}{d}$, where d is the number of data dimension, i.e., 256 for USPS and 784 for MNIST. δ is searched from $\{\frac{1}{32}, \frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1, 2, 4, 8\}$.
- For $\mathcal{U}$-SVM, we first fix ε to 0.01 C. We then tune C, $C_{\mathcal{U}}$, and γ by grid search. The ranges of C and $C_{\mathcal{U}}$ are the same as those in the synthetic datasets. γ is tested the same range as that in SVM. After that, we further tune ε in $\{0.01, 0.1, 0.2, 0.5, 0.8, 1\}$.
- For S^{3}VM, C, $C_{\mathcal{U}}$, and γ are tested in the same range as those in $\mathcal{U}$-SVM. κ is tested in $\{0.01, 0.1, 0.2, 0.5\}$.
- For LapSVM, the corresponding regularized parameters, γ_A and γ_I, are equivalent to the reciprocal of C and $C_{\mathcal{U}}$. Hence we tune them in the same range of C in SVM and $C_{\mathcal{U}}$ in $\mathcal{U}$-SVM. γ is tested in the same range as that in SVM.
- For SS-ELM, the corresponding regularized parameters are C and λ that are the same as γ_A and γ_I in LapSVM. Since the random map of the Sigmoid function can obtain a better performance than the Gaussian function, we adopt the Sigmoid function in the test. The number of random maps is set to 2000 as suggested in Huang et al. (2014).
- For 3C-SVM, the tuning hyperparameters include C, $C_{\mathcal{U}}$, γ, D, ε, and κ. To relieve the hyperparameters tuning, we first set D to 2, κ to 0.01, ε to 0.01 and the kernel width γ to be the optimal one found by SVM. We then tune C and $C_{\mathcal{U}}$ as the same procedure in synthetic datasets. After that, we further tune γ, D, ε, and κ one by one, where the range of γ is five consecutive values of the best γ found in SVM and the tuning range of D, ε, and κ is the same as the above for the corresponding parameters. We present more details in Section 5.3.

Table 4 reports the average (10 runs) accuracies of the six competing algorithms on the two handwritten digit datasets. 3C-SVM consistently attains better results in nearly all cases. By examining the details of the results, we have the following observations:

- For SVM, we observe that by appropriately tuning the hyperparameters, SVM can also achieve satisfactory results that are even better than those models polluted by "unclean" labeled data, e.g., LapSVM when $\tau = 0.1$ for the USPS dataset and S^{3}VM when $U = 1000$ for both handwritten datasets.
- For S^{3}VM, it is observed that this algorithm is very sensitive to the unlabeled data. When the number of unlabeled data increases from 100 to 1000, its performance decreases significantly when τ is 0.5 and 0.9. The mixed unlabeled data really hurt the performance of S^{3}VM.
- For $\mathcal{U}$-SVM, the performance reveals difference characteristics for these two handwritten datasets. For USPS, the performance of $\mathcal{U}$-SVM decreases slightly as the number of $\mathcal{U}_0$ data decreases, while the performance decreases significantly and is even worse than SVM when the number of unlabeled data increases. For MNIST, the performance of $\mathcal{U}$-SVM is relatively stable and is better than SVM. The results show that $\mathcal{U}_0$ data can help improve the performance of $\mathcal{U}$-SVM, but the mixed unlabeled data can also hurt the model performance.
- For LapSVM, the performance increases as the number of unlabeled data increases for both datasets. It implies that the graph Laplacian can capture more information when there are more unlabeled data.
- For SS-ELM, the performance follows a similar pattern to LapSVM. This may be due to the advantage of graph Laplacian in capturing the local structure of the data. By appropriately selecting the random mapping, SS-ELM can attain relative stable performance for both models with different number of unlabeled data.

