SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

December 2017, Vol. 60 122103:1–122103:14
doi: 10.1007/s11432-016-9020-4

PSVM: a preference-enhanced SVM model using preference data for classification

Lerong MA[1,2], Dandan SONG[1*], Lejian LIAO[1*] & Jingang WANG[3]

[1]*Engineering Research Center of High Volume Language Information Processing and Cloud Computing Applications, School of Computer Science and Technology, Beijing Institute of Technology, Beijing* 100081*, China;*
[2]*College of Mathematics and Computer Science, Yan'an University, Yan'an* 716000*, China;*
[3]*Search Business Department Alibaba Group, Beijing* 100020*, China*

Received October 26, 2016; accepted January 21, 2017; published online July 11, 2017

Abstract Classification is an essential task in data mining, machine learning and pattern recognition areas. Conventional classification models focus on distinctive samples from different categories. There are fine-grained differences between data instances within a particular category. These differences form the preference information that is essential for human learning, and, in our view, could also be helpful for classification models. In this paper, we propose a preference-enhanced support vector machine (PSVM), that incorporates preference-pair data as a specific type of supplementary information into SVM. Additionally, we propose a two-layer heuristic sampling method to obtain effective preference-pairs, and an extended sequential minimal optimization (SMO) algorithm to fit PSVM. To evaluate our model, we use the task of knowledge base acceleration-cumulative citation recommendation (KBA-CCR) on the TREC-KBA-2012 dataset and seven other datasets from UCI, StatLib and mldata.org. The experimental results show that our proposed PSVM exhibits high performance with official evaluation metrics.

Keywords preference, SVM, classification, sampling, sequential minimal optimization (SMO)

Citation Ma L R, Song D D, Liao L J, et al. PSVM: a preference-enhanced SVM model using preference data for classification. Sci China Inf Sci, 2017, 60(12): 122103, doi: 10.1007/s11432-016-9020-4

1 Introduction

Classification is a critical task that has wide applications in data mining, machine learning, and pattern recognition. Among existing classification approaches, support vector machine (SVM) is one of the most popular and successful methods. However, an SVM classifier is ineffective in the case of complex target problems, for instance, when training data is unsatisfactory either quantitatively or qualitatively.

To overcome this problem, Vapnik et al. [1] introduced the SVM+ algorithm that uses privileged information. There has been some follow-up studies [2–4] that aim to achieve better performance in image-based object classification. In these studies, privileged information including descriptive attribute, bounding boxes, tags, facial action units, and joint positions of an image are used to provide detailed explanation of training instances. Understandably, a large number of additional efforts are required to

[*] Corresponding author (email: sdd@bit.edu.cn, Liaolj@bit.edu.cn)

obtain the privileged information, distinct from training and testing data. Consequently, the feature space related to privileged information is different from that of the training and testing data.

However, distinctions between two similar objects are important for humans to learn a concept. For example, different features of dogs and cats, such as 'woof' vs. 'meow' of their sounds, are very helpful for children to distinguish these two species. And in real data, even if two samples are from one category, they could be different in degrees such as obviousness or trustworthy score. And with the accumulation of data instances, fine-grained difference information between data samples within one category becomes available. Therefore, we believe this difference information can be helpful for classification models.

In this paper, we propose a novel classification model called the preference-enhanced support vector machine (PSVM) that incorporates the difference information obtained from training instances of one category into SVM. We refer to the difference information obtained from training instances as preference-pair data. Specifically, we explore the characteristics of training samples in the same category, and extract the preference-pairs from the training data. This is suitable for the case where order relations (e.g., preference orders or ranking orders) among instances can be obtained in the same class of training dataset in a classification task. For instance, in the knowledge base acceleration-cumulative citation recommendation (KBA-CCR) task, there are four-relevant levels between entities and documents in the dataset including Central, Relevant, Neutral and Garbage. We can easily build preference-pairs between documents with different levels for a target entity e. Alternatively, a document appearing earlier in the stream corpus is more likely to be relevant to e than the document appearing later, and these two documents can comprise a preference-pair.

To fit our PSVM model, we design an adapted sequential minimal optimization (SMO) algorithm with either one or two variables optimized at each iteration, which is an extension of the conventional SMO algorithm with two variables optimized at each iteration. Owing to the tradeoff between annotation labor and accuracy, a higher number of preference-pair data do not necessarily perform expectedly. Additionally, incorporating too many constraints into PSVM can greatly increase its computational complexity. To address this problem, we propose a two-layer heuristic sampling method to effectively select preference-pairs from training data.

We evaluate our model using the benchmark data of the knowledge base acceleration-cumulative citation recommendation (KBA-CCR) task of TREC-2012, denoted as TREC-KBA-2012, and seven other datasets acquired from UCI[1)], StatLib and mldata.org. Experimental results show that our PSVM model outperforms the reference models including SVM, RankingSVM and SVM+. It also performs better than other three top-ranked approaches in selecting central and relevant documents for the given entities on the TREC-KBA-2012 dataset.

The main contributions of this paper can be summarized as follows:

- We present a PSVM model using preference-pair data as enhancing difference information for classification (Section 3).
- We discuss the model's dual problem and provide an adapted SMO algorithm to solve the problem and increase the PSVM's efficiency (Subsection 3.3).
- We employ a two-layer heuristic sampling algorithm to obtain effective preference-pair data, whose feature space is the same as the training data (Section 4).
- Using the TREC-KBA-2012 dataset, we show that our model perform better in identifying central and relevant documents for given knowledge base entities (Section 5).

2 Related work

Our work is largely related to SVM using privileged information and selective sampling. We review the recent work related to these two tasks in the following subsections.

[1)] https://archive.ics.uci.edu/ml/datasets.html.

2.1 Augmented SVMs

SVMs have been widely studied in classification and regression from a wide variety of perspectives, for example [5–9], which has shown satisfactory performance in data mining and pattern recognition. The conventional SVM learning paradigm did not consider a teacher's role, until Vapnik formally introduced learning with a teacher in [1], which is also called learning using hidden information. Similar ideas were proposed in [1,3,4], which leverage the privileged information to enhance classification performance. The algorithm for learning using privileged information is implemented in [10].

RankingSVM is known to be the first successful method for learning to rank, and has been extensively studied since 2000 in [11–14]. RankingSVM uses ranked pairs to establish difference vectors between different ranking data, and these difference vectors comprise the training data. In [2], the authors proposed a rank transfer model for learning to rank using privileged information. The model first trains an ordinary RankingSVM using privileged information, and the resulting ranking of privileged information is then transferred into a RankingSVM on training data.

The models discussed above assume that privileged information is different from the training and testing features, as the privileged information include explanation, comments, or descriptive information about the training data. In contrast, we focus on preference information between similar objects within the same category of training data, which is directly determined on the basis of features of the training data. We then incorporate this preference information into SVM.

