Pattern Recognition Letters 30 (2009) 1236–1240

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Least squares one-class support vector machine

Young-Sik Choi *

Department of Computer Engineering at Korea Aerospace University, Goyang City, Gyeonggi Province 412-791, Republic of Korea

ARTICLE INFO

Article history:
Received 29 September 2008
Received in revised form 13 April 2009
Available online 18 May 2009

Communicated by R.P.W. Duin

Keywords:
LS (least squares) one-class SVM
Proximity measure
Relevance ranking
One-class SVM (support vector machine)

ABSTRACT

In this paper, we reformulate a standard one-class SVM (support vector machine) and derive a least squares version of the method, which we call LS (least squares) one-class SVM. The LS one-class SVM extracts a hyperplane as an optimal description of training objects in a regularized least squares sense. One can use the distance to the hyperplane as a proximity measure to determine which objects resemble training objects better than others. This differs from the standard one-class SVMs that detect which objects resemble training objects. We demonstrate the performance of the LS one-class SVM on relevance ranking with positive examples, and also present the comparison with traditional methods including the standard one-class SVM. The experimental results indicate the efficacy of the LS one-class SVM.

1. Introduction

One-class classification problem is to make a description of a set of training objects and to detect which objects resemble this training set (Tax and Duin, 2004). Approach to this problem taken by the standard one-class SVMs (support vector machines) is extract the regions where a certain fraction of training objects may locate, and classify an object according to whether the object resides inside the region or not. There are two standard algorithms in the literature (Müller et al., 2001; Schölkopf et al., 2001; Tax and Duin, 1999, 2004), which are equivalent for a certain type of kernel functions such as Gaussian kernel function (Müller et al., 2001). One of the standard one-class SVMs is estimate the sphere of minimum volume which encloses a given fraction of training objects (Tax and Duin, 1999, 2004). The other is extract a hyperplane in a kernel feature space such that a given fraction of training objects may reside beyond the hyperplane, while at the same time the hyperplane has maximal distance to the origin (Schölkopf et al., 2001). These standard one-class SVMs have been successfully applied for novelty detection (Deng and Xu, 2007; Ma and Perkins, 2003; Tax and Duin, 2004).

In this paper, we reformulate the standard one-class SVM in (Schölkopf et al., 2001) and derive a least squares version of the method, which is called the LS (least squares) one-class SVM. The LS one-class SVM uses a quadratic loss function and equality constraints, and extracts a hyperplane with respect to which the distances from training objects are minimized in a regularized least squares sense. This reformulation is very similar to the derivation of the LS SVM from the standard the SVM classifier (Suykens and Vandewalle, 1999; Suykens et al., 2002), in that both LS approaches use the quadratic loss functions. Hence, the proposed LS one-class SVM also loses the sparseness property of the standard one-class SVMs. One may overcome the loss of the sparseness by pruning training samples (Kruif and Vries, 2003; Kuh and De Wilde, 2007).

The hyperplane obtained from the LS one-class SVM is not the boundary of regions as in the standard one-class SVMs. Instead, it represents a hyperplane which most of training objects may lie close to. One can use the distance to the hyperplane as a proximity measure to determine which objects resemble training objects better than others. In this paper, we apply the LS one-class SVM for relevance ranking with positive examples. In the ranking problem, one should rank all documents according to the proximity to the set of training documents. This is important in modern information retrieval problems (Chakrabarti et al., 1999; Chen et al., 2001; Manevitz and Yousef, 2001; Setia et al., 2005).

There have been several attempts to use the distance from the center of sphere obtained from the standard one-class SVM (Tax and Duin, 2004) as a proximity measure to the training set (Chen et al., 2001; Manevitz and Yousef, 2001). Despite of the usefulness of these approaches, the distance to the center of sphere does not necessarily reflect the proximity to the training set. For instance, an object closer to the center might be farther from training objects. This is because in the standard one-class SVM, the training objects inside the regions may not contribute to the construction of the regions. On the other hand, the LS one-class SVM seeks to minimize the sum of distances from all training objects to the hyperplane in a regularized least squares fashion and thus most of training objects may lie close to the hyperplane. Therefore, the proximity to such hyperplane can better reflect the proximity to the training set.

