Information Sciences 314 (2015) 152–168

Contents lists available at ScienceDirect

Information Sciences

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

Laplacian unit-hyperplane learning from positive and unlabeled examples

Yuan-Hai Shao [a,*], Wei-Jie Chen [a], Li-Ming Liu [b], Nai-Yang Deng [c,*]

[a] Zhijiang College, Zhejiang University of Technology, Hangzhou 310024, PR China
[b] School of Statistics, Capital University of Economics and Business, Beijing 100070, PR China
[c] College of Science, China Agricultural University, Beijing 100083, PR China

ARTICLE INFO

Article history:
Received 10 July 2014
Received in revised form 24 March 2015
Accepted 29 March 2015
Available online 3 April 2015

Keywords:
PU learning
Biased support vector machine
One-class support vector machine
Regularization

ABSTRACT

In machine learning and data mining, learning from positive and unlabeled examples (PU learning) has attracted a great deal of attention, and the corresponding classifiers are required because of its applications in many practical areas. For PU learning, we propose a novel classifier called Laplacian unit-hyperplane classifier (LUHC), which determines a decision unit-hyperplane by solving a quadratic programming problem (QPP). The advantages of our LUHC are as follows: (1) Both geometrical and discriminant properties of the examples are exploited, resulting in better classification performance. (2) The size of QPP to be solved is small since it depends only on the number of the positive examples, resulting in faster training speed. (3) A meaningful parameter v is introduced to control the upper bounds on the fractions of positive examples with margin errors. Preliminary experiments on both synthetic and real data sets show high level of agreement with aforementioned hypothesis, suggesting that our LUHC is superior to biased support vector machine, spy-expectation maximization, and naive Bayes in both classification ability and computation efficiency.

1. Introduction

It has been observed that only positively labeled examples and unlabeled examples are available in many applications such as text classifications and information retrieval [36,44,16,15]. As a consequence, learning from positive and unlabeled examples (PU learning) has been received a great deal of attention since both supervised and semi-supervised classification algorithms are inapplicable [29,32,31,14,8,4,30].

For PU learning, several approaches have been proposed [24,18,17,7,47,45,43]. One family of these methods is related to one-class classification, which estimates the distribution of positive class from only positive examples, such as one-class support vector machine (OSVM) [34,49]. In order to train a classifier, only labeled positive examples are required. Since there is no information about the distribution of the negative examples, the number of positive examples should be sufficiently large so that the boundary of the positive class can be induced precisely. In fact, it has been proved that these methods are effective only for the case where the number of positive examples is large enough to capture the characteristics of the positive class, and their performance would be rather poor when this number is very small [25,48].

* Corresponding authors. Tel./fax: +86 057187313551.
E-mail addresses: shaoyuanhai21@163.com (Y.-H. Shao), dengnaiyang@cau.edu.cn (N.-Y. Deng).

http://dx.doi.org/10.1016/j.ins.2015.03.066
0020-0255/

Another family of these methods wishes to seek some examples from unlabeled examples that are reliable to be negative, and then apply supervised leaning to find more and more negative examples iteratively, such as expectation maximization (EM) [15,25,13,6] and positive examples-based learning (PEBL) [46,9]. The EM estimates the parameters in the negative model by a subset of unlabeled examples that are highly reliable to be negative and performs the EM operations to infer the labels of the other unlabeled examples. A drawback of EM is that it works well only when the number of positive examples is large enough because it splits the set of positive examples into a set of positive training examples. The PEBL starts from training a support vector machine (SVM) classifier with the positive examples and an initial set of negative examples. It then uses the obtained model to find more negative examples. The new negative examples are then used to train a new SVM classifier. During each iteration, new negative examples are identified from the unlabeled data set. In a word, the performance of the methods in this family depends heavily on the reliability of the extracted negative examples. However, their initial negative examples are usually identified by heuristical rules or weak classifiers. Therefore, these methods are unreliable in the general cases and their application is limited in practice.

The third family of these methods includes logistic regression and biased support vector machines (BSVM) [17,7,12]. It considers the PU learning to be supervised learning with noise [17], where all the unlabeled examples are assumed to be negative and the noisy labels are taken into consideration, by setting different weights in loss function. That is to say, the weight is large when a labeled positive example is misclassified to be negative, while the weight is small when a negative example is misclassified as positive. However, no direct way has been provided to set up the weights. Furthermore, the performance largely depends on the number of labeled examples [7].

For PU leaning, we propose a novel Laplacian unit-hyperplane classifier (LUHC) in this paper, where both positive and unlabeled examples are used. Comparing with EM where a larger proportion of negative examples in unlabeled examples is assumed and BSVM where the PU learning is treated as supervised learning with noise [17], our LUHC try to discover and employ both geometrical and discriminant properties of the training examples directly, yielding its nice performance. In addition, a meaningful parameter v is introduced to control the upper bounds on the fractions of positive examples with margin errors. Further, the main cost of our LUHC is to solve a quadratic programming problem (QPP) with a rather small size depending only on the number of the positive examples, resulting in its faster training speed, especially when the number of positive examples is small. Our LUHC has been compared with BSVM, EM, and naive Bayes by numerical experiments on both synthetic and real data sets. The preliminary results show that our LUHC is less sensitive to the proportion of labeled examples, and superior to others in both classification ability and computation efficiency.

The paper is organized as follows: Section 2 briefly dwells on the famous biased SVM. Section 3 proposes our Laplacian unit-hyperplane classifier and gives the corresponding theoretical analysis. Section 4 discusses the parameters selection of our classifier. Experimental results are described in Section 5, and at last, concluding remarks are given in Section 6.

2. Related works

In this paper, we consider PU learning, i.e. the problem of learning from positive and unlabeled examples: Suppose that, for every example $x \in \mathbb{R}^n$, there is a class output $y \in \{-1, 1\}$ corresponding to negative class or positive class. Now, we have a training set

$$T = \{(x_1, y_1), \ldots, (x_l, y_l)\} \cup \{x_{l+1}, \ldots, x_{l+q}\}, \tag{1}$$

where $x_i \in \mathbb{R}^n$ is its positive example and $y_i = 1$ is a positive output, $i = 1, \ldots, l; x_j \in \mathbb{R}^n$ is an unlabeled example known to belong to one of the two classes, $j = l+1, \ldots, l+q$. The task is to find a real function $f(x)$ in R^n such that the output value of y for any x can be predicted by the sign function of $f(x)$ as

$$g(x) = \text{sign}(f(x)). \tag{2}$$

Biased support vector machines (BSVM) [23,5] treats all the unlabeled examples as negative examples with noises; that is $\{y_i = 1 : i = \{1, \ldots, l\}\}$ and $\{y_j = -1 : j = \{l+1, \ldots, l+q\}\}$. Then, following the maximal margin principle, the SVM-type classifier is built. More precisely, searching for a hyperplane in the feature spaces

$$f(x) = (w \cdot x) + b = 0, \tag{3}$$

leads to the primal optimization problem

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}\|w\|^2 + \frac{C_+}{2}\sum_{i=1}^{l}\xi_i + \frac{C_-}{2}\sum_{i=l+1}^{l+q}\xi_i, \\ \text{s.t.} \quad & y_i((w \cdot x_i) + b) \geqslant 1 - \xi_i, \quad i = 1, \ldots, l, \ldots, l+q, \\ & \xi_i \geqslant 0, \quad i = 1, \ldots, l, \ldots, l+q, \end{aligned} \tag{4}$$

where C_+ and C_- are positive parameters. The only difference between the above problem and the one in the standard SVM is that there is two penalty parameters in the former while only one in the latter. The reason is that the examples x_j with $j = \{l+1, \ldots, l+q\}$ should be penalized slightly because they are assumed to be negative with noisy; they may be positive actually.

BSVM [23] is a state-of-the-art learning algorithm for PU learning [46,5,33,37]. However, no direct approach is provided to set up its weights up to now, and its performance is not satisfactory when the number of the labeled positive examples is small.

3. Laplacian unit-hyperplane learning

Consider the PU learning described in the beginning Section 2, where positive and unlabeled examples are included in the training set (1). The main idea of our approach is to exploit simultaneously the discriminant information and the geometric property in the training examples. More precisely, similar to BSVM, start from searching for a unit-hyperplane

$$f(x) = (w \cdot x) + b = 0. \tag{5}$$

To distinguish the positive examples x_i from the unlabeled examples, it is required the values $f(x_i) = (w \cdot x_i) + b$ with $i = 1, \ldots, l$ are as large as possible, and at the same time, the regularization term should be introduced, resulting in the primal optimization problem

$$\begin{aligned} \max_{w,b,\rho} \quad & \frac{1}{2}(\|w\|^2 + b^2) - v\rho, \\ \text{s.t.} \quad & (w \cdot x_i) + b \geqslant \rho, \quad i = 1, \ldots, l, \\ & \rho \geqslant 0, \end{aligned} \tag{6}$$

where $v \in [0, 1]$ is a positive parameter. The meaning of the parameter v and the variable ρ are closely related with the counterparts in v-SVM [35]. Note that only positive examples are appeared in (6), which is similar to One-class SVM [27], and this may leads to under-fitting [25]. Therefore, we add an extra regularization term by investigating the geometric property. Suppose that two different examples x_i and x_j are more likely belong to the same class if they have a higher similarity, implying that the difference between $f(x_i)$ and $f(x_j)$ should be smaller if the distance between x_i and x_j is shorter. This is a common hypothesis in machine learning and illustrated by a toy example in Fig. 1. There are many ways to realize this hypothesis [1,20,22,21], and the Laplacian is adopted here. Introduce the matrix $W = (W_{ij})$ defined by the k nearest neighbors or graph kernels [1]

$$W_{ij} = \begin{cases} \exp(-\|x_i - x_j\|_2^2/2\sigma^2), & \text{if } x_i,\ x_j \text{ are neighbor;} \\ 0, & \text{otherwise,} \end{cases} \tag{7}$$

where $\|x_i - x_j\|_2$ denotes the Euclidean norm in $\mathbb{R}^n$ and σ is a positive parameter. So the manifold regularization can be defined by

$$\|f\|_{\mathcal{M}}^2 = \frac{1}{(l+q)^2}\sum_{i,j=1}^{l+p} W_{ij}(f(x_i) - f(x_j))^2 = f^{\top} L f, \tag{8}$$

where f is defined by the function given by (3),

$$f = [f(x_1), \ldots, f(x_{l+p})]^{\top} = Mw + eb, \tag{9}$$

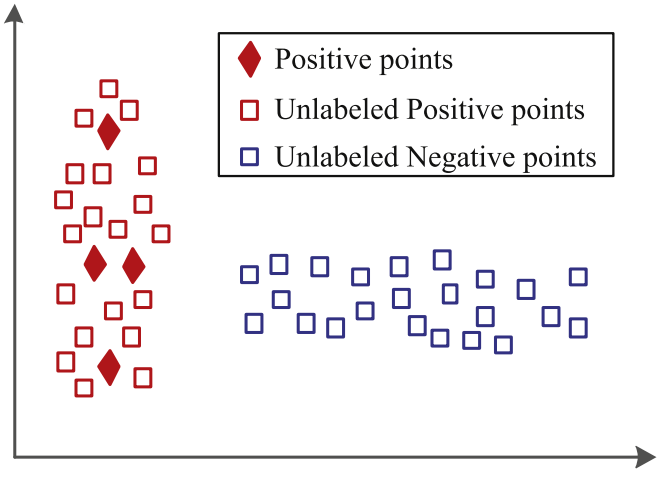


Fig. 1. A toy example.