2.2 Selective sampling

Selective sampling, also known as active learning, has been extensively studied in the machine learning community. The active learning method only selects the most informative instances to be labelled. SVM selective sampling techniques have been developed and proven to be effective in achieving high accuracy with small training instances [15–17]. There has also been an effort to extend selective sampling for ranking [18–21]. Furthermore, data selection techniques have also been investigated to reduce the size of training data for RankingSVM. Lin et al. [22] proposed the pruned RankingSVM model to select the most informative pairs from the order-closest training data before training.

Our sampling method is different from the above mentioned work in two aspects. First, our sampling methods yield more diverse and credible sample pairs from the training data. Second, unlike active learning, which requires the learning of a function at each iteration, our sampling method selects effective preference-pairs on the basis of maximum distance between preference-pairs within a kernel feature space.

3 PSVM

In this section, we present our proposed model PSVM that involves enhancing SVM with information related to preference-pairs sampled from one classification category.

3.1 Primal and dual problems

Suppose that $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_l, y_l)\}$ is a set of training data for binary classification, where $\boldsymbol{x}_i \in R^d$, d is the dimension of input feature space, $y_i \in \{1, -1\}$ and $i = 1, \ldots, l$. $P = \{(\boldsymbol{x}_j^{(1)} - \boldsymbol{x}_j^{(2)}, +1) | \boldsymbol{x}_j^{(1)} \succ \boldsymbol{x}_j^{(2)}\}_{j=1}^{n}$ is a set of difference information of n preference-pairs, where $\boldsymbol{x}_j^{(1)}$ and $\boldsymbol{x}_j^{(2)}$ belong to the training data within the same class.

The basic procedure to formulate the PSVM is to add constraints for the set of difference information of preference-pairs P to the SVM model, which has the same constraints as those of the RankingSVM. We then integrate additional loss term of constraints into the loss function of SVM. From this, we obtain the following primal form for PSVM:

$$
\begin{aligned}
&\min_{\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{\xi}^*} \frac{1}{2} \parallel \boldsymbol{w} \parallel^2 + C\sum_{i=1}^{l}\xi_i + C'\sum_{j=1}^{n}\xi_j^* \\
&\text{s.t.} \ \ y_i(\langle \boldsymbol{w}\cdot\phi(\boldsymbol{x}_i)\rangle + b) \geqslant 1-\xi_i, \\
&\qquad \xi_i \geqslant 0, \quad i=1,\ldots,l, \\
&\qquad \langle \boldsymbol{w}\cdot\phi(\boldsymbol{x}_j^{(1)}-\boldsymbol{x}_j^{(2)})\rangle \geqslant 1-\xi_j^*, \\
&\qquad \xi_j^* \geqslant 0, \quad j=1,\ldots,n,
\end{aligned}
\tag{1}
$$

where $C' > 0$ is an additional parameter to control the importance of the constraints of preference-pair data.

The dual problem of (1) can be derived using standard Lagrangian techniques. Let $\alpha_i \geqslant 0, \gamma_i \geqslant 0, \alpha_j^* \geqslant 0$, and $\eta_j \geqslant 0$ be the Lagrangian multipliers for the inequalities in (1). The compact form of the dual problem of PSVM is represented in the following equations:

$$
\begin{aligned}
&\min_{\boldsymbol{\alpha}} \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}Q\boldsymbol{\alpha} - \boldsymbol{e}^{\mathrm{T}}\boldsymbol{\alpha} \\
&\text{s.t.} \ \ 0 \leqslant \alpha_i \leqslant C, \quad i=1,2,\ldots,l, \\
&\qquad 0 \leqslant \alpha_i \leqslant C', \quad i=l+1,\ldots,l+n, \\
&\qquad \sum_{i=1}^{l}\alpha_i y_i = 0,
\end{aligned}
\tag{2}
$$

where $\boldsymbol{e}$ is a column vector whose element is 1, Q is an $l+n$ by $l+n$ positive semidefinite matrix, and Q_{ij} is defined as follows:

$$
Q_{ij} = \begin{cases}
y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j), & \text{if } i \leqslant l, j \leqslant l, \\
y_i K(\boldsymbol{x}_i, \boldsymbol{x}_j^{(1)} - \boldsymbol{x}_j^{(2)}), & \text{if } i \leqslant l, \ l < j \leqslant l+n, \\
y_j K(\boldsymbol{x}_i^{(1)} - \boldsymbol{x}_i^{(2)}, \boldsymbol{x}_j), & \text{if } l < i \leqslant l+n, \ j \leqslant l, \\
K(\boldsymbol{x}_i^{(1)} - \boldsymbol{x}_i^{(2)}, \boldsymbol{x}_j^{(1)} - \boldsymbol{x}_j^{(2)}), & \text{if } l < j(i) \leqslant l+n,
\end{cases}
$$

where $K(x_i, x_j) \equiv \phi(x_i)^{\mathrm{T}}\phi(x_j)$ is a kernel function.

The number of kernel evaluations required to solve the dual problem (2) is $O((l+n)^2)$, where n is the number of preference-pairs. Thus, reducing the number of preference-pairs can significantly reduce cost. Obviously the dual problem (2) is a convex quadratic problem. Once $\boldsymbol{\alpha}$ is obtained by solving this problem, the decision function for a new input vector $\boldsymbol{x}$ can be computed using the following equation:

$$
\operatorname{sgn}(\boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x})+b) = \operatorname{sgn}\left(\sum_{i=1}^{l} y_i\alpha_i K(\boldsymbol{x}_i,\boldsymbol{x}) + b + \sum_{j=1}^{n}\alpha_j^* K(\boldsymbol{x}_j^{(1)} - \boldsymbol{x}_j^{(2)}, \boldsymbol{x})\right). \tag{3}
$$

The determination of parameter b will be addressed in the next subsection. According to the representation theorem [23], the decision function of PSVM is a linear combination of terms of the SVM and the preference-pairs.