* Tel.: +82 2 300 0189; fax: +82 2 3158 1419.
E-mail address: choimail@kau.ac.kr

0167-8655/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2009.05.007

There are several research works (Suykens et al., 2003; Roth, 2004) related to the proposed LS one-class SVM. Suykens et al. (2003) showed that the kernel PCA (principal component analysis) can be interpreted as a one-class modeling problem with zero target value. Roth has presented a one-class kernel Fisher discriminant which is a kernel ridge regression (Saunders et al., 1998) with a single target value with assumption of Gaussian distribution of data samples. The LS one-class SVM can be also regarded as a kind of kernel ridge regression with a single target value. Unlike Roth's approach, the proposed LS one-class SVM does not make any assumption on the distribution of data samples. Thus the LS one-class SVM can provide more flexibility on the handling of one-class problems.

In Section 2, we briefly introduce the standard one-class SVMs, and present the proposed LS one-class SVM. In Section 3, we discuss the differences between the LS and the standard one-class SVMs. Section 4 presents experimental results with several collections of Web pages, comparing the standard one-class SVMs. We make conclusions in Section 5.

2. Least squares one-class support vector machines

In this section, we briefly introduce the standard one-class SVMs and then present the LS one-class SVM. First, we define a mapping function to be used in the following description. Suppose that we are given an input data set S containing n points $\{\boldsymbol{x}_j: j = 1,\ldots,n\}$, where $S \subseteq \boldsymbol{X}$ and $X \subseteq \mathrm{R}^d$. Then, we define a map ϕ: $X \to \boldsymbol{F}$ to be the mapping of $\boldsymbol{X}$ into feature space $\boldsymbol{F}$ such that a dot product in feature space can be computed by a kernel function, i.e. $\phi(\boldsymbol{x}_i) \cdot \phi(\boldsymbol{x}_j) = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

2.1. Standard one-class support vector machines

The standard one-class SVM (Schölkopf et al., 2001) can be stated as the following objective function to be minimized:

$$\frac{1}{2}\|\boldsymbol{w}\|^2 - \rho + C\sum_j \xi_j, \tag{1}$$

subject to $\boldsymbol{w} \cdot \phi(\boldsymbol{x}_j) \geqslant \rho - \xi_j$ and $\xi_j \geqslant 0$. Here, $\boldsymbol{w} \cdot \phi(\boldsymbol{x}) = \rho$ represents a hyperplane in feature space, $\|\cdot\|$ denotes Euclidean norm, and ξ_j slack variables. The parameter C is predefined and controls the fraction of outliers (Müller et al., 2001; Schölkopf et al., 2001).

Eq. (1) seeks to extract a hyperplane which has the maximal distance $\rho/\|\boldsymbol{w}\|^2$ from the origin and beyond which most of training examples may reside. The hyperplane can be obtained by solving the following dual objective function to be maximized:

$$-\sum_{i,j} \alpha_i\alpha_j K(\boldsymbol{x}_i, \boldsymbol{x}_j), \tag{2}$$

subject to $0 \leqslant \alpha_j \leqslant C$ and $\sum_j \alpha_j = 1$, where α_j denotes Lagrangian multiplier. The obtained hyperplane $f(\boldsymbol{x})$ can be written as

$$f(\boldsymbol{x}) = \sum_i \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}) - \rho. \tag{3}$$

One can determine the values of α_j using the traditional quadratic programming with a linear constraint. The bias term ρ can be also obtained from $f(\boldsymbol{x}_s) = 0$, where $\boldsymbol{x}_s$ denotes one of the support vectors obtained. The decision function $g(\boldsymbol{x})$ for one-class classification is simply to take the sign of $f(\boldsymbol{x})$ as follows.

$$g(\boldsymbol{x}) = \mathrm{sgn}(f(\boldsymbol{x})) = \mathrm{sgn}\left(\sum_i \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i) - \rho\right). \tag{4}$$

Another standard one-class SVM (Vapnik, 1998; Tax and Duin, 1999, 2004), which is also called support vector data description, can be formulated as the following objective function to be minimized.

$$R^2 + C\sum_j \xi_j, \tag{5}$$

subject to $\|\phi(\boldsymbol{x}_j) - \boldsymbol{a}\|^2 \leqslant R^2 + \xi_j$ and $\xi_j \geqslant 0$ for all $\boldsymbol{x}_j$, where vector $\boldsymbol{a}$ denotes the center of the sphere.