$M \in \mathbb{R}^{(l+p)\times n}$ includes all of labeled and unlabeled examples, e is an appropriate vector of ones, $L = D - W$ is the graph Laplacian, D is a diagonal matrix with its i-th diagonal $D_{ii} = \sum_{j=1}^{l+q} W_{ij}$. Obviously, taking (8) as a regularization item realizes the aforementioned requirement: if the x_i is near to x_j, implying that they have a high similarity, the difference $|f(x_i) - f(x_j)|$ between $f(x_i)$ and $f(x_j)$ should be severely punished since W_{ij} is large, resulting in that $|f(x_i) - f(x_j)|$ is small and $f(x_i)$ is near to $f(x_j)$.

Adding the regularization term (6) into (8), we obtain the problem

$$\begin{aligned} \min_{w,b,\xi,\rho} \quad & \frac{1}{2} f^\top L f + \frac{\lambda}{2}(\|w\|^2 + b^2) - v\rho + \frac{1}{l}\sum_{i=1}^{l}\xi_i, \\ \text{s.t.} \quad & (w \cdot x_i) + b \geqslant \rho - \xi_i, \quad i = 1, \ldots, l, \\ & \rho \geqslant 0, \quad \xi_i \geqslant 0, \quad i = 1, \ldots, l, \end{aligned} \tag{10}$$

where v and λ are positive parameters. Note that f is given by (9), the above problem (10) leads to our primal problem

$$\begin{aligned} \min_{w,b,\xi,\rho} \quad & \frac{1}{2}(w^\top M^\top + e^\top b) L (Mw + eb) + \frac{\lambda}{2}(\|w\|^2 + b^2) - v\rho + \frac{1}{l} e_+^\top \xi, \\ \text{s.t.} \quad & (w \cdot X_+) + e_+ b \geqslant e_+ \rho - \xi, \\ & \xi \geqslant 0, \quad \rho \geqslant 0, \end{aligned} \tag{11}$$

where $\xi = (\xi_1, \ldots, \xi_l), X_+ = (x_1, \ldots, x_l)$, and e_+ is an appropriate ones vector.

The Lagrangian corresponding to the problem (11) is given by

$$\begin{aligned} L(w, b, \xi, \rho, \alpha, \beta, \delta) = {} & \frac{1}{2}(w^\top M^\top + e^\top b) L (Mw + eb) + \frac{\lambda}{2}(\|w\|^2 + b^2) - v\rho + \frac{1}{l} e_+^\top \xi - \alpha^\top ((w \cdot X_+) + e_+ b - e_+ \rho + \xi) \\ & - \beta^\top \xi - \delta\rho, \end{aligned} \tag{12}$$

where α, β, and δ represent the Lagrange multipliers. The Karush–Kuhn–Tucker (KKT) necessary and sufficient optimality conditions [2] are given by

$$\frac{\partial L}{\partial w} = M^\top L (Mw + eb) + \lambda w - X_+ \alpha = 0, \tag{13}$$

$$\frac{\partial L}{\partial b} = e^\top L (Mw + eb) + \lambda b - e_+ \alpha = 0, \tag{14}$$

$$\frac{\partial L}{\partial \xi} = \frac{1}{l} e_+ - \alpha - \beta = 0, \tag{15}$$

$$\frac{\partial L}{\partial \rho} = -v + e_+ \alpha - \delta = 0. \tag{16}$$

Since $v \geqslant 0$, (15) and (16) turn out to be

$$0 \leqslant \alpha \leqslant \frac{1}{l} e_+ \quad \text{and} \quad v \leqslant e_+ \alpha. \tag{17}$$

Next, combining (13) and (14) leads to

$$\begin{bmatrix} M^\top \\ e^\top \end{bmatrix} L \begin{bmatrix} M & e \end{bmatrix} \begin{bmatrix} w \\ b \end{bmatrix} + \lambda \begin{bmatrix} w \\ b \end{bmatrix} - \begin{bmatrix} X_+ \\ e_+ \end{bmatrix} \alpha = 0. \tag{18}$$

Let

$$H = \begin{bmatrix} X_+ \\ e_+ \end{bmatrix}, \quad J = \begin{bmatrix} M \\ e \end{bmatrix}, \quad u = \begin{bmatrix} w \\ b \end{bmatrix}. \tag{19}$$

Eq. (18) can be rewritten as:

$$J^\top L J u + \lambda u - H\alpha = 0 \quad \text{or} \quad u = (J^\top L J + \lambda I)^{-1} H \alpha, \tag{20}$$

since the matrix $J^\top L J + \lambda I$ is positive definite according to matrix theory [11], where I is an identity matrix of appropriate dimensions. Substituting the above equations into problem (12), we obtain the Wolfe dual of the primal problem (11)

$$\begin{aligned} \max_{\alpha} \quad & -\frac{1}{2}\alpha^\top H (J^\top L J + \lambda I)^{-1} H^\top \alpha, \\ \text{s.t.} \quad & 0 \leqslant \alpha \leqslant \frac{1}{l} e_+, \quad v \leqslant e_+ \alpha. \end{aligned} \tag{21}$$

There are several advantages in our LUHC. Firstly, in (11), both positive and unlabeled examples are inserted into the precomputable matrix L, which can capture their geometrical structure. At the same time, the positive examples are used, leading to discover both the geometrical and discriminant structure of the training samples. Secondly, a Tikhonov regularization term [5,38] is introduced, which may be interpreted as a tradeoff between empirical risk and a margin loss defined upon the energy of all training examples. Last but not least, the number of constraints in the optimization problem (11) is only related to positive examples and usually is rather small (since $l \ll q$). As we can see, problem (21) is a quadratic programming problem (QPP) with only l variables and can be solved with the computational time complexity of $O(l^3)$, where l is the number of the positive examples. In contrast, the BSVM needs to solve a QPP with the time complexity of $O((l+q)^3)$, where q is the number of the unlabeled examples. That is to say, our approach should be faster, especially when q is large.

After getting a solution α^* of (21), w^*, b^*, and ρ^* in the solution $(w^*, b^*, \rho^*, \xi^*)$ of the problem (11) can be found by the following theorem.

Theorem 1. *If α^* is the solution of (21), then $\begin{bmatrix} w^* \\ b^* \end{bmatrix} = (J^\top LJ + \lambda I)^{-1} H\alpha^*$, and if there exists a component of $\alpha^* : \alpha_j^*$, such that $\alpha_j^* \in (0, \frac{1}{l})$, then the ρ^* in the solution $(w^*, b^*, \rho^*, \xi^*)$ can be obtained by $\rho^* = (w^* \cdot x_j) + b^*$.*

Proof. From (19) and (20), we can obtain that $\begin{bmatrix} w^* \\ b^* \end{bmatrix} = (J^\top LJ + \lambda I)^{-1} H\alpha^*$. Further, from the KKT conditions, we know that if $\alpha_j^* \neq 0$, then $(w^* \cdot x_j) + b^* - \rho^* + \xi_j = 0$. While, if $\alpha_j^* \in (0, \frac{1}{l})$, from (15), we have $\beta_j \neq 0$, then $\xi_j = 0$. So, $(w^* \cdot x_j) + b^* - \rho^* = 0$. □

Ones w^* and b^* is found, a new example x can be classified as positive class or negative class by the following decision function