3.2 Optimality condition for dual problem

To derive appropriate stopping conditions for algorithms that solve the equivalent dual problem (2) and also determine the threshold b, it is important to define the optimality conditions for the problem (2). Let $f(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}Q\boldsymbol{\alpha} - \boldsymbol{e}^{\mathrm{T}}\boldsymbol{\alpha}$ be a function corresponding to $\boldsymbol{\alpha}$. The Lagrangian for the dual problem (2) can be expressed as follows:

$$
L_d = f(\boldsymbol{\alpha}) + b\sum_{i=1}^{l}\alpha_i y_i - \sum_{i=1}^{l+n}\lambda_i\alpha_i - \sum_{i=1}^{l}\xi_i(C-\alpha_i) - \sum_{i=l+1}^{l+n}\eta_i(C'-\alpha_i), \tag{4}
$$

where the Lagrangian multipliers λ_i, ξ_i and η_i are non-negative, while b can take any value. The Karush-Kuhn-Tucker (KKT) conditions related to $\boldsymbol{\alpha}$ can be defined as follows:

$$\begin{gathered}\frac{\partial L_d}{\partial \alpha_i} = \triangledown_i f(\boldsymbol{\alpha}) + by_i - \lambda_i + \xi_i = 0, \\ \lambda_i \alpha_i = 0, \quad \xi_i(C - \alpha_i) = 0, \quad i = 1, \ldots, l,\end{gathered} \tag{5}$$

$$\begin{gathered}\frac{\partial L_d}{\partial \alpha_i} = \triangledown_i f(\boldsymbol{\alpha}) - \lambda_i + \eta_i = 0, \\ \lambda_i \alpha_i = 0, \quad \eta_i(C' - \alpha_i) = 0, \quad i = l+1, \ldots, l+n,\end{gathered} \tag{6}$$

where $\triangledown f(\boldsymbol{\alpha}) \equiv \boldsymbol{Q\alpha} - \boldsymbol{e}$ is the gradient of $f(\boldsymbol{\alpha})$, while $\triangledown_i f(\boldsymbol{\alpha}) = [\boldsymbol{Q\alpha} - \boldsymbol{e}]_i$ is the partial derivative of $f(\boldsymbol{\alpha})$ associated with α_i, and where $[\boldsymbol{Q\alpha} - \boldsymbol{e}]_i$ denotes the i-th element of $\boldsymbol{Q\alpha} - \boldsymbol{e}$. The condition (5) can be rewritten as

$$\triangledown_i f(\boldsymbol{\alpha}) + by_i = \begin{cases} -\xi_i \leqslant 0, & \text{if} \quad \alpha_i > 0, \\ \lambda_i \geqslant 0, & \text{if} \quad \alpha_i < C, \end{cases} \tag{7}$$

where i runs over $1, \ldots, l$. Since $y_i = \pm 1$, condition (7) is equivalent to the condition that there exists a b such that

$$m^c(\boldsymbol{\alpha}) \leqslant b \leqslant M^c(\boldsymbol{\alpha}), \tag{8}$$

where

$$m^c(\boldsymbol{\alpha}) \equiv \max_{i \in I^c_{\text{up}}(\boldsymbol{\alpha})} \left\{-y_i \triangledown_i f(\boldsymbol{\alpha})\right\},$$

$$M^c(\boldsymbol{\alpha}) \equiv \min_{i \in I^c_{\text{low}}(\boldsymbol{\alpha})} \left\{-y_i \triangledown_i f(\boldsymbol{\alpha})\right\},$$

$$I^c_{\text{up}}(\boldsymbol{\alpha}) \equiv \left\{i | (y_i = +1, \alpha_i < C) \text{ or } (y_i = -1, \alpha_i > 0); i = 1, \ldots, l\right\},$$

and

$$I^c_{\text{low}}(\boldsymbol{\alpha}) \equiv \left\{i | (y_i = +1, \alpha_i > 0) \text{ or } (y_i = -1, \alpha_i < C); i = 1, \ldots, l\right\}.$$

The condition (7) can also be rewritten as

$$\triangledown_i f(\boldsymbol{\alpha}) = \begin{cases} -\eta_i \leqslant 0, & \text{if} \quad \alpha_i > 0, \\ \lambda_i \geqslant 0, & \text{if} \quad \alpha_i < C', \end{cases} \tag{9}$$

where $i = l+1, \ldots, l+n$. The condition (9) is equivalent to the following inequalities:

$$m^r(\boldsymbol{\alpha}) \leqslant 0 \text{ and } M^r(\boldsymbol{\alpha}) \geqslant 0, \tag{10}$$

where $m^r(\boldsymbol{\alpha}) \equiv \max_{i \in I^r_{\text{up}}(\boldsymbol{\alpha})} \left\{\triangledown_i f(\boldsymbol{\alpha})\right\}, M^r(\boldsymbol{\alpha}) \equiv \min_{i \in I^r_{\text{low}}(\boldsymbol{\alpha})} \left\{\triangledown_i f(\boldsymbol{\alpha})\right\}$, $I^r_{\text{up}}(\boldsymbol{\alpha}) \equiv \{i | \alpha_i > 0, i = l+1, \ldots, l+n\}$ and $I^r_{\text{low}}(\boldsymbol{\alpha}) \equiv \{i | \alpha_i < C', i = l+1, \ldots, l+n\}$. A feasible $\boldsymbol{\alpha}$ is a stationary point of problem (2) if and only if

$$m^c(\boldsymbol{\alpha}) \leqslant M^c(\boldsymbol{\alpha}), \; m^r(\boldsymbol{\alpha}) \leqslant 0 \; \text{ and } \; M^r(\boldsymbol{\alpha}) \geqslant 0. \tag{11}$$

From (11), we obtain a suitable stopping condition as follows:

$$m^c(\boldsymbol{\alpha}) - M^c(\boldsymbol{\alpha}) \leqslant \epsilon, \; m^r(\boldsymbol{\alpha}) \leqslant \frac{1}{2}\epsilon \; \text{ and } \; M^r(\boldsymbol{\alpha}) \geqslant -\frac{1}{2}\epsilon, \tag{12}$$

where ϵ is the tolerance, usually 0.0001. If there exists α_i such that $0 < \alpha_i < C$ where i is between 1 and l, then from the KKT condition (7), $b = -y_i \triangledown_i f(\boldsymbol{\alpha})$.

3.3 Extended SMO algorithm

In this subsection, we extend the SMO algorithm that was first introduced in [24]. The key idea involves beginning with a valid initial point, selecting a working set including two variables using second order information [25], and optimizing a sub-problem of two variables with constraints. The sub-problem can be solved analytically. As is known in (2), there are two different types of constraints. One is linear equations and box constraints for training instances, and the other is only box constraints for preference-pairs. Therefore, we need to consider both types of constraints. Unlike selecting a working set including two variables as in [25], our working set includes either two or one variables corresponding to either the first type of constraints or the second type. As a result, the sub-problem has either two variables or one variable, and the sub-problem can be solved analytically. The following are our working set selection algorithm and its corresponding sub-problems.

Working set selection algorithm.

(1) For all $t, s \in \{1, 2, \ldots, l\}$, define

$$a_{ts} \equiv K_{tt} + K_{ss} - 2K_{ts}{}^{1)}, \quad b_{ts} \equiv -y_t \bigtriangledown_t f(\boldsymbol{\alpha}^k) + y_s \bigtriangledown_s f(\boldsymbol{\alpha}^k) > 0, \tag{13}$$

and

$$\bar{a}_{ts} \equiv \begin{cases} a_{ts}, & \text{if} \quad a_{ts} > 0, \\ \tau, & \text{otherwise,} \end{cases}$$

where τ is a small positive constant.