Eq. (5) seeks to extract the sphere of the minimum radius R enclosing the fraction of training objects. One can obtain the sphere by solving the following dual objective function to be maximized.

$$\sum_j \alpha_j K(\boldsymbol{x}_j, \boldsymbol{x}_j) - \sum_{i,j} \alpha_i\alpha_j K(\boldsymbol{x}_i, \boldsymbol{x}_j), \tag{6}$$

with $0 \leqslant \alpha_j \leqslant C$ and $\sum_j \alpha_j = 1$, where α_j denotes Lagrangian multiplier. Note that (6) is equivalent to (2) for the kernel functions satisfying with $K(\boldsymbol{x}_j,\boldsymbol{x}_j) = 1$. The obtained center of the sphere can be written as follows:

$$\boldsymbol{a} = \sum_j \alpha_j \phi(\boldsymbol{x}_j). \tag{7}$$

The values of α_j can be determined using the quadratic programming, and the value of R^2 can be computed from $\|\phi(\boldsymbol{x}_s) - \boldsymbol{a}\|^2 = R^2$, where $\boldsymbol{x}_s$ denotes one of the support vectors. The decision function for one-class classification simply becomes

$$g(\boldsymbol{x}) = \mathrm{sgn}(R^2 - \|\phi(\boldsymbol{x}) - \boldsymbol{a}\|^2). \tag{8}$$

2.2. Least squares one-class support vector machine

To derive a LS (least squares) version of the standard one-class SVM, we reformulate the one-class SVM described in (1) by using a quadratic error function and the equality conditions. The corresponding LS one-class SVM can be written as the following objective function to be minimized:

$$\frac{1}{2}\|\boldsymbol{w}\|^2 - \rho + \frac{1}{2}C\sum_j \xi_j^2, \tag{9}$$

subject to $\boldsymbol{w} \cdot \phi(\boldsymbol{x}_j) = \rho - \xi_j$. Now, the conditions for the slack variables, $\xi_j \geqslant 0$ in (1) no longer hold. Instead, the variable ξ_j represents an error caused by a training object $\boldsymbol{x}_j$ with respect to the hyperplane, i.e. $\xi_j = \rho - \boldsymbol{w} \cdot \phi(\boldsymbol{x}_j)$.

The LS one-class SVM described in (9) seeks to extract a hyperplane which has the maximal distance $\rho/\|\boldsymbol{w}\|^2$ from the origin, and with respect to which the sum of the squares of errors, ξ_j^2 are minimized. One can solve the problem in (9) as follows. By introducing Lagrangian multipliers α_j, the corresponding objective function can be written as the following.

$$L = \frac{\|\boldsymbol{w}\|^2}{2} - \rho + \frac{C}{2}\sum_j \xi_j^2 - \sum_j \alpha_j(\phi(\boldsymbol{x}) \cdot \boldsymbol{w} + \xi_j - \rho). \tag{10}$$

Setting to zero the first derivatives of (10) with respect to $\boldsymbol{w}$, ξ_j, ρ, and α_j leads to the following relations:

$$\begin{aligned}
&\frac{\partial L}{\partial \boldsymbol{w}} = 0 \to \boldsymbol{w} = \sum_j \alpha_j \phi(\boldsymbol{x}_j),\\
&\frac{\partial L}{\partial \xi_j} = 0 \to \xi_j = \alpha_j / C,\\
&\frac{\partial L}{\partial \rho} = 0 \to \sum_{j=1} \alpha_j = 1,\\
&\frac{\partial L}{\partial \alpha_j} = 0 \to \phi(\boldsymbol{x}_j) \cdot \boldsymbol{w} + \xi_j - \rho = 0.
\end{aligned} \tag{11}$$