$$\text{Class } i = \text{sign}((w^* \cdot x) + b^* - \rho^*). \tag{22}$$

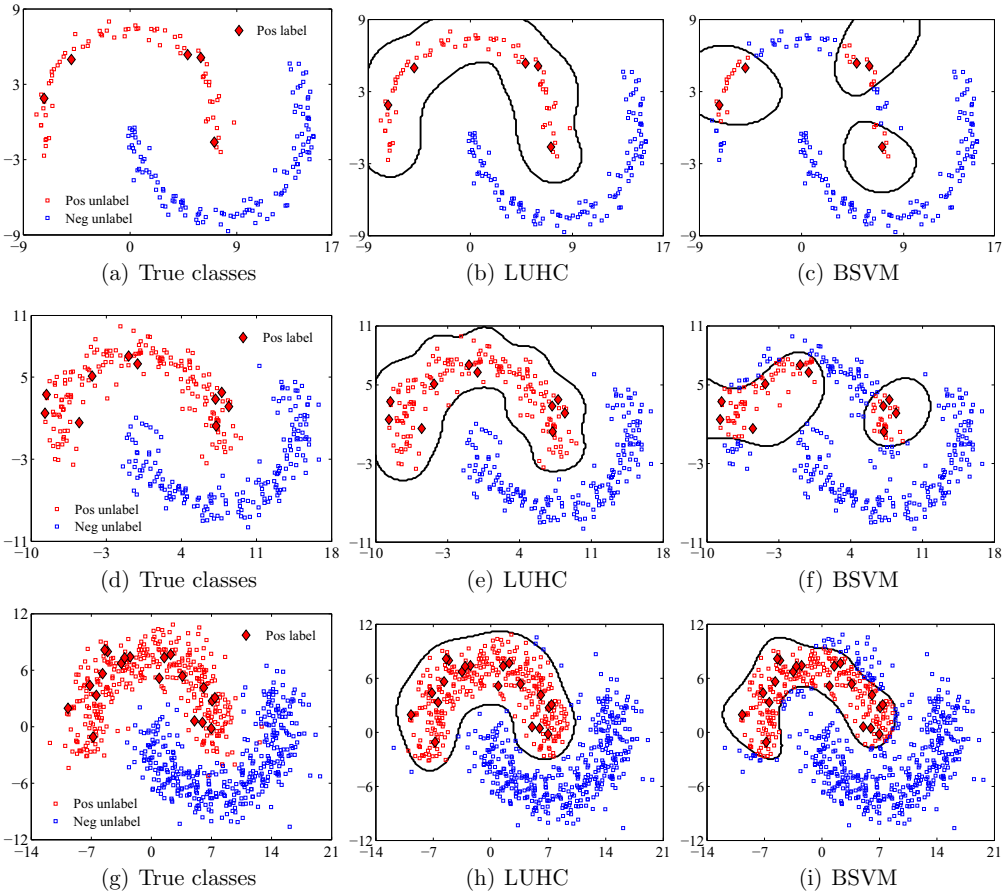


Fig. 2. Comparison results of our LUHC and BSVM on three types of artificial half-moons data sets.

Table 1
Results of our LUHC and BSVM on three types of artificial half-moons data sets. The best $\hat{F}$ value is shown by bold figure.

Data sets	Classifier	Sen (%)	Spe (%)	Acc (%)	$\hat{F}$ value	Time (s)
Type (a)	LUHC	100.00	100.00	100.00	**1.000**	0.0387
(202 × 2 5%)	BSVM	48.68	100.00	80.69	0.655	0.1029
Type (d)	LUHC	100.00	100.00	100.00	**1.000**	0.0967
(378 × 2 10%)	BSVM	53.43	100.00	76.71	0.696	0.2619
Type (g)	LUHC	95.75	99.73	97.74	**0.977**	0.1788
(754 × 2 20%)	BSVM	76.65	99.62	88.19	0.866	0.6407

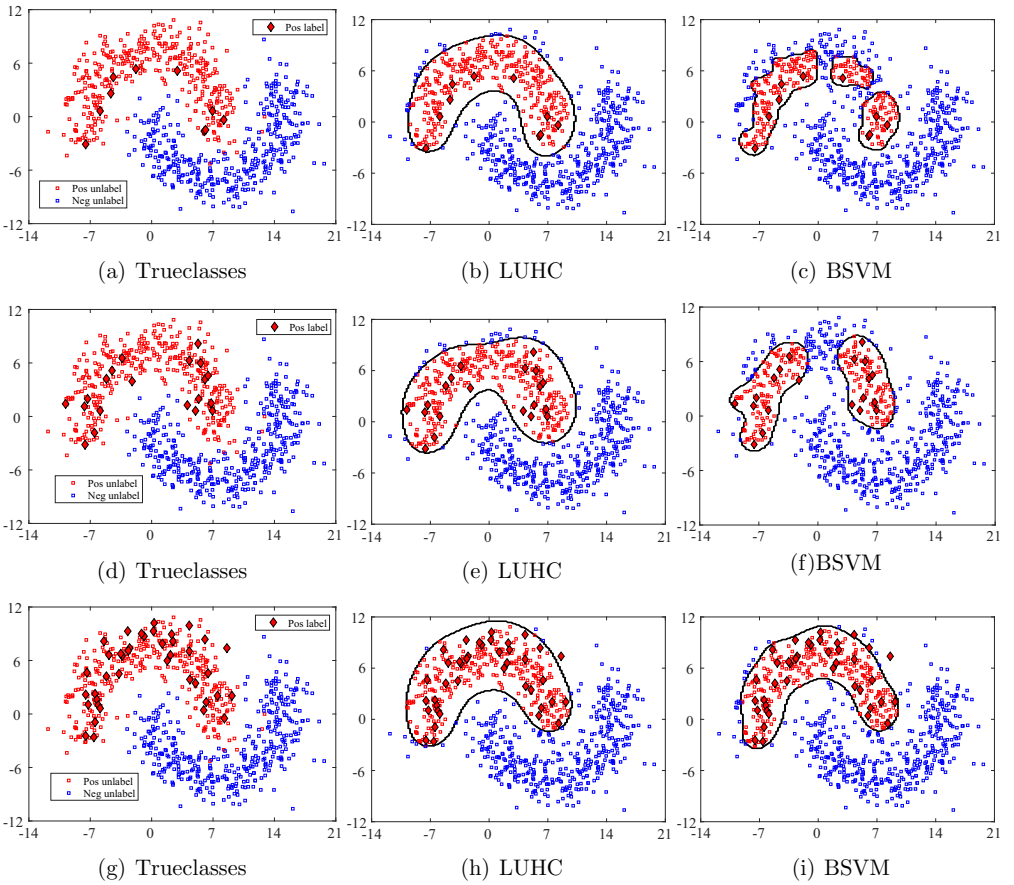


Fig. 3. Comparison results of our LUHC and BSVM on artificial half-moons data sets with different number of labeled examples.

Table 2
Results of our LUHC and BSVM on artificial half-moons data set with different number of labeled examples. The best $\hat{F}$ value is shown by bold figure.

Data sets	Classifier	Sen (%)	Spe (%)	Acc (%)	$\hat{F}$ value	Time (s)
Type (a)	LUHC	92.31	97.35	94.83	**0.947**	0.2186
(754 × 2 5%)	BSVM	46.69	100.00	73.34	0.637	0.7680
Type (d)	LUHC	90.19	99.47	94.83	**0.946**	0.1942
(754 × 2 15%)	BSVM	66.84	100.00	83.42	0.801	0.7709
Type (g)	LUHC	90.98	100.00	95.49	**0.953**	0.2308
(754 × 2 30%)	BSVM	84.88	100.00	92.44	0.918	0.8272

Table 3
Details of the UCI data sets. Example: Total number of examples, Feature: Number of features, Test: Number of test examples, Train (10%): Number of training examples, where 10% positive examples are selected as positive set and the rest as the unlabeled set, Train (20%): Number of training examples, where 20% positive examples are selected as positive set and the rest as the unlabeled set, Train (30%): Number of training examples, where 10% positive examples are selected as positive set and the rest as the unlabeled set.

Data sets	Example	Feature	Test	Train (10%)	Train (20%)	Train (30%)
Hepatitis	155	9	77	3/75	6/72	10/68
Hearts	270	14	135	15/120	30/105	45/90
CMC	1473	9	737	114/622	228/508	342/394
Ionosphere	351	34	175	13/163	25/151	38/138
Australian	690	14	345	31/314	61/284	92/253
German	1000	24	500	70/430	140/360	210/290

Table 4
Results of $LUHC_{lin}$, $LUHC_{rbf}$, $BSVM_{lin}$, $BSVM_{rbf}$, NB, and S-EM on UCI data sets at the test set with 10% of labeled positive examples, in terms of accuracy (Acc) and $\hat{F}$ value. Ave. Acc and Ave. $\hat{F}$ denotes the average accuracy and the average $\hat{F}$ value of each classifier over all data sets.

Data sets	$LUHC_{lin}$ Acc (%) / $\hat{F}$ value	$LUHC_{rbf}$ Acc (%) / $\hat{F}$ value	$BSVM_{lin}$ Acc (%) / $\hat{F}$ value	$BSVM_{rbf}$ Acc (%) / $\hat{F}$ value	NB Acc (%) / $\hat{F}$ value	S-EM Acc (%) / $\hat{F}$ value
Hepatitis	68.74	**71.76**	58.13	70.09	54.81	62.56
	0.516	0.481	0.256	0.571	0.355	0.425
Hearts	67.44	**73.01**	65.88	71.41	62.64	66.29
	0.494	0.537	0.481	0.516	0.475	**0.546**
CMC	**64.65**	63.24	52.23	46.73	53.88	57.85
	0.438	**0.482**	0.396	0.366	0.399	0.453
Ionosphere	72.15	67.35	70.15	**74.01**	68.21	64.84
	0.565	0.518	0.462	0.527	**0.569**	0.386
Australian	72.30	**76.11**	64.10	69.33	66.82	73.41
	0.598	**0.671**	0.487	0.562	0.510	0.621
German	65.20	**68.32**	66.23	63.73	58.79	66.76
	0.539	0.548	**0.578**	0.536	0.480	0.522
Ave. Acc	68.41	**69.97**	62.79	65.88	60.86	65.29
Ave. $\hat{F}$	0.525	**0.540**	0.443	0.513	0.465	0.492

Table 5
Results of $LUHC_{lin}$, $LUHC_{rbf}$, $BSVM_{lin}$, $BSVM_{rbf}$, NB, and S-EM on UCI data sets at the test set with 20% of labeled positive examples, in terms of accuracy (Acc) and $\hat{F}$ value. Ave. Acc and Ave. $\hat{F}$ denotes the average accuracy and the average $\hat{F}$ value of each classifier over all data sets.