Find

$$\begin{aligned} i &\in \underset{t}{\operatorname{argmax}} \left\{ -y_t \bigtriangledown_t f(\boldsymbol{\alpha}^k) | t \in I^c_{\text{up}}(\boldsymbol{\alpha}^k) \right\}, \\ j &\in \underset{t}{\operatorname{argmin}} \left\{ -\frac{b_{it}^2}{\bar{a}_{it}} | t \in I^c_{\text{low}}(\boldsymbol{\alpha}^k), -y_t \bigtriangledown_t f(\boldsymbol{\alpha}^k) < -y_i \bigtriangledown_i f(\boldsymbol{\alpha}^k) \right\}. \end{aligned} \tag{14}$$

Let $B_c = \{i, j\}$.

(2) For all $t \in \{l+1, \ldots, l+n\}$, define

$$\begin{aligned} \bar{a}_t &\equiv \begin{cases} Q_{tt}, & \text{if} \quad Q_{tt} > 0, \\ \tau, & \text{otherwise,} \end{cases} \\ V_R &\equiv \{t | t \in I^r_{\text{up}}(\boldsymbol{\alpha}^k), \bigtriangledown_t f(\boldsymbol{\alpha}^k) > 0 \text{ or } t \in I^r_{\text{low}}(\boldsymbol{\alpha}^k), \bigtriangledown_t f(\boldsymbol{\alpha}^k) < 0\}. \end{aligned} \tag{15}$$

Select

$$h \in \underset{t}{\operatorname{argmin}} \left\{ -\frac{[\bigtriangledown_t f(\boldsymbol{\alpha}^k)]^2}{\bar{a}_t} | t \in V_R \right\}. \tag{16}$$

Let $B_r = \{h\}$.

(3) Check $-\frac{b_{ij}^2}{\bar{a}_{ij}}$ and $-\frac{[\bigtriangledown_h f(\boldsymbol{\alpha}^k)]^2}{\bar{a}_h}$, and let $B = B_c$ or B_r depending on which is smaller of the two values.

(4) Return B.

Define $N \equiv \{1, \ldots, l+n\} \setminus B$. Let $\boldsymbol{\alpha}_B^k$ and $\boldsymbol{\alpha}_N^k$ be sub-vectors of $\boldsymbol{\alpha}^k$ corresponding to B and N, respectively. When the working set return $B_c = \{i, j\}$ and $a_{ij} > 0$, solve the following sub-problem with two variables $\boldsymbol{\alpha}_B = [\alpha_i, \alpha_j]^{\mathrm{T}}$:

$$\begin{aligned} \min_{\alpha_i, \alpha_j} \frac{1}{2} [\alpha_i, \alpha_j] \begin{bmatrix} Q_{ii} & Q_{ij} \\ Q_{ij} & Q_{jj} \end{bmatrix} \begin{bmatrix} \alpha_i \\ \alpha_j \end{bmatrix} + (-\boldsymbol{e}_B + Q_{BN} \boldsymbol{\alpha}_N^k)^{\mathrm{T}} \begin{bmatrix} \alpha_i \\ \alpha_j \end{bmatrix} \\ \text{s.t.} \quad 0 \leqslant \alpha_i, \alpha_j \leqslant C, y_i \alpha_i + y_j \alpha_j = -\boldsymbol{y}_N^{\mathrm{T}} \boldsymbol{\alpha}_N^k. \end{aligned} \tag{17}$$

1) $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $K(\boldsymbol{x}_i^{(1)} - \boldsymbol{x}_i^{(2)}, \boldsymbol{x}_j^{(1)} - \boldsymbol{x}_j^{(2)})$ abbreviate to K_{ij}.

If the working set returns $B_c = \{i, j\}$ and $a_{ij} \leqslant 0$, solve the following sub-problem with two variables $\boldsymbol{\alpha}_B = [\alpha_i, \alpha_j]^{\mathrm{T}}$:

$$\min_{\alpha_i,\alpha_j} \frac{1}{2}[\alpha_i, \alpha_j] \begin{bmatrix} Q_{ii} & Q_{ij} \\ Q_{ij} & Q_{jj} \end{bmatrix} \begin{bmatrix} \alpha_i \\ \alpha_j \end{bmatrix} + (-\boldsymbol{e}_B + Q_{BN}\boldsymbol{\alpha}_N^k)^{\mathrm{T}} \begin{bmatrix} \alpha_i \\ \alpha_j \end{bmatrix} + \frac{\tau - a_{ij}}{4}((\alpha_i - \alpha_i^k)^2 + (\alpha_j - \alpha_j^k)^2) \\ \text{s.t.} \quad 0 \leqslant \alpha_i, \alpha_j \leqslant C,\ y_i\alpha_i + y_j\alpha_j = -\boldsymbol{y}_N^{\mathrm{T}}\boldsymbol{\alpha}_N^k. \tag{18}$$

Similarly, when the working set returns $B_r = \{h\}$ and $Q_{hh} > 0$, solve the following sub-problem with only one variable α_h:

$$\min_{\alpha_h} \frac{1}{2}\alpha_h Q_{hh}\alpha_h + (-1 + Q_{hN}\boldsymbol{\alpha}_N^k)\alpha_h \quad \text{s.t.} \quad 0 \leqslant \alpha_h \leqslant C'. \tag{19}$$

If the working set returns $B_r = \{h\}$ and $Q_{hh} \leqslant 0$, then solve the following sub-problem with only one variable α_h:

$$\min_{\alpha_h} \frac{1}{2}\alpha_h Q_{hh}\alpha_h + (-1 + Q_{hN}\boldsymbol{\alpha}_N^k)\alpha_h + \frac{\tau - Q_{hh}}{2}(\alpha_h - \alpha_h^k)^2 \quad \text{s.t.} \quad 0 \leqslant \alpha_h \leqslant C'. \tag{20}$$

4 Two-layer heuristic sampling algorithm

Preference-pairs can be selected from training data in the same category using some criteria to distinguish them. Given a knowledge base entity, for instance, an earlier occurring document related to the entity is preferred to a later occurring one, or higher ranked documents are preferred to lower ranked ones. From the dual problem (2), the kernel evaluation cost is $O((l+n)^2)$, causing the number of preference-pairs n being always too huge to solve the problem. Moreover, when a large number of preference-pairs are available, there would be more noise among them. Thus, more preference-pairs do not help improve the model's accuracy. Therefore, wisely selected preference-pairs could improve the model's performance with manageable computational complexity.