Table 4
The average (10 runs) accuracies (%) of SVM, S^3VM, $\mathcal{U}$-SVM, LapSVM, SS-ELM, and 3C-SVM on the USPS and the MNIST ("5" vs. "8") datasets for different combinations of mixed unlabeled data. The best accuracy is denoted in bold. 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	85.6 ± 5.5 (**0.8**)	85.6 ± 5.5 (**2.4**)	85.6 ± 5.5 (**0.4**)
		S^3VM	87.6 ± 4.2 (7.1)	87.6 ± 4.2 (8.8)	87.6 ± 4.2 (7.6)
		$\mathcal{U}$-SVM	89.0 ± 2.9 (34.3)	88.2 ± 3.0 (**2.7**)	87.4 ± 3.5 (**1.7**)
		LapSVM	83.7 ± 6.2 (**1.0**)	85.1 ± 5.8 (**2.2**)	85.6 ± 6.1 (**1.2**)
		SS-ELM	87.8 ± 3.4 (**5.0**)	88.4 ± 3.0 (47.7)	88.1 ± 2.2 (15.6)
		3C-SVM	**89.3** ± 2.6	**88.8** ± 3.0	**89.1** ± 3.7
USPS	$L = 10$ $U = 1000$	SVM	85.6 ± 5.5 (**2.0**)	85.6 ± 5.5 (7.2)	85.6 ± 5.5 (9.0)
		S^3VM	84.0 ± 10.2 (**1.4**)	76.8 ± 18.7 (**2.8**)	76.0 ± 20.2 (**4.7**)
		$\mathcal{U}$-SVM	86.1 ± 3.4 (**2.1**)	83.4 ± 3.3 (**0.0**)	78.3 ± 4.0 (**0.0**)
		LapSVM	85.3 ± 6.4 (**3.6**)	85.6 ± 6.1 (**4.2**)	85.5 ± 6.3 (15.1)
		SS-ELM	86.6 ± 3.4 (**1.1**)	86.9 ± 2.5 (8.6)	86.1 ± 3.4 (74.1)
		3C-SVM	**88.6** ± 2.8	**87.6** ± 3.2	**86.4** ± 3.7
MNIST	$L = 10$ $U = 100$	SVM	68.7 ± 13.5 (**1.5**)	68.7 ± 13.5 (**1.3**)	68.7 ± 13.5 (**1.0**)
		S^3VM	69.4 ± 11.6 (**4.7**)	69.1 ± 12.0 (**4.5**)	68.5 ± 12.5 (**1.8**)
		$\mathcal{U}$-SVM	68.8 ± 13.5 (**2.0**)	68.8 ± 13.5 (**2.6**)	68.8 ± 13.5 (**2.4**)
		LapSVM	63.7 ± 10.7 (**0.3**)	65.1 ± 11.4 (**0.8**)	65.9 ± 11.5 (**1.6**)
		SS-ELM	72.2 ± 9.5 (15.6)	71.6 ± 9.2 (**0.3**)	71.7 ± 9.6 (**4.9**)
		3C-SVM	**73.0** ± 9.5	**72.9** ± 9.3	**73.0** ± 9.3
MNIST	$L = 10$ $U = 1000$	SVM	68.7 ± 13.5 (**2.0**)	68.7 ± 13.5 (**2.6**)	68.7 ± 13.5 (**3.9**)
		S^3VM	68.0 ± 4.1 (**1.2**)	58.2 ± 2.2 (**0.0**)	56.6 ± 1.6 (**0.0**)
		$\mathcal{U}$-SVM	72.0 ± 9.6 (17.2)	72.5 ± 9.3 (24.8)	72.7 ± 9.2 (36.8)
		LapSVM	65.0 ± 10.7 (**1.0**)	66.3 ± 10.8 (**2.6**)	66.9 ± 11.6 (**4.9**)
		SS-ELM	72.7 ± 9.7 (27.0)	72.8 ± 9.4 (35.6)	72.8 ± 9.3 (49.3)
		3C-SVM	**73.5** ± 8.8	**73.5** ± 8.6	**73.5** ± 8.3

- For 3C-SVM, it attains the best results among all compared cases. Especially, it significantly outperforms SVM and S^3VM at 90% significance level, while outperforms LapSVM at 95% significance level. It is also significantly better than $\mathcal{U}$-SVM for 8 cases and SS-ELM for 4 cases among all 12 compared cases. Overall, the performance of 3C-SVM is relatively stable with respect to different τ.