Data sets	$LUHC_{lin}$ Acc (%) / $\hat{F}$ value	$LUHC_{rbf}$ Acc (%) / $\hat{F}$ value	$BSVM_{lin}$ Acc (%) / $\hat{F}$ value	$BSVM_{rbf}$ Acc (%) / $\hat{F}$ value	NB Acc (%) / $\hat{F}$ value	S-EM Acc (%) / $\hat{F}$ value
Hepatitis	71.90	**74.76**	68.59	72.25	60.29	69.56
	0.561	0.548	0.428	0.514	0.449	0.532
Hearts	69.79	**76.41**	70.26	72.59	64.58	71.59
	0.547	**0.592**	0.529	0.563	0.499	0.588
CMC	61.82	**65.82**	58.42	62.82	52.38	61.74
	0.468	0.514	0.426	0.479	0.384	**0.527**
Ionosphere	72.82	70.40	71.94	**73.63**	71.63	69.22
	0.584	0.533	0.486	0.549	0.544	0.501
Australian	73.69	75.81	67.35	71.42	63.53	**76.38**
	0.614	**0.656**	0.514	0.583	0.498	0.631
German	66.84	**69.84**	64.59	67.21	56.79	67.42
	0.547	**0.584**	0.520	0.561	0.474	0.550
Ave. Acc	69.48	**72.17**	66.86	69.99	61.53	69.32
Ave. $\hat{F}$	0.554	**0.571**	0.484	0.542	0.475	0.555

Noticing that the above approach is based on the Laplacian in (6) and unit-hyperplane in (5), so it is called Laplacian unit-hyperplane classifier (LUHC).

In order to extend the above results to nonlinear classifier, consider the following kernel-generated surface [5,39] instead of the hyperplane defined in (5)

$$K(x^\top, X^\top)w + b = 0, \tag{23}$$

where X is the whole training input and K is an appropriately chosen kernel. The above surface is determined by the following primal problem

$$\begin{aligned} \min_{w,b,\xi,\rho} \quad & \frac{1}{2}(w^\top M^\top + e^\top b)L(Mw + eb) + \frac{\lambda}{2}(\|w\|^2 + b^2) - \nu\rho + \frac{1}{l}e_+^\top \xi, \\ \text{s.t.} \quad & K(X_+^\top, X^\top)w + e_+ b \geqslant e_+\rho - \xi, \\ & \xi \geqslant 0, \quad \rho \geqslant 0, \end{aligned} \tag{24}$$

where $\xi = (\xi_1, \ldots, \xi_l), X_+ = (x_1, \ldots, x_l)$, and e_+ is an appropriate ones vector.

Similar to linear case, using the K.T.T. conditions, we can obtain the dual problem of (24) as

$$\begin{aligned} \max_{\alpha} \quad & -\frac{1}{2}\alpha^\top Q(J^\top LJ + \lambda I)^{-1}Q^\top \alpha, \\ \text{s.t.} \quad & 0 \leqslant \alpha \leqslant \frac{1}{l}e_+, \quad \nu \leqslant e_+\alpha, \end{aligned} \tag{25}$$

where $Q = [K(X_+^\top, X^\top), e_+]$.

By solving the (25), the optimal solution of α^* could be obtained, and the w^* and b^* could be obtain by

$$\begin{bmatrix} w^* \\ b^* \end{bmatrix} = (J^\top LJ + \lambda I)^{-1}Q\alpha^*. \tag{26}$$

Similar to linear case, if there exists a component of $\alpha^* : \alpha_j^*$, such that $\alpha_j^* \in (0, \frac{1}{l})$, then the solution ρ^* in (24) can be obtained by $\rho^* = K(x_j, X)w^* + b^*$. Ones w^* and b^* is found, a new example x can be classified as positive class or negative class by the following decision function

$$\text{Class } i = \text{sign}(K(x, X)w^* + b^* - \rho^*). \tag{27}$$

4. The selection of the parameters

In this section, we analysis the parameter selection of our LUHC. Firstly, we give a property of the parameter ν. Similar to the ν-SVM in [35,41], we define a positive training example x_i with $\alpha_i^* > 0$ as a positive support vector and define a positive training example x_i with $f(x_i) < \rho^*$ as a positive example with margin error. To theoretically analyze the proposed LUHC,

Table 6
Results of LUHC_{lin}, LUHC_{rbf}, BSVM_{lin}, BSVM_{rbf}, NB, and S-EM on UCI data sets at the test set with 30% of labeled positive examples, in terms of accuracy (Acc) and $\hat{F}$ value. Ave. Acc and Ave. $\hat{F}$ denotes the average accuracy and the average $\hat{F}$ value of each classifier over all data sets.

Data sets	LUHC_{lin} Acc (%) / $\hat{F}$ value	LUHC_{rbf} Acc (%) / $\hat{F}$ value	BSVM_{lin} Acc (%) / $\hat{F}$ value	BSVM_{rbf} Acc (%) / $\hat{F}$ value	NB Acc (%) / $\hat{F}$ value	S-EM Acc (%) / $\hat{F}$ value
Hepatitis	72.10 / 0.587	**75.28** / **0.620**	69.49 / 0.500	71.37 / 0.538	64.28 / 0.489	74.29 / 0.608
Hearts	72.31 / 0.544	75.22 / **0.607**	71.08 / 0.529	74.34 / 0.582	69.71 / 0.471	**75.92** / 0.595
CMC	63.04 / 0.475	**66.42** / **0.531**	60.17 / 0.450	64.21 / 0.592	56.12 / 0.402	63.58 / 0.514
Ionosphere	**78.52** / **0.605**	72.49 / 0.531	73.20 / 0.557	75.10 / 0.523	69.54 / 0.539	74.21 / 0.584
Australian	74.50 / 0.618	**76.25** / **0.663**	70.85 / 0.567	74.12 / 0.607	67.49 / 0.547	75.07 / 0.644
German	67.80 / 0.537	69.22 / **0.577**	65.36 / 0.538	67.93 / 0.519	58.84 / 0.486	**70.25** / 0.562
Ave. Acc / Ave. $\hat{F}$	71.38 / 0.561	**72.48** / **0.588**	68.36 / 0.524	71.18 / 0.560	64.33 / 0.489	72.22 / 0.585

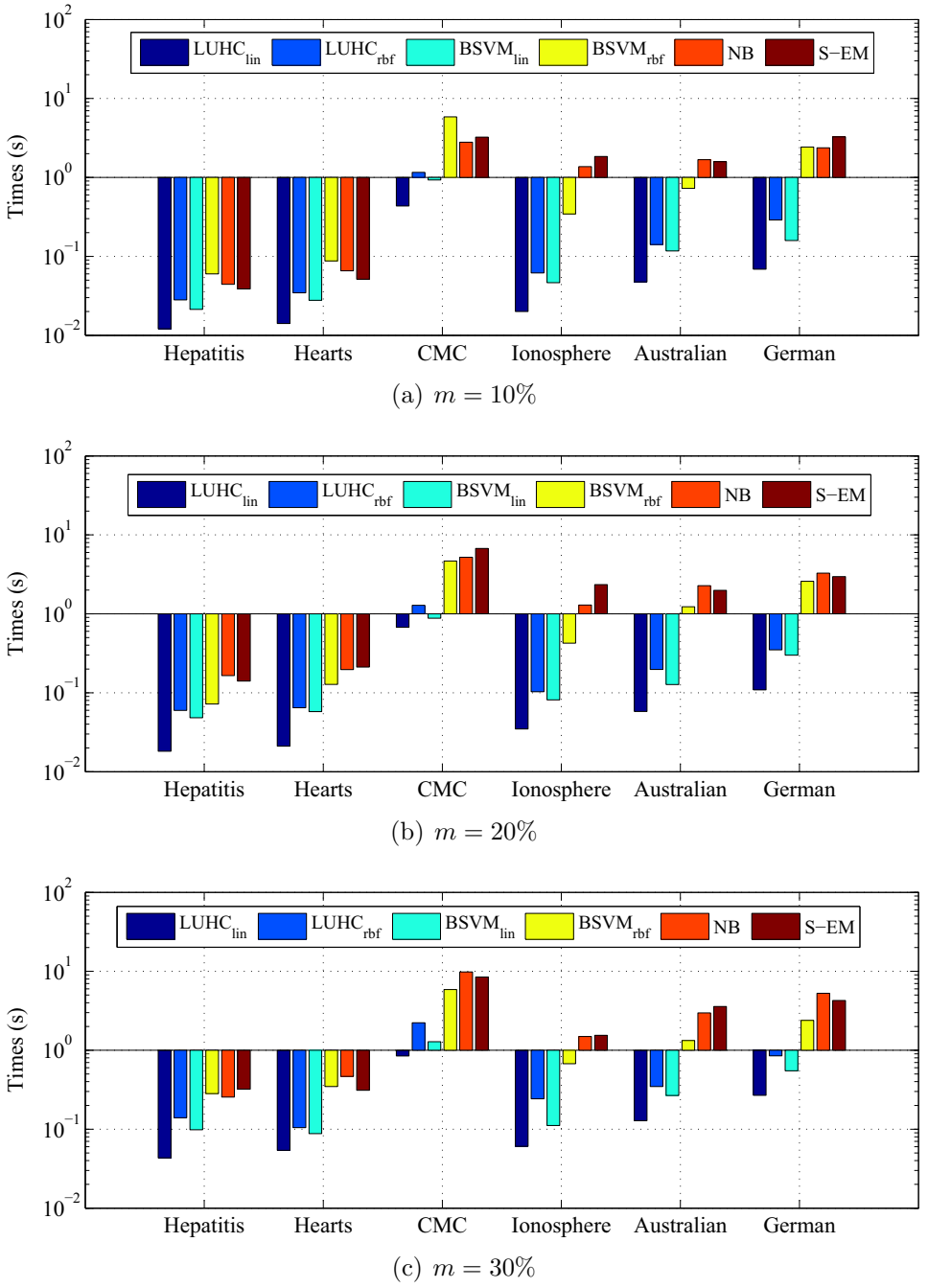


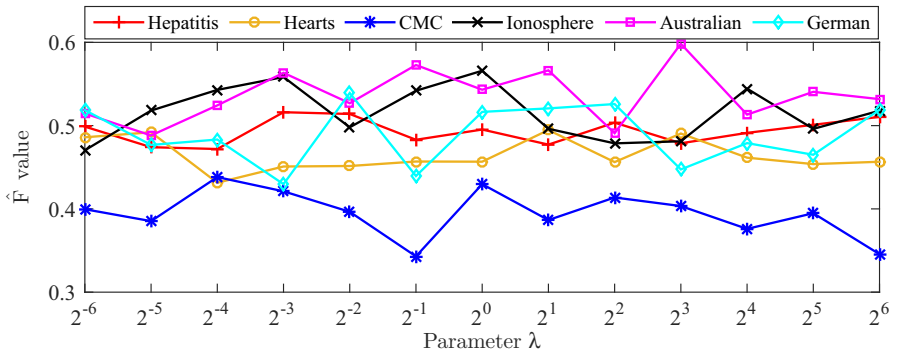
Fig. 4. The training times of LUHC$_{lin}$, LUHC$_{rbf}$, BSVM$_{lin}$, BSVM$_{rbf}$, NB, and S-EM on UCI data sets, where m is the ratio of labeled positive examples.