In this section, we propose a two-layer heuristic sampling method to select preference-pairs. The first layer samples more diverse data from the same rank. The Euclidean distance between two vectors in the kernel space with the RBF kernel function is determined as follows:

$$\parallel \phi(\boldsymbol{x}_i) - \phi(\boldsymbol{x}_j) \parallel^2 = 2 - 2K(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{21}$$

We build a graph whose vertices denote training samples from the same rank within a category, and weights of edges between vertices are calculated from (21). We then employ the maximum spanning tree algorithm to generate a maximum spanning tree, and obtain the diversity samples corresponding to the top m edges of the maximum spanning tree. We refer this process as diversity maximum spanning tree (DIV-MST) algorithm. This strategy is rational because it can avoid biased sampling. The complexity of the DIV-MST is $O(l^2)$.

Figure 1 illustrates the DIV-MST algorithm based on some toy data. The graph is built by three clusters, where the set of vertices consists of six data points. The weights of edges are also shown in Figure 1. After the maximum spanning tree of the graph is generated (as shown in the middle of Figure 1), we select the highest two connected edges (edges between 1, 3 and 1, 4) in terms of weight, and choose the corresponding three data points (vertices 1, 3, and 4) from the three clusters to avoid biased sampling.

In the second layer, to effectively sample preference-pairs between different ranks within the same category, we investigate the following heuristic example.

Example 1. Assume the decision function is $f(\boldsymbol{x}) = w^{\mathrm{T}}\phi(\boldsymbol{x}) + b$, $\boldsymbol{x}_1 \succ \boldsymbol{x}_2$ is a preference-pair, and let $d(\boldsymbol{x}_1, \boldsymbol{x}_2) = \|\phi(\boldsymbol{x}_1) - \phi(\boldsymbol{x}_2)\|$. We can derive how a feasible range of w is related to the distance $d(\boldsymbol{x}_1, \boldsymbol{x}_2)$. Since $\boldsymbol{x}_1 \succ \boldsymbol{x}_2$, then $f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2) \geqslant \triangle > 0$, we can easily follow that

$$\|w\| \geqslant \frac{\triangle}{\|\phi(\boldsymbol{x}_1) - \phi(\boldsymbol{x}_2)\|} = \frac{\triangle}{d(\boldsymbol{x}_1, \boldsymbol{x}_2)}. \tag{22}$$

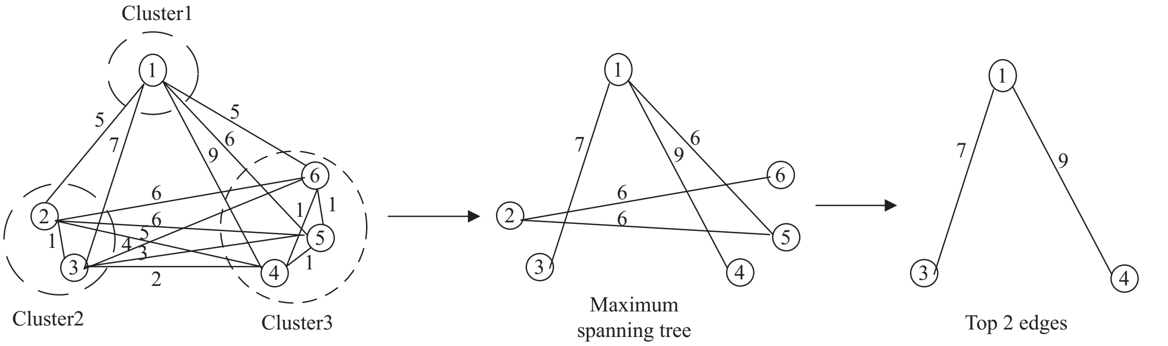

Figure 1 Demonstration of DIV-MST by using a toy dataset.

Table 1 Datasets used in experiments

Name	#Instances	#Attributes	Source
pyrim	74	27	UCI
ailerons	8694	40	UCI
concrete	1030	8	UCI
bank8fm	8192	9	Mldata.org
Wisconsin	194	33	Mldata.org
bodyfat	252	14	StatLib
kin8nm	8192	9	Mldata.org

It is clear from (22), $\|w\|$ may become smaller when $d(\boldsymbol{x}_1, \boldsymbol{x}_2)$ becomes bigger. Therefore, the classification hyperplane margin could be larger and have a greater generalization ability for unknown data.

Based on the above assumptions, we select preference-pairs whose Euclidean distances are ranked as the top H in the kernel space, which is equivalent to the top H minimal kernel function of the preference-pairs. H varies in terms of tasks.

5 Experiments and results

In this section, we present the experiments conducted to evaluate our model, results, and related discussions. We performed two sets of experiments to test our PSVM model. The first set of experiments used seven datasets of varying sizes that were acquired from the UCI machine learning repository, the StatLib out of CMU, and the machine learning data set repository at http://Mldata.org. The second set of experiments applies the PSVM to the TREC-KBA-2012 dataset.

5.1 Experiments with seven datasets

In this set of experiments, we employ seven regression datasets to evaluate our PSVM model. They are acquired from the UCI machine learning repository, the StatLib of CMU, and the machine learning dataset repository in http://Mldata.org. The datasets, their sizes, attributes and sources are listed in Table 1.

For all these datasets, we convert initial continuous output variables into binary class variables by applying a threshold on them. For instance, the variable indicating the strength of concrete in the concrete dataset is applied to define positive and negative classes: a concrete with high strength is a positive class, while a concrete with low strength is a negative class. Here, thresholds for different datasets are determined in terms of separating the whole dataset into positive and negative instances equally. For each dataset, two-thirds was allotted to the training set, while one-thirds was allotted to the testing set. Moreover, we interpret continuous output variables as preference information to generate preference-pairs only in the training set. For example, if the output value of an example is greater than

Table 2 F1 performance of different methods on seven datasets

Dataset	PSVM-L	PSVM-R	SVM	RankingSVM	SVM+
pyrim	0.896	0.774	0.758	0.722	0.838
ailerons	0.833	0.805	0.728	0.717	0.806
concrete	0.849	0.748	0.744	0.729	0.894
bank8fm	0.935	0.928	0.851	0.766	0.896
Wisconsin	0.705	0.647	0.647	0.658	0.647
bodyfat	0.969	0.913	0.853	0.847	0.968
kin8nm	0.879	0.814	0.794	0.799	0.877

Table 3 Accuracy performance of different methods on seven datasets

Dataset	PSVM-L	PSVM-R	SVM	RankingSVM	SVM+
pyrim	0.885	0.731	0.731	0.615	0.807
ailerons	0.815	0.764	0.765	0.603	0.805
concrete	0.832	0.772	0.764	0.627	0.886
bank8fm	0.937	0.930	0.853	0.695	0.896
Wisconsin	0.667	0.621	0.636	0.591	0.621
bodyfat	0.965	0.907	0.849	0.791	0.965
kin8nm	0.889	0.798	0.787	0.755	0.879

the output value of another example in the training set, the two examples can produce a preference-pair. Similarly, we use the continuous output variables as privileged information for SVM+ model.