Eliminating $\boldsymbol{w}$ and ξ_j through substitution in (11) yields

$$\sum_i \alpha_i \phi(\boldsymbol{x}_i) \cdot \phi(\boldsymbol{x}_j) - \rho + \alpha_j / C = 0. \tag{12}$$

With the linear constraint for α_j, i.e. $\sum_{j=1} \alpha_j = 1$ in (11), equations in (14) reduce to the following set of linear equations to solve.

$$\left[\begin{array}{c|c} 0 & \mathbf{e}' \\ \hline \mathbf{e} & \mathbf{K} + \mathbf{I}/C \end{array}\right] \left[\begin{array}{c} -\rho \\ \hline \boldsymbol{\alpha} \end{array}\right] = \left[\begin{array}{c} 1 \\ \hline \mathbf{0} \end{array}\right] \tag{13}$$

Here, $\boldsymbol{\alpha}$ denotes the column vector of Lagrangian multipliers, $[\alpha_1, \ldots, \alpha_n]'$. The vectors $\boldsymbol{e}$ and $\boldsymbol{0}$ represent all one and all zero column vectors of n dimension, i.e.$[1, \ldots, 1]'$ and $[0, \ldots, 0]'$, respectively. The matrix $\boldsymbol{K}$ denotes the kernel matrix with entries $\boldsymbol{K}_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$, and $\boldsymbol{I}$ denotes the identity matrix.

Using the block matrix inversion (Kailath, 1980), one can easily obtain ρ and $\boldsymbol{\alpha}$ as follows.

$$\begin{aligned} &\rho = 1/(\boldsymbol{e}'(\boldsymbol{I}/C + \boldsymbol{K})^{-1}\boldsymbol{e}), \\ &\boldsymbol{\alpha} = ((\boldsymbol{I}/C + \boldsymbol{K})^{-1}\boldsymbol{e})/(\boldsymbol{e}'(\boldsymbol{I}/C + \boldsymbol{K})^{-1}\boldsymbol{e}). \end{aligned} \tag{14}$$

Note that $\boldsymbol{\alpha}$ satisfies the linear constraint $\boldsymbol{\alpha} \cdot \boldsymbol{e} = 1$. The hyperplane obtained from (15) can be written in a vector form as follows:

$$f(\boldsymbol{x}) = \boldsymbol{w} \cdot \phi(\boldsymbol{x}) - \rho = \boldsymbol{k}'\boldsymbol{\alpha} - \rho, \tag{15}$$

where $\boldsymbol{k}$ denotes a vector with entries $\boldsymbol{k}_i = K(\boldsymbol{x}_i, \boldsymbol{x})$, $i = 1, \ldots, n$. Then, substituting (14) into (15), one can obtain the following equation for the hyperplane.

$$f(\boldsymbol{x}) = (\boldsymbol{k}'(\boldsymbol{I}/C + \boldsymbol{K})^{-1}\boldsymbol{e} - 1)/(\boldsymbol{e}'(\boldsymbol{I}/C + \boldsymbol{K})^{-1}\boldsymbol{e}). \tag{16}$$

One can show the relation of the proposed LS one-class SVM to other approaches to the modeling of one-class data distributions by setting the bias ρ to a predefined constant value. (Note that in the LS one-class SVM formulation (9), the LS one-class SVM does not make any assumption on the bias ρ and instead it is a variable to be optimized.) For instance, if we set the bias ρ to y, then the objective function (9) reduces to

$$\frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{1}{2}C\sum_j \xi_j^2, \tag{17}$$

subject to $\xi_j = y - \boldsymbol{w} \cdot \phi(\boldsymbol{x}_j)$. This new formulation (17) is equivalent to a kernel ridge regression (Cristianini and Shawe-Taylor, 2000) with a single target value for all training samples. This can be also considered as the one-class kernel Fisher discriminant (Roth, 2004) if we set y to 1, and select the appropriate scaling factor for $\boldsymbol{w}$ to satisfy the Fisher's normalization conditions (Hastie et al., 1995; Roth, 2004). Moreover, if y is set to zero, then (17) can be interpreted as the kernel PCA (principal component analysis) described in (Suykens et al., 2003) where $\boldsymbol{\alpha}$ can be obtained as the eigenvector associated with the largest eigenvalue of the centered Gram matrix.