investigate the fractions of the positive examples with margin errors and the fractions of the positive support vectors, which are respectively defined by

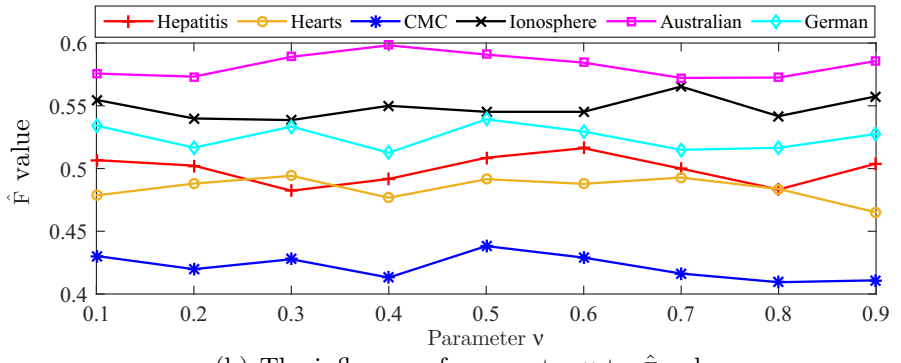
$$\frac{1}{l}|\{x_i : \alpha_i > 0, x_i \in \mathcal{X}_+\} \tag{28}$$

and

$$\frac{1}{l}|\{x_i : f(x_i) < \rho^*, x_i \in \mathcal{X}_+\}. \tag{29}$$



(a) The influence of parameter λ to $\hat{F}$ value



(b) The influence of parameter ν to $\hat{F}$ value

Fig. 5. The influence of parameter λ and v on classification performance for linear LUHC.

The following theorem gives the significance of the parameter v by showing the relationship between v and the above two fractions.

Theorem 2. *Suppose the LUHC obtains the nontrivial unit-hyperplanes, then the following statements hold:*

(i) *the value v is an upper bounds on the fractions of positive examples with margin errors;*
(ii) *the value v is a lower bounds on the fractions of positive support vectors.*

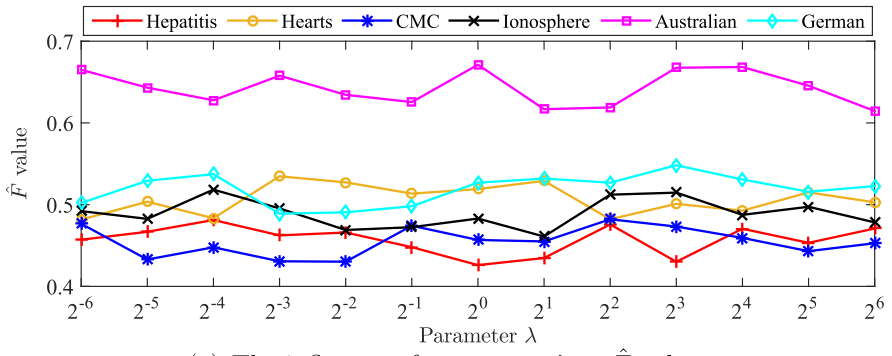
Proof.

(i) The constraints (17) imply that at most a fraction v of all training positive examples can have $\alpha_i^* = \frac{1}{l}$ All training positive examples with $\xi_i^* > 0$ certainly satisfy $\alpha_i^* = \frac{1}{l}$ (if not, α_i^* could grow further to reduce $\xi_i^* > 0$).
(ii) Positive support vectors can contribute at most $\frac{1}{l}$ to the left-hand side of (17); hence there must be at least v-th of them. □

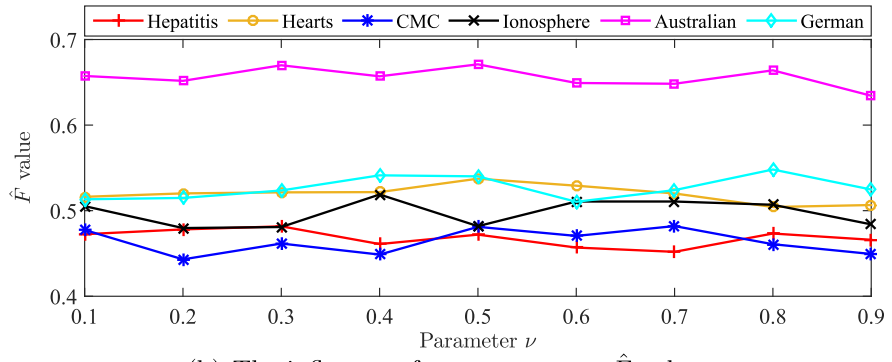
Theorem 2 is helpful to select the value of the parameter v. In practice, the common approach to select the parameters is to try a range of values of the parameters, say λ and v here, to find their optimal values, where the optimality usually depends on an evaluation measure on some testing set. Because the testing set is composed of only positive and unlabeled examples instead of positive and negative examples, there are some difficulty to find a reasonable evaluation measure. Fortunately, A nice one is proposed by

$$\hat{F}\text{ value} = \frac{\mathrm{TP}^2}{\mathrm{TP}+\mathrm{UP}} \cdot \frac{m}{(\mathrm{TP}+\mathrm{FN})^2}, \tag{30}$$

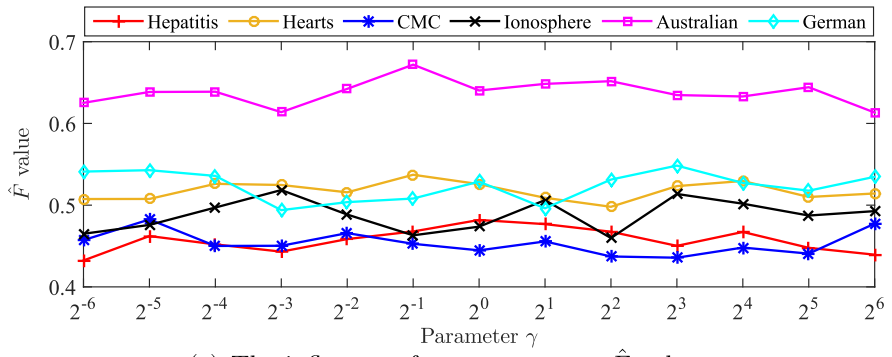
see e.g. [5], where UP (Unlabeled Positive) denotes the number of unlabeled examples classified positive and m denotes the number of testing examples. Since the values m and $(\mathrm{TP}+\mathrm{FN})$ are fixed and not dependent on the classifier, it is seen that the numerator maximizes the number of labeled positive examples to be correctly classified as positive and the denominator



(a) The influence of parameter λ to $\hat{F}$ value



(b) The influence of parameter ν to $\hat{F}$ value



(c) The influence of parameter γ to $\hat{F}$ value

Fig. 6. The influence of parameter λ, ν, and γ on classification performance for nonlinear LUHC.

minimizes the number of unlabeled examples classified as positive. Here, we point out that other evaluation measure index also promising.

Then, we give the parameter selection algorithm for our LUHC as follows:

Algorithm 1. Parameters selection for LUHC

- Set $\bar{\lambda} = \{\lambda_{min}; \ldots; \lambda_{max}\}, \bar{\nu} = \{\nu_{min}; \ldots; \nu_{max}\}$, and N_{fold}
- Partition the data set P and U into N_{fold} partitions, $Q(i) = (P(i); U(i)), i = 1; \ldots; N_{fold}$
- For each hyperparameter value $(\lambda, \nu) \in (\bar{\lambda}, \bar{\nu})$
  (1) Perform N_{fold} fold CV with the $\hat{F}$-value using the partitions $Q(i); i = 1; \ldots; N_{fold}$
  (2) Find the optimal threshold value $g^*(\lambda, \nu)$ that maximizes the average $\hat{F}$ value using the predictions
  (3) Update the best (λ, ν) value (λ^*, ν^*) that maximizes the average $\hat{F}$ value using the predictions
- Solve (21) with the entire data set $Q(i) = (P(i); U(i))$ with (λ^*, ν^*)
- Output: α^* and $g^*(\lambda^*, \nu^*)$

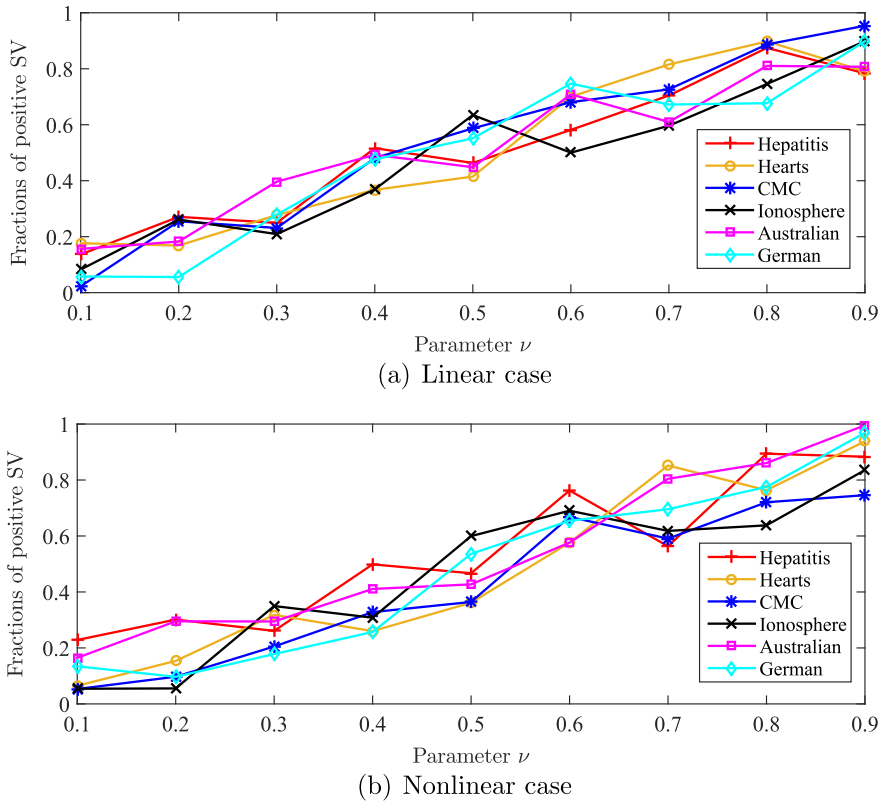


Fig. 7. The influence of parameter ν on the fractions of the positive SVs for our LUHC.