5.3. Sensitivity analysis

Here, we conduct sensitivity analysis on the hyperparameters of 3C-SVM for $U = 100$ with the validation sets of both handwritten digit datasets. For $U = 1000$, the analysis is similar. As the best kernel width γ is $\frac{1}{d} \times \frac{1}{4}$ in SVM, we fix γ to that value and set other parameters for the min loss function as default, i.e., $D = 2$, $\varepsilon = 0.01$, and $\kappa = 0.01$. We then tune C and $C_{\mathcal{U}}$ by the grid search in the range of $\{1, 10, 100\}$ and $\{0.1, 1, 10\}$, respectively. After that, we tune γ in $\frac{1}{d} \times \{\frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1\}$, five consecutive values of the best γ in SVM. Next, we tune D in $\{1, 2, 5, 10\}$, ε in $\{0, 0.01, 0.1, 0.2, 0.5, 0.8, 1.0\}$, and κ in $\{0.01, 0.1, 0.2, 0.5\}$. Fig. 7 shows the performance of 3C-SVM with different hyperparameter settings when $\tau = 0.5$. From these results, we have the following observations:

- For the regularized parameters, 3C-SVM achieves the best performance when $C = 10$ and $C_{\mathcal{U}} = 0.1$ for both USPS and MNIST datasets. The regions in Fig. 7(a) and (b) show that the performance of 3C-SVM becomes more stable and better when C is large and $C_{\mathcal{U}}$ is small.
- For the kernel width of the RBF kernel, 3C-SVM attains the best performance when γ is $\frac{1}{d} \times \frac{1}{2}$ for USPS and γ is $\frac{1}{d} \times \frac{1}{4}$ for MNIST. The results in Fig. 7(c) show that searching around the best kernel parameter found by SVM is an efficient way to identify the best kernel parameter for 3C-SVM.
- For the parameter D, it is observed that the best performance of 3C-SVM is attained when $D = 5$ for USPS and $D = 2$ for MNIST. The results in Fig. 7(d) reveal that when D is relatively large, the performance is stable.
- For the parameter ε, the best performance of 3C-SVM is obtained when ε is 0.01 for USPS and 0.1 for MNIST. The curves in Fig. 7(e) show that when the value of ε increases, the performance decreases gradually. Finally, the best performance is obtained when ε is small.
- From results in Fig. 7(f), it is observed that the performance is insensitive with respect to the parameter κ.
- Overall, we observed that the regularization parameters, C and $C_{\mathcal{U}}$, and the kernel parameter γ will significantly affect the performance of 3C-SVM. When the above best parameters are found, the parameters of D, ε, and κ for the min loss function can fine-tune the final performance of 3C-SVM.

5.4. Efficiency of 3C-SVM

The convergence of 3C-SVM has been demonstrated in the right part of Fig. 5, one trial result of the objective function values and test errors at each CCCP iteration with different number of labeled data ($L = 20$ and $L = 200$) and fixed number of unlabeled data ($U = 500$). The results show that at each CCCP iteration, both the objective function values and test errors decrease and the algorithm converges in a constant number of iterations, as shown in less than 10 iterations.

In the following, we show the efficiency of 3C-SVM by comparing its implementation of CCCP with the implementation of SDP (Huang et al., 2008). In the test, we adopt the first kind of the synthetic data. The number of labeled data from $\mathcal{L}_{\pm 1}$ is set to 20 and 200, respectively, and the number of unlabeled data is selected from $\{20, 50, 100, 200, 500, 1000\}$. τ equals 0.5, a balance setting for the unlabeled data. Fig. 8 shows the time cost of 3C-SVM implemented by CCCP and SDP, respectively, in the log–log scale. It is shown that 3C-SVM implemented by SDP spends much more time (over hundreds, or even thousand times slower) than 3C-SVM implemented by CCCP. It is noted that our 3C-SVM can still be

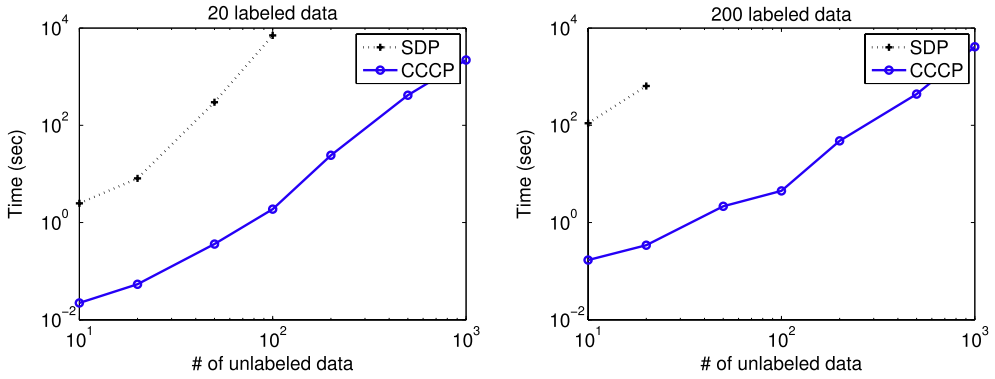


Fig. 8. Time cost of 3CSVM on the first synthetic dataset.

improved by controlling the sparsity of the solution and adopting the warm-start scheme. We leave this work as future work.