One can easily obtain the hyperplane for (17) as the following.

$$f(\boldsymbol{x}) = y(\boldsymbol{k}'(\boldsymbol{I}/C + \boldsymbol{K})^{-1}\boldsymbol{e} - 1). \tag{18}$$

where $\boldsymbol{\alpha} = y(\boldsymbol{I}/C + \boldsymbol{K})^{-1}\boldsymbol{e}$. The difference between (16) and (18) is that (18) has no normalization factor as (16) since the linear constraint, $\boldsymbol{\alpha} \cdot \boldsymbol{e} = 1$, no longer holds in (17). Therefore, both (16) and (18) can produce equivalent results in some applications such as relevance ranking problems where the relative distance to the hyperplane only matters.

It is noteworthy that the proposed LS one-class SVM loses the sparseness due to the quadratic loss function in the objective function (9). In order to overcome the loss of sparseness, various approaches have been proposed in the literature (Kruif and Vries, 2003; Kuh and De Wilde, 2007).

3. Proximity measure

Note that unlike the standard one-class SVM, the proposed LS one-class SVM does not have a decision function as described in (4) and (8). Instead, the hyperplane in (16) itself represents the optimal hyperplane in a regularized least squares sense, where most of training objects may reside. This feature differentiates the LS one-class SVM from the standard one-class SVMs, and leads to different application areas, which will be discussed in the following sections.

The LS one-class SVM seeks to minimize the sum of squares errors, i.e. the sum of $\xi_j^2 = | w \cdot \phi(x_j) - \rho|^2 = | f(x_j)|^2$. An object with a low value of $|f(\boldsymbol{x})|$ lies close to the hyperplane and thus resembles the training set better than other objects with a higher value of $|f(\boldsymbol{x})|$. Therefore, one can use the value of $|f(\boldsymbol{x})|$ as a proximity measure to the training set of examples. On the other hand, the standard one-class SVMs obtain the regions where the given fraction of training objects may reside. Therefore, the corresponding decision functions in (4) and (8) can be used to classify an object according to whether or not the objects fall inside these regions.

To clarify the differences between the two one-class SVMs, we have shown the experimental results in Figs. 1 and 2 with a training set of two-dimensional objects which was generated out of a

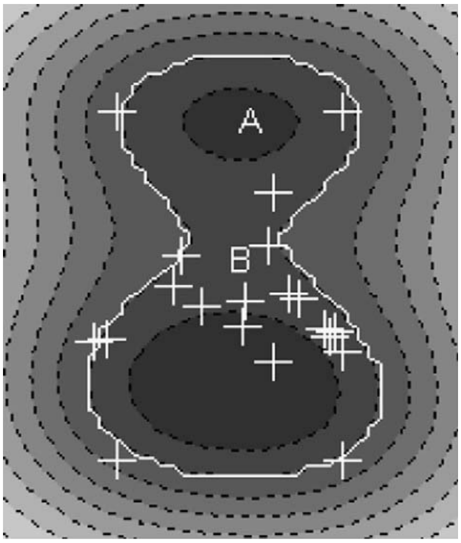

Fig. 1. Contour Map from one-class SVM, where + denotes data points.

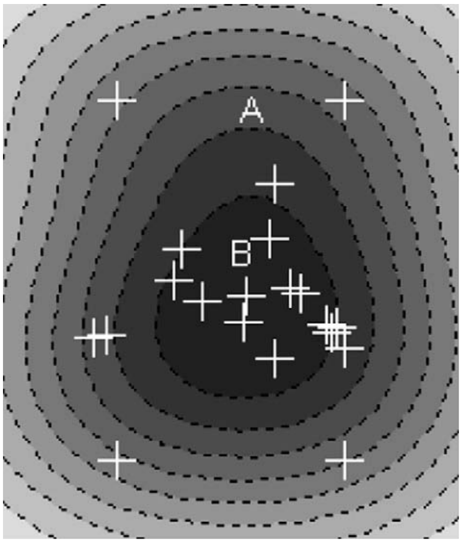

Fig. 2. Contour Map from LS one-class SVM, where + denotes data points.