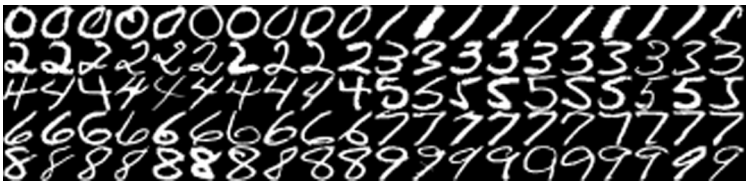


Fig. 8. An illustration of 10 subjects in the USPS database.

5. Experimental results

In this section, we evaluate the performance of our LUHC by numerical experiments which are conducted on two sets of artificial data sets, six UCI benchmark data sets [3], one USPS handwritten image data set, and two text corpora data sets. Our LUHC is compared to three classifiers: BSVM [23], naive Bayes (NB) [19], and spy-expectation maximization (S-EM) [26]. All the classifiers are implemented in MATLAB 7.0 [28] environment on a PC with Intel P4 processor (2.9 GHz) with 4 GB RAM. The parameters λ, C_1, and C_2 are chosen from the set $\{\lambda, C_1, C_2 = 2^i \mid i = -6, -5, \ldots, 5, 6\}$. And, ν is selected from the set $\{\nu = j \mid j = 0.1, 0.2, \ldots, 0.9\}$, while RBF kernel parameter γ is selected from $\{\gamma = 2^i \mid i = -6, -5, \ldots, 5, 6\}$. The final decision function is obtained by the optimal parameters selected.

5.1. Artificial data sets

Firstly, consider two types of two-dimensional synthetic "two half-moons" data sets with a few labeled positive examples. The first type of data sets are shown in Fig. 2(a), (d), and (g), where the number of examples are 202, 378, 745, and the proportions of positive examples are 5%, 10%, 20%, respectively. The second type of data sets have 745 examples, with 5%, 15%, 30% positive examples, respectively. In this experiment, our LUHC and BSVM with the RBF kernel are compared.

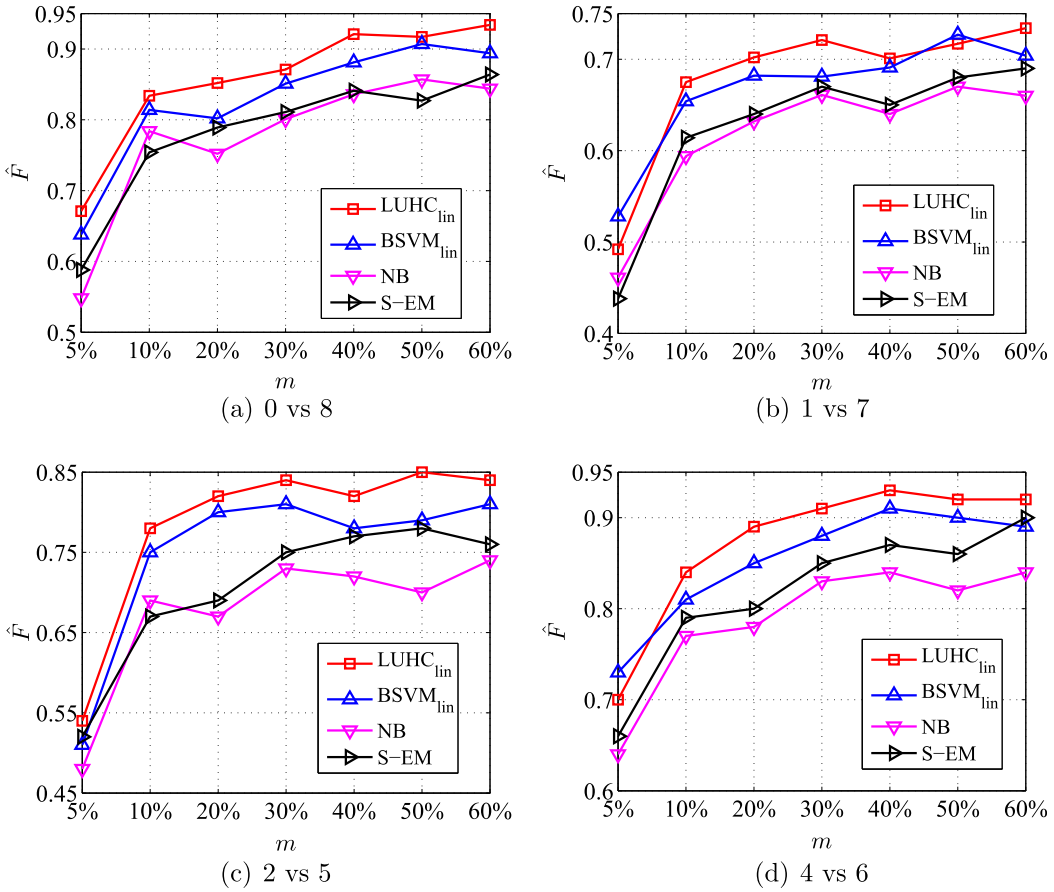


Fig. 9. The $\hat{F}$ values of LUHC$_{lin}$, BSVM$_{lin}$, NB, and S-EM on handwritten image data sets, where m is the ratio of positive labeled examples.

For the first type of data, the experimental results are summarized in Fig. 2 and Table 1. From Fig. 2, it can be seen that: (1) our LUHC always gets better classification performance than BSVM, due to its ability to hold the geometric structure of the data and (2) the number of the labeled positive examples has a stronger impact on BSVM, but weaker impact on our LUHC. In Table 1, the $\hat{F}$ value and training central processing unit (CPU) time are listed. The results in Table 1 shows that our LUHC not only obtains the better classification results than BSVM but also has faster training speed.

For the second type of data, the experimental results are summarized in Fig. 3 and Table 2. From Fig. 3, we can see that our LUHC performs much better than BSVM, especially when the number of positive examples is small. This is because our LUHC employs both the discriminate and geometric structure of the data, while BSVM neglects exploiting the geometric structure of the data and works well only when the number of positive examples is large. In comparison, the number of the labeled positive examples has a weaker impact on our LUHC, confirming the above conclusions. Further, The boundaries of both our LUHC and BSVM become smoother when the number of labeled positive examples are increasing. To see the results more clearly, we summarize the classification results of the second type data in Table 2, including the $\hat{F}$ value, accuracies, and CPU time of our LUHC and BSVM. By observing the $\hat{F}$ values and accuracies, we see that our LUHC behaves much better than BSVM.

5.2. UCI data sets

Noticing the scale of the labeled positive examples play an important role on the performance of the classifiers, it is interesting to investigate the impact of the ratio of the number of labeled positive examples to that of the total training examples. We carry out the experiments on six data sets from the UCI machine learning repository [3] summarized in Table 3. These data sets represent a wide range of fields (pathology, vehicle engineering, biological information, finance), sizes from 155 to 1473 examples and the numbers of features from 9 to 34. Similar to [40,10], our experiments are set up in the following way: first, each data set is divided into two subsets: 50% for training and 50% for testing. Then, we randomly select m of the training set as positive set, and use the remainder as unlabeled data, where m is the ratio of positive labeled data. Finally, we transform them into PU tasks. Each experiment is repeated 10 times.

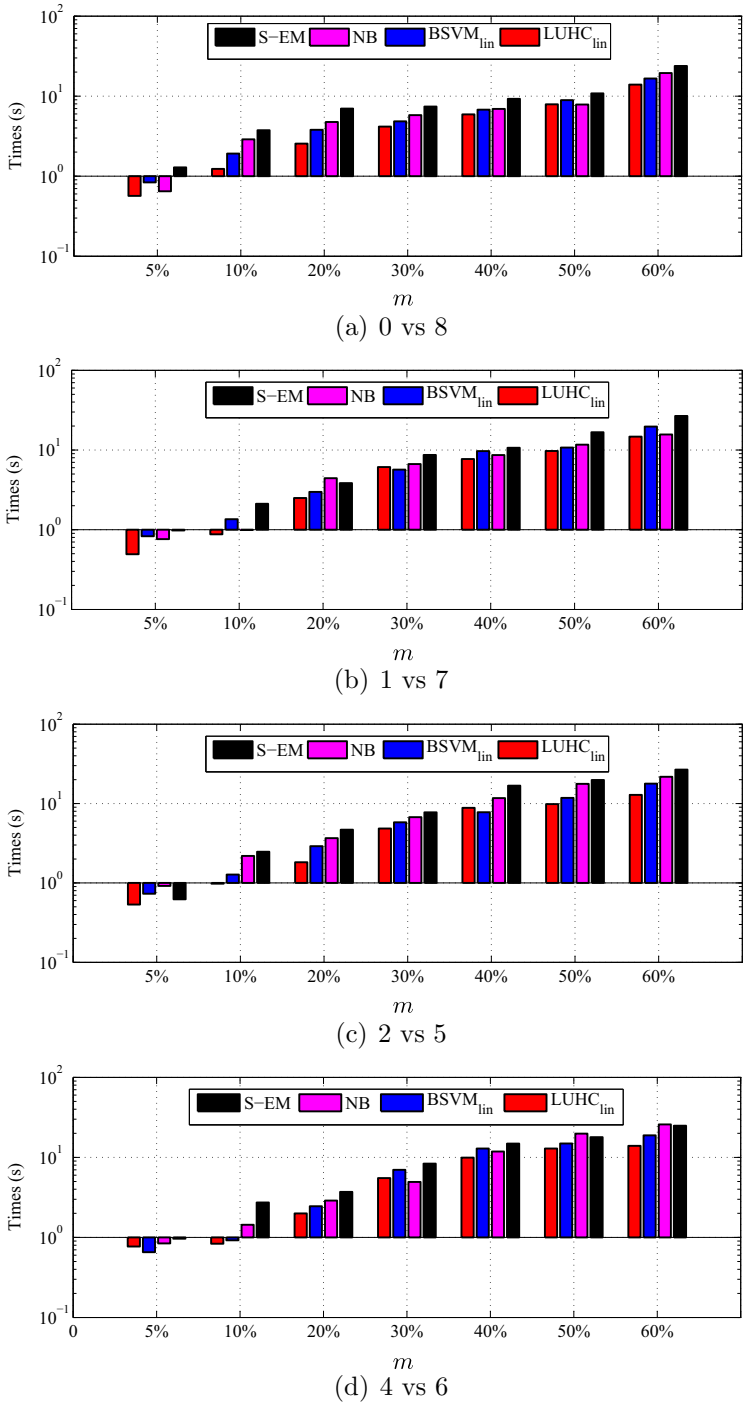


Fig. 10. The training times of LUHC$_{lin}$, BSVM$_{lin}$, NB, and S-EM on handwritten image data sets, where m is the ratio of positive labeled examples.