6. Conclusion

In this paper, we have proposed a maximum margin semi-supervised model, named 3C-SVM, to learn from labeled and mixed unlabeled data. In order to alleviate the effect of mixed unlabeled data, we build up the formulation based on the logistic principle and maximum entropy principle. More specifically, we introduce a new min loss function to distinguish the mixed unlabeled data into relevant and irrelevant data based on which error occurring is smaller when assigning the data to the associated class. The irrelevant data can then play the role on seeking the optimal decision subspace. Moreover, 3C-SVM generalizes several popular maximum margin classifiers, including SVMs, S^3VMs, and $\mathcal{U}$-SVMs. Furthermore, in implementation, we transform and relax 3C-SVM from an integer programming problem to solve a sequence of QP problems. The approximation by the concave–convex procedure can speed up the model significantly and yield the same worst case time complexity as that of S^3VMs. We demonstrate the effectiveness and efficiency of 3C-SVM through a series of experiments.

There are some interesting research problems left. One direction is to extend the model to solve the multi-class semi-supervised classification problem. The other direction is to design more efficient scheme for tuning the models' hyperparameters, e.g., via Bayesian inference framework. Finally, it is worthy to explore the sparsity structure of the problem and apply the warm-start technique to further speed up 3C-SVM.

Acknowledgments

The work described in this paper was fully supported by the National Grand Fundamental Research 973 Program of China (Nos. 2014CB340401 and 2014CB340405), the Research Grants Council of the Hong Kong Special Administrative Region, China (Project Nos. CUHK 413213 and CUHK 415113), National Natural Science Foundation of China (NSFC) (No. 61473236), and Microsoft Research Asia Regional Seed Fund in Big Data Research (Grant No. FY13-RES-SPONSOR-036).

Appendix. Proof of Theorem 3

Proof. With all available data, labeled $\mathcal{L}_{\pm 1}$ data (new paired $\mathcal{L}_0$ data as in Eqs. (15) and (16)), and new paired unlabeled data as in Eqs. (13) and (14), we can expand the relaxed optimization problem Eq. (18) as follows:

$$\begin{aligned}
&\min_{\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\xi}^*,\mathbf{d}} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \sum_{i=-|\mathcal{L}_0|,i\neq 0}^{L} r_i\xi_i + \sum_{i=L+1}^{L+2U} r_i(\xi_i+\xi_i^*) \\
&\quad + \sum_{i=L+1}^{L+2U} \mu_i y_i f_{\vartheta}(\mathbf{x}_i) \\
&\text{s.t.} \quad \begin{cases}
y_i f_{\vartheta}(\mathbf{x}_i) + \varepsilon + \xi_i \geq 0, & i = -|\mathcal{L}_0|,\ldots,|\mathcal{L}_0|, i\neq 0 \\
y_i f_{\vartheta}(\mathbf{x}_i) - 1 + \xi_i \geq 0, & i = |\mathcal{L}_0|+1,\ldots,L, \\
y_{k+L} f_{\vartheta}(\mathbf{x}_{k+L}) + D(1-d_k) - 1 + \xi_{k+L} \geq 0, & \\
y_{k+LU} f_{\vartheta}(\mathbf{x}_{k+LU}) + D(1-d_k) - 1 + \xi_{k+LU} \geq 0, & \\
y_{k+L} f_{\vartheta}(\mathbf{x}_{k+L}) + Dd_k + \varepsilon + \xi_{k+L}^* \geq 0, & \\
y_{k+LU} f_{\vartheta}(\mathbf{x}_{k+LU}) + Dd_k + \varepsilon + \xi_{k+LU}^* \geq 0, & \\
\xi_i \geq 0, \quad i = -|\mathcal{L}_0|,\ldots,L+2U, i\neq 0, & \\
\xi_i^* \geq 0, \quad i = L+1,\ldots,L+2U, & \\
0 \leq d_k \leq 1, \quad k = 1,\ldots,U.
\end{cases}
\end{aligned} \tag{26}$$