We use the radial bias function (RBF) kernel in the experiments,

$$K(x_i, x_j) = \exp(-\gamma \parallel x_i - x_j \parallel).$$

We conduct experiments on a 64-bit machine with Intel Xeon 2.4 GHz (L5530), 4 MB cache and 24 GB memory. Similar to Subsection 5.2.4, the constants C, $C'$and γ for PSVM were optimized using 3-fold cross-validation approach in a search grid. For each dataset, we sample 150 preference-pairs in the generated preference-pair set using the random sampling method and the proposed heuristic sampling algorithm given in Section 4. In this section, the corresponding models are abbreviated as PSVM-L and PSVM-R, respectively. For comparison, we run SVM and SVM+ as the baseline classification methods, and RankingSVM as the comparative learning to rank method on these seven datasets.

Results and discussion. We evaluate the performance of different methods by calculating the harmonic mean F1 and accuracy measurements of each testing set for the seven datasets. The F1 and accuracy results of different methods for the seven testing datasets are shown in Tables 2 and 3, respectively. The results on all seven datasets clearly indicate the benefit of using preference-pairs. PSVM-L on all seven datasets outperforms SVM and RankingSVM according to F1 and accuracy values. PSVM-R also outperforms SVM and RankingSVM on all seven datasets. Furthermore, PSVM-L performs better than PSVM-R on all datasets. SVM+ yields better F1 measures when compared with SVM on seven datasets, and SVM+ obtain higher accuracy than SVM except for the Wisconsin data set. PSVM-L outperforms SVM+ on six datasets in F1 and accuracy measurements; however, SVM+ yields higher F1 and accuracy than PSVM-L on the concrete dataset. When we checked the datasets in detail, we found that the mean and variance values (35.898 and 283.714) in Concrete dataset are larger than in other datasets (e.g., ‘pyrim’: 0.655 and 0.010 as well as ‘ailerons’: -0.00088 and 1.712). In other words, the samples in this dataset can be distinguished easier. For other less different data, our PSVM-L performed better, which is also a demonstration of the effectiveness of our model.

We also report experimental time results on the seven datasets in Table 4. The results show that the learning time of SVM+ is longer than that of other methods. Moreover, the learning time of SVM is mostly shorter. The learning time of our PSVM model is between that for SVM and RankingSVM. As PSVM is based on SVM, and is similar to RankingSVM with an extended SMO algorithm for optimization, these results are in line with our expectation. However, SVM+ introduced a correcting function including

Table 4 Learning time of evaluated models for all datasets (ms)

Dataset	PSVM-L	PSVM-R	SVM	RankingSVM	SVM+
pyrim	14	3	4	91	21
ailerons	2399	2396	5618	11946	1410733
concrete	57	60	52	176	28264
bank8fm	3108	2699	2587	5813	37767
Wisconsin	1	6	2	8	5
bodyfat	2	1	5	3	7
kin8nm	1330	7048	1744	17639	16400

Table 5 Four-level relevance estimation scale

Level	Definition
Garbage	Not relevant; e.g., spam
Neutral	Not relevant; nothing can be learned about the target entity
Relevant	Relates indirectly to the target entity; e.g., mentions topics or events that are likely to have an impact on the entity
Central	Relates directly to the target entity; e.g., the entity is a central figure in the mentioned topics or events

privileged information. It involves minimizing the combine functional of two spaces composed of original training data and privileged information [1]. This leads to more complex model than SVM. Therefore, the learning time of SVM+ is much longer than that for SVM, PSVM-R, and PSVM-L.

5.2 Experiments on the TREC-KBA-2012 dataset

5.2.1 *Dataset*

We conduct our experiments on the second dataset of KBA-CCR task of the TREC-2012 competition [2] (TREC-KBA-2012). The following is the goal of the task: given a textual stream corpus consisting of news and social media contents, and an input entity from a knowledge base such as Wikipedia, generate a score for each document based on how pertinent it is to the target entity.

In the manually created training and target annotations, the relevance of entity-document pairs are judged on the basis of a four-level relevance estimation, including Garbage, Neutral, Relevant and Central, whose definitions are listed in Table 5. Therefore, when a binary classification are conducted, for example to distinguish the Central+Relevant samples from Neutral+Garbage samples, samples under a same class but with different levels could be explored as reference-pairs.

Entity set. The entity set consists of 29 Wikipedia items, more specifically, 27 persons and 2 organizations. These entities are described by semi-structured articles in Wikipedia. Each of the entities is identified uniquely by a `urlname`.

Stream corpus. The stream corpus, covering the period from October 2011 to April 2012, includes documents crawled from news, social media, and Linking. Each stream document is time-stamped and uniquely identified by a `steam_id` indicating its date of crawling. The corpus is divided as training and testing instances, with documents from October to December 2011 period as training instances, and the remainder for testing. We follow this setup.

The detailed annotations about training and testing instances are listed in Table 6.

5.2.2 *Evaluation scenarios*

According to different granularity settings, we evaluate the proposed model in the following two classification scenarios. These two scenarios are also the official target evaluations conducted in the KBA-CCR task.

[2] http://trec-kba.org/kba-ccr-2012.shtml.

Table 6 Number of training and testing instances labelled

Level	#Training instances	#Testing instances	#Subtotal	#Total
Garbage	9382	20439	29821	57755
Neutral	1757	2470	4227	
Relevant	6500	8426	14926	
Central	3525	5256	8781	

Central vs. others. In this scenario, only Central entity-document pairs are treated as positive instances, and the others as negative instances. Therefore, we denote this scenario as Central Only in the following descriptions. We sample preference-pair data from Relevant vs. Neutral, Relevant vs. Garbage, and Neutral vs. Garbage samples, and set $m = 40$ with the proposed two-layer heuristic sampling algorithm in Section 4.

Central+Relevant vs. Neutral+Garbage. In this scenario, both Central and Relevant entity-document pairs are considered as positive instances, and the others as negative instances. It is denoted as Central+Relevant in the following descriptions. Preference-pair samples are selected between Central and Relevant instances using the two-layer heuristic algorithm proposed in Section 4, and also $m = 40$ is set.

5.2.3 *Evaluation metrics*

We follow the evaluation methodology of the KBA-CCR track of TREC-2012. For each target entity, a document is assigned a confidence score in the range of 0 to 1000 with respect to how likely it is for a human to cite that document. Scoring is done by sweeping a confidence cutoff from 0 to 1000 in steps of 50, and documents with a score above this cutoff are treated as positive instances.