Tables 4–6 list the mean accuracy (Acc) and $\hat{F}$ value when 10%, 20%, and 30% ratio of the training set are selected as the labeled positive examples, respectively. The best Acc and $\hat{F}$ value are shown by bold figures. Not surprisingly, with the ratio of the labeled positive examples increasing, the Acc and $\hat{F}$ value of all classifiers are with a growing trend. Furthermore, the generalization capability of our LUHC is better than BSVM, NB, and S-EM on most of the data sets. In fact, by comparing the average of Acc and $\hat{F}$ values that are given at the bottom of Tables 4–6, it can be obtained that our LUHC owns the best performance among all compared classifiers.

We also record the computation CPU time of each classifier for the above UCI data sets experiments in Fig. 4, where the mean time is reported. From Fig. 4(a)–(c), we can see that no matter for the linear case or the kernel case, with different m, our LUHC is faster than the other methods for several orders of magnitude. This is because LUHC just solves a small scale QPP, while BSVM needs to solve a large one. To sum up, compared with BSVM, our LUHC not only has better classification ability but also spends remarkably less computational time.

Next, we analyze the influence of different parameters to the classification results for our LUHC on UCI data sets. Figs. 5 and 6 show the $\hat{F}$ value along with the variation of parameters for the linear and nonlinear cases (with 10% labeled positive examples), respectively. From Fig. 5, we obtain that both λ and v significantly influence the performance of our LUHC, and the optimal λ and v always vary for different data sets. Specially, it can be seen that too small v does not get the best performance. Compared Fig. 6 with Fig. 5, we obtain the similar conclusions on λ and v for nonlinear LUHC. As for γ, it also influences the performance of our LUHC greatly, so the proper selection of the parameters is necessary for specific data sets.

At last, we analyze the influence of different parameter v to the fractions of the positive support vectors in our LUHC. Fig. 7(a) and (b) shows the mean fractions of the positive support vectors with different v for the linear and nonlinear kernel on UCI data sets, respectively. From Fig. 7, we observe that as the increasing of v, the proportion of positive support vectors is also increasing, which confirms Theorem 2.

5.3. Handwritten image and text corpora data sets

In the following, we conduct the performance evaluation on big data, including one handwritten image data set and two text corpora data sets:

USPS: The USPS [42] database consists of grayscale handwritten digit images from 0 to 9, as shown in Fig. 8. Each digit contains 1100 images, and the size of each image is 16×16 pixels with 256 gray levels. Here we select four pairwise digits of varying difficulty for odd vs. even digit classification.

20 Newsgroups: The 20 Newsgroups is a popular data set for document classification and related tasks. It contains approximately 20,000 documents collected from 20 electronic newsgroups. The number of documents from each newsgroup is about 1000. These documents cover different topics from computer graphics, hardware to politics, sports, etc. Here we select one group as the positive examples and the others as the negative one.

Reuters: The Reuters-21578 is one of the most frequently used data sets in document classification and learning from positive and unlabeled examples. This document corpus contains 21,578 documents that are manually clustered into 135 classes based on their topics. Each document may belong to multiple topics. All documents with multiple labels are excluded from our experiments. We select the most examples of the classes as the positive examples and the others as the negative one. This data set is more difficult than the 20 Newsgroups data because the data set is imbalanced where documents in each class contain a wider range of topics.

In our experiments, the images and the texts of each project are partitioned randomly into two parts with the same sizes such that one part is selected for training and the remaining part is used for testing. This process is repeated 10 times. In addition, only the linear kernel for these classifiers is considered.

For USPS data sets, Figs. 9 and 10 show the $\hat{F}$ value and the training time with 5%, 10%, 20%, 30%, 40%, 50%, and 60% ratio of the training sets are considered as labeled positive examples, respectively. From Figs. 9 and 10, we obtain that with the ratio of the labeled positive examples increasing, the $\hat{F}$ value and time of all classifiers have a growing trend. Furthermore, the superiority of our LUHC is clear because the red lines that represents LUHC are almost always lie above the other lines that

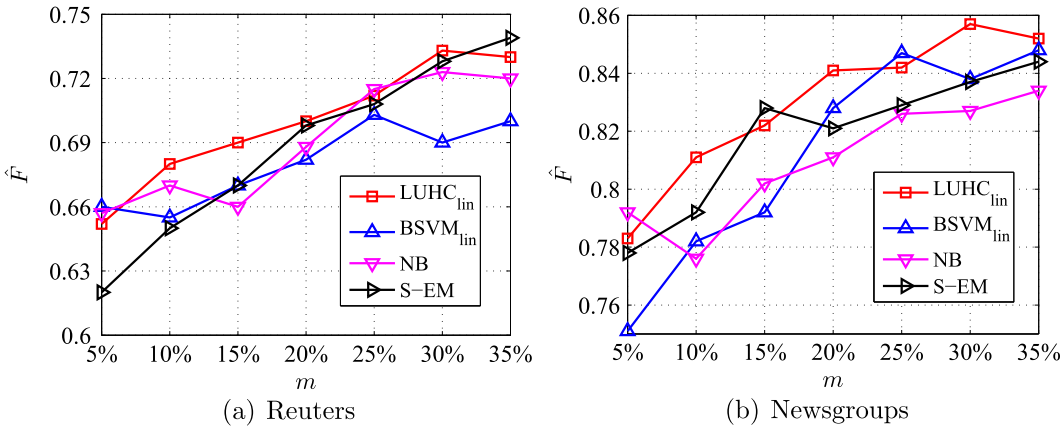


Fig. 11. The $\hat{F}$ value of LUHC$_{lin}$, BSVM$_{lin}$, NB, and S-EM on two text corpora data sets, where m is the ratio of positive labeled examples.

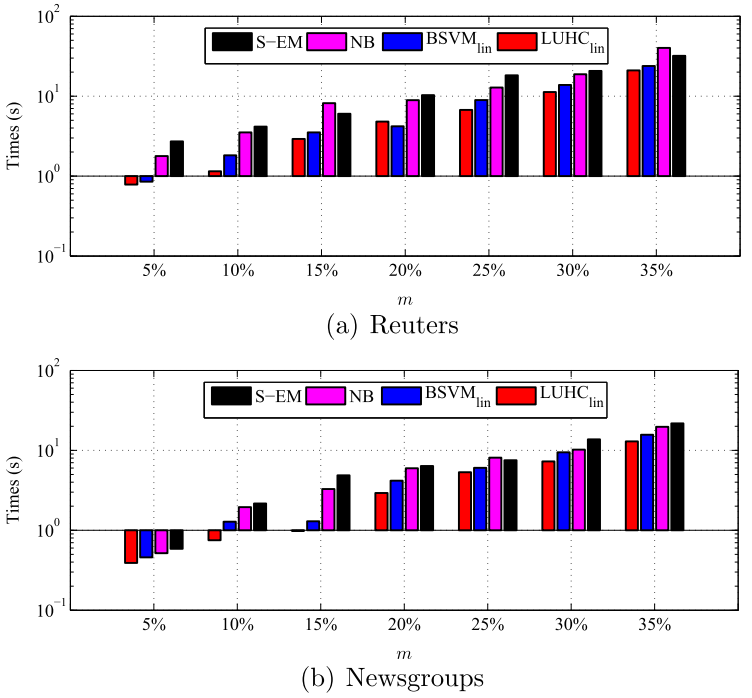


Fig. 12. The training times of LUHC$_{lin}$, BSVM$_{lin}$, NB, and S-EM on two text corpora data sets, where m is the ratio of positive labeled examples.

corresponds to the other three classifiers in Fig. 9. As for the CUP time showing in Fig. 10, we see that the red bars are always lie below 10^0 with the longest length while the other bars that correspond to the other classifiers either above the line 10^0 or below 10^0 but with short length. This demonstrates that LUHC has less training time.

For text corpora data sets, Figs. 11 and 12 show the $\hat{F}$ value and the training time with 5%, 10%, 15%, 20%, 25%, 30%, and 35% ratio of the labeled positive examples, respectively. From Figs. 11 and 12, we obtain that with the ratio of the labeled positive examples increasing, the $\hat{F}$ value and time of all classifiers have a growing trend. In summary, the generalization capability of our LUHC is better than BSVM, NB, and S-EM on most cases, while it spends lesser training time.