This is a standard QP problem with inequality constraints. Standard Lagrange multiplier method (Bertsekas, 1999; Boyd & Vandenberghe, 2004) can be adopted to seek its dual form. Hence, we construct the corresponding Lagrange function, $\mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\xi}_i, \mathbf{d}, \boldsymbol{\alpha}, \boldsymbol{\alpha}^*, \boldsymbol{\gamma}, \boldsymbol{\gamma}^*, \mathbf{p}, \mathbf{q})$, as follows:

$$\begin{aligned}
\mathcal{L} = {} & \frac{\lambda}{2}\|\mathbf{w}\|^2 + \sum_{i=-|\mathcal{L}_0|,i\neq 0}^{L} r_i\xi_i + \sum_{L+1}^{L+2U} r_i(\xi_i+\xi_i^*) \\
& + \sum_{i=L+1}^{L+2U} \mu_i y_i f_{\vartheta}(\mathbf{x}_i) - \sum_{i=-|\mathcal{L}_0|,i\neq 0}^{|\mathcal{L}_0|} \alpha_i(y_i f_{\vartheta}(\mathbf{x}_i) + \varepsilon + \xi_i) \\
& - \sum_{i=|\mathcal{L}_0|+1}^{L} \alpha_i(y_i f_{\vartheta}(\mathbf{x}_i) - 1 + \xi_i) \\
& - \sum_{k=1}^{U} \alpha_{k+L}(y_{k+L} f_{\vartheta}(\mathbf{x}_{k+L}) + D(1-d_k) - 1 + \xi_{k+L}) \\
& - \sum_{k=1}^{U} \alpha_{k+LU}(y_{k+LU} f_{\vartheta}(\mathbf{x}_{k+LU}) + D(1-d_k) - 1 + \xi_{k+LU}) \\
& - \sum_{k=1}^{U} \alpha_{k+L}^*(y_{k+L} f_{\vartheta}(\mathbf{x}_{k+L}) + Dd_k + \varepsilon + \xi_{k+L}^*) \\
& - \sum_{k=1}^{U} \alpha_{k+LU}^*(y_{k+LU} f_{\vartheta}(\mathbf{x}_{k+LU}) + Dd_k + \varepsilon + \xi_{k+LU}^*) \\
& - \sum_{i=-|\mathcal{L}_0|,i\neq 0}^{L+2U} \gamma_i\xi_i - \sum_{i=L+1}^{L+2U} \gamma_i^*\xi_i^* - \sum_{k=1}^{U} p_k(1-d_k) - \sum_{k=1}^{U} q_k d_k.
\end{aligned}$$

Hence, taking the derivative of $\mathcal{L}$ with respect to the primal variables, setting them to zeros, and utilizing the conditions of $\boldsymbol{\gamma} \geq \mathbf{0}$ and $\boldsymbol{\gamma}^* \geq \mathbf{0}$, we obtain

$$\mathbf{w} = \frac{1}{\lambda}\left(\sum_{\substack{i=-|\mathcal{L}_0| \\ i\neq 0}}^{L+2U} \alpha_i y_i \phi(\mathbf{x}_i) + \sum_{i=L+1}^{L+2U} (\alpha_i^* - \mu_i) y_i \phi(\mathbf{x}_i)\right), \tag{27}$$

and

$$\sum_{i=-|\mathcal{L}_0|, i\neq 0}^{L+2U} \alpha_i y_i + \sum_{i=L+1}^{L+2U} \alpha_i^* y_i = \sum_{i=L+1}^{L+2U} \mu_i y_i, \tag{28}$$

$$0 \leq \alpha_i \leq r_i, \quad i = -|\mathcal{L}_0|, \ldots, L+2U, i \neq 0,$$

$$0 \leq \alpha_i^* \leq r_i, \quad i = L+1, \ldots, L+2U,$$

$$D(\alpha_{k+L} + \alpha_{k+LU} - \alpha_{k+L}^* - \alpha_{k+LU}^*) = q_k - p_k, \tag{29}$$

where $p_k, q_k \geq 0,\ k = 1, \ldots, U$.