Next, the scoring tool computes precision and recall for each entity and for each cutoff value with respect to the assessors's judgments. Then, macro-average precision and recall are computed for each cutoff, where macro-averaged precision (recall) means summing up the precision (recall) scores for all the query entities, and then dividing the sum by the number of entities for a given cutoff. Finally, the harmonic mean F1 of the macro-average precision and macro-average recall are determined for each cutoff. We choose the highest F1 as the result of the current model, using the evaluation scripts provided by the TREC committee. The scoring tool also computes scale utility, a metric from general information filtering, used to evaluate the ability of a system to accept relevant and reject non-relevant documents from a document stream [26].

5.2.4 *Experimental setting*

Twelve variants of PSVM are evaluated using random and two-layers heuristic sampling algorithms in terms of the number of sampling preference-pairs. They are abbreviated as PSVM-R_XX and PSVM-2L_XX, respectively, where XX is the number of sampled preference-pairs. For comparison, we run LIBSVM[3] as the baseline classification method, and RankingSVM[4] as the comparative Learning to Rank method with three rankings consisting of Central, Relevant, and others. Moreover, we run SVM+ implemented by Pechyony [5] by using label information and source of documents as privileged information.

In our current experiments, we use the features proposed in [27,28] that have been proved to be effective for the training and testing samples.

We use RBF kernel

$$K(x_i, x_j) = \exp(-\gamma \parallel x_i - x_j \parallel),$$

and a 3-fold cross-validation for selecting hyper-parameters. We conduct the experiments on a 64-bit machine with Intel Xeon 2.4 GHZ (L5530), 4 MB cache and 24 GB memory. For the proposed method, we

[3] https://www.csie.ntu.edu.tw/cjlin/libsvm/index.html.

[4] http://www.csie.ntu.edu.tw/˜cjlin/libsvmtools/#large_scale_ranksvm.

[5] http://www.cs.technion.ac.il/pechyony/conj_svm.tar.gz.

Table 7 Overall results of evaluated methods

Method	Central Only				Central+Relevant			
	F1	SU	σ	Time (ms)	F1	SU	σ	Time (ms)
PSVM-2L_10	0.320	0.344	0.229	1660	0.713	0.710	0.214	2230
PSVM-2L_30	0.321	0.370	0.227	7620	0.690	0.682	0.277	6270
PSVM-2L_50	0.322	0.349	0.231	17660	0.701	0.712	0.253	8090
PSVM-2L_100	0.318	0.358	0.228	5710	0.689	0.682	0.279	2380
PSVM-2L_250	0.335	0.338	0.213	30260	0.694	0.694	0.232	17750
PSVM-2L_300	0.333	0.354	0.217	20580	**0.717**	**0.714**	0.220	23570
PSVM-R_10	0.320	0.344	0.225	3410	0.688	0.680	0.279	2820
PSVM-R_30	0.327	0.333	0.222	2090	0.688	0.680	0.279	5020
PSVM-R_50	0.318	0.319	0.228	8680	0.688	0.680	0.279	13430
PSVM-R_100	0.322	0.315	0.227	21060	0.688	0.680	0.279	14440
PSVM-R_250	0.327	0.342	0.225	24720	0.688	0.680	0.279	10490
PSVM-R_300	0.318	0.320	0.228	18030	0.688	0.680	0.279	17260
SVM	0.338	0.371	0.222	2190	0.688	0.681	0.279	2110
RankingSVM	0.325	0.291	0.235	44867	0.613	0.604	0.276	44867
SVM+	0.329	0.327	0.226	259161	0.689	0.682	0.278	438359
HLTCOE	**0.359**	**0.402**	0.242	–	0.492	0.555	0.256	–
UDel	0.355	0.331	0.208	–	0.597	0.591	0.213	–
3-step RF	0.351	0.347	0.215	–	0.691	0.673	0.260	–

'–' means that it is a reference baseline, so we can not conduct the experiments again.

first choose the one that has the best accuracy on grids of (C, C', γ) values, where $C, C' \in \{2^{-5}, 2^{-4}, \ldots, 2^{15}\}$ and $\gamma \in \{0.05, 0.1, 0.2, 0.4, 0.45, 0.5, 0.55, 0.6, 0.7, 1.6, 3.2, 6.4, 12.8\}$. We choose the one that has the best prediction performance in the 3-fold cross-validation, and then train the model using the whole training data. The hyper-parameter selection in the SVM, SVM+ and RankingSVM are the same as the PSVM on parameter grids (C, γ).

For further references, we also include the two best performing official runs HLTCOE [29] and UDEL [30] from TREC-2012, and a 3-step Random Forest (RF) [31] run.

5.2.5 *Results and discussion*

Here we present the overall performance of all the experimental methods including F1, scale utility (SU), standard deviations σ and time complexity of experiments. The results are listed in Table 7.

In comparison to other methods listed in the last two blocks of Table 7, PSVM with two-layer heuristic sampling achieves higher or competitive F1 in the Central+Relevant scenario considerably. Specifically, compared with the SVM baseline, our best PSVM-2L_300 improves F1 by about 4.2%. In contrast to SVM+, PSVM-2L_300 improves F1 by roughly 4.1%. In comparison to RankingSVM, our PSVM-2L_300 model improves F1 by approximately 16%. These results validate our motivations that (i) difference information from preference-pairs can enhance classification performance. However, in comparison to SVM, F1 of PSVM-R_xx which randomly samples the preference-pairs between Central and Relevant annotation data is the same as the SVM result. This indicates that randomly sampling a small number of preference-pairs does not enhance classification performance. (ii) The two-layer heuristic sampling algorithm helps obtain effective preference-pairs. (iii) The PSVM with a small number of effective preference-pairs can achieve higher performance.

We conduct a t-test in the Central+Relevant scenario to evaluate the statistical performance difference between PSVM and other methods. More specifically, we first acquire the F1 values of all the entities corresponding to the cutoff with which the model achieves the maximum F1 performance (cutoff ranges from 0 to 1000 by steps 50) for each of the above methods. Then, we use the F1 values of all entities to compute p-values between different methods by using t-test with double tail and paired samples. The p-values between PSVM-2L_300 and SVM, SVM+, RankingSVM, HLTCOE, UDel and 3-step RF are

0.0247, 0.047, 0.0176, 0.0005, 0.0001 and 0.048, respectively, which are all less than 0.05. According to these p-values, the performance of PSVM is approved to be significantly different with the other six comparative methods.

In addition to p-values, we also compute the standard deviations σ of different methods as shown in the 4th and 8th columns of Table 7. The standard deviation of PSVM-2L_300 (0.22) is lower than most of the other methods in the Central+Relevant scenario.

We include experimental results for time complexity in the 5th and 9th columns in Table 7. The results show that the learning time of PSVM is between that for the SVM and RankingSVM. However, the learning time of SVM+ is much longer than other methods.