6. Conclusions

For PU learning, we have proposed a novel Laplacian unit-hyperplane classifier, called LUHC. In our LUHC, both the positive and unlabeled examples are used to discover the geometrical structure, while only the positive examples are used to discover the discriminant structure. This makes our LUHL have a very fast training speed since the QPP solved is rather small. Preliminary experiments on synthetic and real data sets show that our LUHC is superior to BSVM in both classification ability and computation efficiency. The corresponding LUHC Matlab codes can be downloaded from: http://www.optimal-group.org/Resource/LUHC.html. For future research, finding faster solving methods and designing sparse and robust coding in LUHC should be interesting. In addition, some different geometric regularization terms, for example related to [20], could be useful to make further improvement.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Nos. 11201426, 11426202, 11426200, and 11371365), the Zhejiang Provincial Natural Science Foundation of China (Nos. LY15F030016, LQ12A01020, LQ13F030010, and LQ14G010004), the Ministry of Education, Humanities and Social Sciences Research Project of China (No. 13YJC910011), and Beijing Natural Science Foundation (No. 9122004).

References

[1] Mikhail Belkin, Partha Niyogi, Vikas Sindhwani, Manifold regularization: a geometric framework for learning from labeled and unlabeled examples, J. Mach. Learn. Res. 7 (2006) 2399–2434.
[2] D.P. Bertsekas, Nonlinear Programming, Athena Scientific, Belmont, MA, 1995.
[3] C.L. Blake, C.J. Merz, UCI Repository for Machine Learning Databases, 1998. <http://www.ics.uci.edu/mlearn/MLRepository.html>.
[4] Olivier Chapelle, Bernhard Schölkopf, Alexander Zien, et al, Semi-supervised Learning, vol. 2, MIT press, Cambridge, 2006.

[5] N.Y. Deng, Y.J. Tian, C.H. Zhang, Support Vector Machines: Optimization-Based Theory, Algorithms, and Extensions, CRC Press, 2013.
[6] Marthinus Christoffel Du Plessis, Masashi Sugiyama, Semi-supervised learning of class balance under class-prior change by distribution matching, Neural Networks 50 (2014) 110–119.
[7] Charles Elkan, Keith Noto, Learning classifiers from only positive and unlabeled data, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008, pp. 213–220.
[8] Akinori Fujino, Naonori Ueda, Masaaki Nagata, Adaptive semi-supervised learning on labeled and unlabeled data with different distributions, Knowl. Inform. Syst. 37 (1) (2013) 129–154.
[9] Gabriel Pui Cheong Fung, Jeffrey Xu Yu, Hongjun Lu, Philip S. Yu, Text classification without negative examples revisit, IEEE Trans. Knowl. Data Eng. 18 (1) (2006) 6–20.
[10] Haitao Gan, Nong Sang, Rui Huang, Xiaojun Tong, Zhiping Dan, Using clustering analysis to improve semi-supervised classification, Neurocomputing 101 (0) (2013) 290–298.
[11] F.R. Gantmacher, Matrix Theory, Chelsea, New York, 1990.
[12] Priyanka Garg, S. Sundararajan, Active learning in partially supervised classification, in: Proceedings of the 18th ACM Conference on Information and Knowledge Management, ACM, 2009, pp. 1783–1786.
[13] David Heckerman, A Tutorial on Learning with Bayesian Networks, Springer, 1998.
[14] Ping Ji, Na Zhao, Shijie Hao, Jianguo Jiang, Automatic image annotation by semi-supervised manifold kernel density estimation, Inform. Sci. 281 (2014) 648–660.
[15] Slim Kanoun, Adel M. Alimi, Yves Lecourtier, Natural language morphology integration in off-line arabic optical text recognition, IEEE Trans. Syst., Man, Cybernet., Part B: Cybernet. 41 (2) (2011) 579–590.
[16] Maxime Latulippe, Alexandre Drouin, Philippe Giguere, François Laviolette, Accelerated robust point cloud registration in natural environments through positive and unlabeled learning, in: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, AAAI Press, 2013, pp. 2480–2487.
[17] W.S. Lee, B. Liu, Learning with positive and unlabeled examples using weighted logistic regression, in: Proc. 20th International Conference on Machine Learning, 2003, pp. 448–455.
[18] Xiao-Li Li, Bing Liu, Learning from positive and unlabeled examples with different data distributions, in: Machine Learning: ECML 2005, vol. 3720, 2005, pp. 218–229.
[19] Xiaoli Li, Bing Liu, Learning to classify texts using positive and unlabeled data, in: International Joint Conference on Artificial Intelligence, vol. 18, 2003, pp. 587–594.
[20] Zechao Li, Jing Liu, Hanqing Lu, Sparse constraint nearest neighbour selection in cross-media retrieval, in: Image Processing (ICIP), 2010 17th IEEE International Conference on, IEEE, 2010, pp. 1465–1468.
[21] Zechao Li, Jing Liu, Yi Yang, Xiaofang Zhou, Hanqing Lu, Clustering-guided sparse structural learning for unsupervised feature selection, IEEE Trans. Knowl. Data Eng. 26 (9) (2013) 2138–2150.
[22] Zechao Li, Yi Yang, Jing Liu, Xiaofang Zhou, Hanqing Lu, Unsupervised feature selection using nonnegative spectral analysis, in: Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI-12), AAAI Press, 2012, pp. 1026–1032.
[23] B. Liu, Y. Dai, X. Li, W.S. Lee, P.S. Yu, Building text classifiers using positive and unlabeled examples, in: Proc. Third IEEE Intl Conf. Data Mining, 2003, pp. 179–188.
[24] B. Liu, W.S. Lee, P.S. Yu, X. Li, Partially supervised classification of text documents, in: Proc. 19th International Conference on Machine Learning, 2002, pp. 387–394.
[25] Bing Liu, Web Data Mining, Springer, 2007.
[26] Bing Liu, Wee Sun Lee, Xiaoli Li, Partially supervised classification of text documents, in: International Conference on Machine Learning, Citeseer, 2002, pp. 8–12.
[27] Larry M. Manevitz, Malik Yousef, One-class svms for document classification, J. Mach. Learn. Res. 2 (2002) 139–154.
[28] MATLAB, The MathWorks, Inc., 2007. <http://www.mathworks.com>.
[29] Tongguang Ni, Fu-Lai Chung, Shitong Wang, Support vector machine with manifold regularization and partially labeling privacy protection, Inform. Sci. 294 (2015) 390–407.
[30] Feiping Nie, Dong Xu, Xuelong Li, Shiming Xiang, Semisupervised dimensionality reduction and classification through virtual label regression, IEEE Trans. Syst., Man, Cybernet., Part B: Cybernet. 41 (3) (2011) 675–685.
[31] K. Nigam, A.K. McCallum, S. Thrun, T.M. Mitchell, Text classification from labeled and unlabeled documents using EM, Mach. Learn. 39 (2000) 103–134.
[32] Zhibin Pan, Xinge You, Hong Chen, Dacheng Tao, Baochuan Pang, Generalization performance of magnitude-preserving semi-supervised ranking with graph-based regularization, Inform. Sci. 221 (2013) 284–296.
[33] Tao Peng, Wanli Zuo, Fengling He, SVM based adaptive learning method for text classification from positive and unlabeled documents, Knowl. Inform. Syst. 16 (3) (2008) 281–301.
[34] Bernhard Schölkopf, John C. Platt, John Shawe-Taylor, Alex J. Smola, Robert C. Williamson, Estimating the support of a high-dimensional distribution, Neural Comput. 13 (7) (2001) 1443–1471.
[35] Bernhard Schölkopf, Alex J. Smola, Robert C. Williamson, Peter L. Bartlett, New support vector algorithms, Neural Comput. 12 (5) (2000) 1207–1245.
[36] Friedhelm Schwenker, Edmondo Trentin, Pattern classification and clustering: a review of partially supervised learning approaches, Pattern Recogn. Lett. 37 (2014) 4–14.
[37] Sundararajan Sellamanickam, Priyanka Garg, Sathiya Keerthi Selvaraj, A pairwise ranking based approach to learning with positive and unlabeled examples, in: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, ACM, 2011, pp. 663–672.
[38] Y.H. Shao, C.H. Zhang, X.B. Wang, N.Y. Deng, Improvements on twin support vector machines, IEEE Trans. Neural Networks 22 (6) (2011) 962–968.
[39] Yuan-Hai Shao, Wei-Jie Chen, Nai-Yang Deng, Nonparallel hyperplane support vector machine for binary classification problems, Inform. Sci. 263 (2014) 22–35.
[40] R.G.F. Soares, Huanhuan Chen, Xin Yao, Semisupervised classification with cluster regularization, IEEE Trans. Neural Networks Learn. Syst. 23 (11) (2012) 1779–1792.
[41] Y. Tian, Z. Qi, X. Ju, Y. Shi, X. Liu, Nonparallel support vector machines for pattern classification, IEEE Trans. Cybernet. 44 (7) (2014) 1067–1079.
[42] USPS, The USPS Database, 1998. <http://www.cs.nyu.edu/roweis/data.html>.
[43] Zhijie Xu, Zhiquan Qi, Jianqin Zhang, Learning with positive and unlabeled examples using biased twin support vector machine, Neural Comput. Appl. 25 (6) (2014) 1303–1311.
[44] Peng Yang, Xiaoli Li, Hon-Nian Chua, Chee-Keong Kwoh, See-Kiong Ng, Ensemble positive unlabeled learning for disease gene identification, PloS one 9 (5) (2014) e97079.
[45] Zhi-Xia Yang, Nonparallel hyperplanes proximal classifiers based on manifold regularization for labeled and unlabeled examples, Int. J. Pattern Recognit. Artif. Intell. 27 (05) (2013).
[46] Hwanjo Yu, Jiawei Han, K.C. -C Chang, Pebl: Web page classification without negative examples, IEEE Trans. Knowl. Data Eng. 16 (1) (2004) 70–81.
[47] Joey Tianyi Zhou, Sinno Jialin Pan, Qi Mao, Ivor W. Tsang, Multi-view positive and unlabeled learning, J. Mach. Learn. Res.-Proc. Track 25 (2012) 555–570.
[48] Ke Zhou, Xue Gui-Rong, Qiang Yang, Yong Yu, Learning with positive and unlabeled examples using topic-sensitive PLSA, IEEE Trans. Knowl. Data Eng. 22 (1) (2010) 46–58.
[49] Fa Zhu, Ning Ye, Wei Yu, Sheng Xu, Guobao Li, Boundary detection and sample reduction for one-class support vector machines, Neurocomputing 123 (2014) 166–173.