Hence, minimizing the objective function in Eq. (25) is equivalent to maximizing the following objective function (Vapnik, 1999):

$$\max_{\boldsymbol{\alpha}, \boldsymbol{\alpha}^*, \mathbf{p}, \mathbf{q}} -\frac{1}{2\lambda}[\boldsymbol{\alpha}; \boldsymbol{\alpha}^*]^\top \boldsymbol{\Omega}[\boldsymbol{\alpha}; \boldsymbol{\alpha}^*] + \boldsymbol{\varrho}^\top [\boldsymbol{\alpha}; \boldsymbol{\alpha}^*] - \mathbf{p}^\top \mathbf{d} \tag{30}$$

$$\text{s.t. (28)–(29), and } \mathbf{p} \geq \mathbf{0},\ \mathbf{q} \geq \mathbf{0}, \tag{31}$$

where $\boldsymbol{\Omega}$ and $\boldsymbol{\varrho}$ are defined as in Theorem 3. In Eq. (30), the variable $[\boldsymbol{\alpha}; \boldsymbol{\alpha}^*]$ is a vector consisting of an $|\mathcal{L}_0|+L+4U$ elements. The (i, j)-element of $\boldsymbol{\Lambda}_{i,j}$ is defined by $\phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$, which can be calculated by a kernel function.

In the following, we analyze the optimization in Eq. (30) on how to discard variables $\mathbf{p}$ and $\mathbf{q}$. The following are two reasons:

1. Since p_k and d_k are non-negative, in order to maximize the objective in Eq. (30), we will get $p_k d_k = 0$, for all k. In addition, from the KKT conditions, we have $p_k(1 - d_k) = 0$. Summarizing these two equalities, we obtain $p_k = 0$.
2. After p_k vanishes, adding the condition of $q_k \geq 0$, we can transform the inequality constraint of Eq. (29) to $\alpha_{k+L} + \alpha_{k+LU} - \alpha_{k+L}^* - \alpha_{k+LU}^* \geq 0$, for $k = 1, \ldots, U$.

Hence, removing the vectors $\mathbf{p}$ and $\mathbf{q}$, we can attain the QP optimization problem in Eq. (21). ■

References

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

Bennett, K. P., & Demiriz, A. (1998). Semi-supervised support vector machines. In *Advances in neural information processing systems (NIPS)* (pp. 1831–1850).

Bertsekas, D. P. (1999). *Nonlinear programming* (2nd ed.). Belmont, Massachusetts: Athena Scientific.

Bie, T. D., & Cristianini, N. (2003). Convex methods for transduction. In *Advances in neural information processing systems 16* (pp. 73–80). MIT Press.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Chapelle, O., Schölkopf, B., & Zien, A. (Eds.) (2006). *Semi-supervised learning*. Cambridge, MA: MIT Press.

Chen, X., Yang, H., King, I., & Lyu, M.R. (2015). Training-efficient feature map for shift-invariant kernels. In *IJCAI*. Buenos Aires, Argentina.

Collobert, R., Sinz, F. H., Weston, J., & Bottou, L. (2006). Large scale transductive SVMs. *Journal of Machine Learning Research*, *7*, 1687–1712.

Dehdarbehbahani, I., Shakery, A., & Faili, H. (2014). Semi-supervised word polarity identification in resource-lean languages. *Neural Networks*, *58*, 50–59.

Goldfarb, D., & Liu, S. (1991). An $o(n^3l)$ primal interior point algorithm for convex quadratic programming. *Mathematical Programming*, *49*(3), 325–340.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction* (2nd ed.). Springer.

Hu, J., Yang, H., King, I., Lyu, M.R., & So, A.M.-C. (2015). Kernelized online imbalanced learning with fixed budgets. In *AAAI*. Austin Texss, USA, Jan. 25–30.

Huang, G., Song, S., Gupta, J. N. D., & Wu, C. (2014). Semi-supervised and unsupervised extreme learning machines. *IEEE Transactions on Cybernetics*, *44*(12), 2405–2417.

Huang, K., Xu, Z., King, I., & Lyu, M. R. (2008). Semi-supervised learning from general unlabeled data. In *Proceedings of the 8th IEEE international conference on data mining, ICDM 2008, December 15–19, 2008* (pp. 273–282). Pisa, Italy.

Iosifidis, A., Tefas, A., & Pitas, I. (2014). Regularized extreme learning machine for multi-view semi-supervised action recognition. *Neurocomputing*, *145*, 250–262.

Joachims, T. (1999). Transductive inference for text classification using support vector machines. In *International conference on machine learning, ICML* (pp. 200–209). Bled, Slowenien.

Lawrence, N. D., & Jordan, M. I. (2005). Semi-supervised learning via Gaussian processes. In L. K. Saul, Y. Weiss, & L. Bottou (Eds.), *Advances in neural information processing systems 17* (pp. 753–760). Cambridge, MA: MIT Press.