In the Central Only scenario, compared with the HLTCOE, UDEL and 3-step RF methods which achieved best results in the TREC-2012 competition, F1 values of PSVMs are lower. This is because, the preference-pairs that we incorporated are among Relevant vs. Neutal, Relevant vs. Garbage, and Neutral vs. Garbage, without Central information. As these three levels are not crucial for the entity-document citation application, there is too much noise in their annotations. Therefore, the F1 values of PSVM in the Central Only scenario are lower than the other baselines.

6 Conclusion

In this paper, we proposed a PSVM model that utilizes preference data for classification and a two-layer heuristic sampling algorithm to select preference-pair data. Experimental results on TREC-KBA-2012 dataset demonstrated that our model is able to exhibit better performance than other baselines in the Central+Relevent scenario. Moreover, experiment results on seven other datasets acquired from UCI, StatLib, and MLdata.org show that our model is also able to outperform other baselines. We conclude that difference information underlying preference-pairs helps improve classification performance. An interesting direction of future work is to explore general difference information to enhance classification performance. In addition, there is a focus on determining the number of selection diversity instances and the number of top preference-pairs in a data set in the future work.

Acknowledgements This work was supported by National Key Research and Development Program of China (Grant No. 2016YFB1000902), National Basic Research Program of China (973 Program) (Grant No. 2013CB329 600), National Natural Science Foundation of China (Grant No. 61472040), and National Natural Science Basic Research Plan in Shaanxi Province of China (Grant No. 2016JM6082).

Conflict of interest The authors declare that they have no conflict of interest.

References

1 Vapnik V, Vashist A, Pavlovitch N. Learning using hidden information (learning with teacher). In: Proceedings of International Joint Conference on Neural Networks, Atlanta, 2009. 3188–3195

2 Sharmanska V, Quadrianto N, Lampert C H. Learning to rank using privileged information. In: Proceedings of IEEE International Conference on Computer Vision (ICCV), Sydney, 2013. 825–832

3 Wang Z, Gao T, Ji Q. Learning with hidden information using a max-margin latent variable model. In: Proceedings of the 22nd International Conference on Pattern Recognition (ICPR), Stockholm, 2014. 1389–1394

4 Feyereisl J, Kwak S, Son J, et al. Object localization based on structural SVM using privileged information. Adv Neural Inf Process Syst, 2014, 1: 208–216

5 Vladimir V N, Vapnik V. The nature of statistical learning theory. IEEE Trans Neural Netw, 1995, 8: 1564–1564

6 Tsang I W, Kwok J T, Cheung P M. Core vector machines: fast SVM training on very large data sets. J Mach Learn Res, 2005, 6: 363–392

7 Sun C Y, Mu C X, Li X M. A weighted LS-SVM approach for the identification of a class of nonlinear inverse systems. Sci China Ser F-Inf Sci, 2009, 52: 770–779

8 Yang T, Li Y F, Mahdavi M, et al. Nyström method vs random fourier features: a theoretical and empirical comparison. Adv Neural Inf Process Syst, 2012, 1: 476–484

9 Qu A P, Chen J M, Wang L W, et al. Segmentation of hematoxylin-eosin stained breast cancer histopathological images based on pixel-wise SVM classifier. Sci China Inf Sci, 2015, 58: 092105

10 Pechyony D, Izmailov R, Vashist A, et al. SMO-style algorithms for learning using privileged information. In: Proceedings of the 2010 International Conference on Data Mining, Las Vegas, 2010. 235–241

11 Kuo T M, Lee C P, Lin C J. Large-scale kernel rankSVM. In: Proceedings of the SIAM International Conference on Data Mining. New York: ACM, 2014. 812–820

12 Herbrich R, Graepel T, Obermayer K. Large margin rank boundaries for ordinal regression. In: Advances in Large Margin Classfiers. Cambridge: MIT Press, 2000. 115–132

13 Cao Y, Xu J, Liu T Y, et al. Adapting RankingSVM to document retrieval. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM, 2006. 186–193

14 Yu H, Kim Y, Hwang S. RV-SVM: an efficient method for learning ranking SVM. In: Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining. Berlin: Springer, 2009. 426–438

15 Schohn G, Cohn D. Less is more: active learning with support vector machines. In: Proceedings of the 17th International Conference on Machine Learning. San Francisco: Morgan Kaufmann Publishers Inc., 2000. 839–846

16 Tong S, Koller D. Support vector machine active learning with applications to text classification. J Mach Learn Res, 2001, 2: 45–66

17 Brinker K. Incorporating diversity in active learning with support vector machines. In: Proceedings of the 20th International Conference on Machine Learning, Washington, 2003. 59–66

18 Brinker K. Active learning of label ranking functions. In: Proceedings of the 21st International Conference on Machine Learning. New York: ACM, 2004. 17

19 Fürnkranz J, Hüllermeier E. Pairwise preference learning and ranking. In: Proceedings of the 14th European Conference on Machine Learning. Berlin: Springer, 2003. 145–156

20 Yu H. SVM selective sampling for ranking with application to data retrieval. In: Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining. New York: ACM, 2005. 354–363

21 Yu H. Selective sampling techniques for feedback-based data retrieval. Data Min Knowl Disc, 2011, 22: 1–30

22 Lin K Y, Jan T K, Lin H T. Data selection techniques for large-scale rank SVM. In: Proceedings of International Conference on Technologies and Applications of Artificial Intelligence (TAAI), Taipei, 2013. 25–30

23 Schölkopf B, Herbrich R, Smola A J. A generalized representer theorem. In: Proceedings of the 14th Annual Conference on Computational Learning Theory. London: Springer, 2001. 416–426

24 John P. Fast training of support vector machines using sequential minimal optimization. Cambridge: MIT Press, 1999. 185–208

25 Fan R E, Chen P H, Lin C J. Working set selection using second order information for training support vector machines. J Mach Learn Res, 2005, 6: 1889–1918

26 Robertson S E, Soboroff I. The Trec 2002 Filtering Track Report. Technical Report, In TREC'02. 2003

27 Wang J G, Song D D, Lin C Y, et al. Bit and Msra at Trec Kba Ccr Track 2013. Technical Report, DTIC Document. 2013

28 Wang J G, Liao L J, Song D D, et al. Resorting relevance evidences to cumulative citation recommendation for knowledge base acceleration. In: Proceedings of International Conference on Web-Age Information Management. Berlin: Springer, 2015. 169–180

29 Kjersten B, McNamee P. The Hltcoe Approach to the Trec 2012 Kba Track. Technical Report, In TREC'12. 2013

30 Liu X, Fang H. Entity Profile Based Approach in Automatic Knowledge Finding. Technical Report, In TREC'12. 2013

31 Balog K, Ramampiaro H, Takhirov N, et al. Multi-step classification approaches to cumulative citation recommendation. In: Proceedings of the 10th Conference on Open Research Areas in Information Retrieval. New York: ACM, 2013. 121–128