Li, Y.-F., & Zhou, Z.-H. (2010). S4VM: Safe semi-supervised support vector machine. arXiv:1005.1545.

Melacci, S., & Belkin, M. (2011). Laplacian support vector machines trained in the primal. *Journal of Machine Learning Research*, *12*, 1149–1184.

Nigam, K., McCallum, A., Thrun, S., & Mitchell, T. M. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning*, *39*(2–3), 103–134.

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345–1359.

Schohn, G., & Cohn, D. (2000). Less is more: Active learning with support vector machines. In *ICML* (pp. 839–846). Morgan Kaufmann.

Schölkopf, B., & Smola, A. (2002). *Learning with kernels*. Cambridge, MA: MIT Press.

Settles, B. (2010). *Active learning literature survey. Technical Report 1648*. Computer Sciences, University of Wisconsin–Madison.

Singh, A., Nowak, R.D., & Zhu, X. (2008). Unlabeled data: Now it helps, now it doesn't. In *NIPS* (pp. 1513–1520).

Sinz, F.H., Chapelle, O., Agarwal, A., & Schölkopf, B. (2008). An analysis of inference with the universum. In *NIPS* (pp. 1369–1376).

Smola, A. J., Vishwanathan, S. V. N., & Hofmann, T. (2005). Kernel methods for missing variables. In *Proceedings of the tenth international workshop on artificial intelligence and statistics*.

Sturm, J. F. (1999). Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization Methods & Software*, *11*, 625–653.

Valizadegan, H., & Jin, R. (2006). Generalized maximum margin clustering and unsupervised kernel learning. In *Advances in neural information processing systems 19, proceedings of the twentieth annual conference on neural information processing systems, Vancouver, British Columbia, Canada, December 4–7, 2006* (pp. 1417–1424).

Vapnik, V. (1999). *The nature of statistical learning theory* (2nd ed.). New York: Springer.

Vapnik, V., & Kotz, S. (2006). *Information science and statistics, Estimation of dependences based on empirical data: empirical inference science* (2nd ed.). Secaucus, NJ, USA: Springer-Verlag New York, Inc..

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

Weston, J., Collobert, R., Sinz, F.H., Bottou, L., & Vapnik, V. (2006). Inference with the universum. In *ICML* (pp. 1009–1016).

Wolsey, L. A. (1998). *Integer programming* (1st ed.). Wiley-Interscience.

Xu, Z., Jin, R., Zhu, J., King, I., & Lyu, M.R. (2007). Efficient convex relaxation for transductive support vector machine. In *Advances in neural information processing systems 20, proceedings of the twenty-first annual conference on neural information processing systems, Vancouver, British Columbia, Canada, December 3–6, 2007*.

Yang, H., King, I., & Lyu, M.R. (2010). Multi-task learning for one-class classification. In *IJCNN, Barcelona, Spain* (pp. 1–8).

Yang, H., King, I., & Lyu, M. R. (2011). *Sparse learning under regularization framework*. LAP Lambert Academic Publishing.

Yang, H., Lyu, M. R., & King, I. (2013). Efficient online learning for multi-task feature selection. *ACM Transactions on Knowledge Discovery from Data*, *7*(2), 1–27.

Yang, H., Xu, Z., Ye, J., King, I., & Lyu, M. R. (2011). Efficient sparse generalized multiple kernel learning. *IEEE Transactions on Neural Networks*, *22*(3), 433–446.

Yang, H., Zhu, S., King, I., & Lyu, M.R. (2011). Can irrelevant data help semi-supervised learning, why and how? In *CIKM2011, Glasgow, UK* (pp. 937–946).

Yuille, A. L., & Rangarajan, A. (2003). The concave–convex procedure. *Neural Computation*, *15*(4), 915–936.

Zhang, D., Wang, J., Wang, F., & Zhang, C. (2008). Semi-supervised classification with universum. In *SDM* (pp. 323–333).

Zhao, M., Zhang, Z., Chow, T. W. S., & Li, B. (2014). A general soft label based linear discriminant analysis for semi-supervised dimensionality reduction. *Neural Networks*, *55*, 83–97.

Zhou, Z.-H., & Li, M. (2010). Semi-supervised learning by disagreement. *Knowledge and Information Systems*, *24*(3), 415–439.

Zhu, X., & Goldberg, A. B. (2009). *Introduction to semi-supervised learning*. Morgan & Claypool.