Gaussian distribution with 10% of random noise. In Figs. 1 and 2, the gray value represents the distance from the center of the sphere, $\|\phi(\boldsymbol{x})-\boldsymbol{a}\|^2$, and the proximity to the hyperplane, $|f(\boldsymbol{x})|$, respectively; dark is close and light is remote. The white solid line in Figs. 1 and 2 represents the boundary of regions extracted from the one-class SVM. Note that we set C to 1 for the LS one-class SVM, adjusted the value of C to allow 10% of outliers for the standard one-class SVM, and used the same RBF kernel functions for both the one-class SVMs.

As shown in Fig. 1, the standard one-class SVM well finds the sphere of minimum volume enclosing the given fraction of training objects. However, the contour map inside the extracted regions does not well reflect the distribution of training objects therein. Thus, the proximity of some objects to the training set might be distorted. For instance, point A is closer to the center of the sphere than point B, where the other way seems to be more proper. On the other hand, as shown in Fig. 2, the LS one-class SVM well captures the distribution of training objects. This suggests that the LS one-class SVM is favored over the standard one-class SVM as long as the proximity measure is concerned.

4. Experimental results

To investigate the performance of the proposed method in real applications, we have conducted relevance ranking with real datasets: 20NG (News Group), WebKB, and Reuters datasets, well known for text classification problems (Chakrabarti, 2003). The 20NG dataset contains 20 topics and we used them all in this experiment. The WebKB dataset consists of seven categories, and we only used four major categories: Course, Faculty, Project, and Student. The Reuters dataset contains about 10,700 documents with 135 categories. In this experiment, we only used top 10 largest categories, comprising about 8891 documents.

We removed stop words and terms with low document frequency ($\leqslant 3$). Then, we transformed documents into vectors of term frequency with unit-length. The dimensions of the vectors were 39,366 for 20NG, 26,184 for WebKB, and 10,164 for Reuters, respectively.

To evaluate the performance for relevance ranking, we used the average-precision (Chakrabarti, 2003), which reflects the precision and recall in one measure. Suppose that there is a corpus of n documents D and a set of queries Q. For each query $q \in Q$, an exhaustive set of relevant documents $D_q \subseteq D$ is identified manually and a ranked list of documents $(d_1, d_2, \ldots, d_n)$ is returned. For the ranked list of documents, we can obtain a relevance list $(r_1, r_2, \ldots, r_n)$, where $r_i = 1$ if $d_i \in D_q$ and 0 otherwise. Then, the average-precision is computed as

$$\frac{1}{|D_q|}\sum_{1\leqslant k\leqslant |D|} r_k \times p(k), \tag{19}$$

where $p(k)$ represents the precision at rank k defined as $\sum_{1\leqslant i\leqslant k} r_i/k$. Note that the average-precision is 1 only if the system retrieves all relevant documents and ranks them ahead of any irrelevant document.

We compared the performance of the LS one-class SVM with other traditional methods, including the standard one-class SVM, the kernel Mean, and the kernel PCA (principal component analysis) (Schölkopf et al., 1999). The LS and standard one-class SVMs use $|f(\boldsymbol{x})|$ and $\|\phi(\boldsymbol{x})-\boldsymbol{a}\|^2$ as the proximity measures, respectively, as described in Section 3. The dissimilarity measure from the kernel Mean is defined to be $\|\phi(\boldsymbol{x})-\boldsymbol{m}\|^2$, where $\boldsymbol{m}$ is the kernel mean vector in a feature space. The proximity measure from the kernel PCA is defined as $|\boldsymbol{w}\cdot(\phi(\boldsymbol{x})-\boldsymbol{m})|$. Note that in the kernel PCA each data point is centered in general.

For the determination of parameters C and σ in the RBF kernel function, we used the following approach similar to the one described in (Van Gestel et al., 2004).

(1) Select a subset S with 1/3 of the data from each class.
(2) Train the LS one-class SVM with S using the pair of parameters $(C,\sigma) \in \{0.01, 0.1, 1.0, 10, 100\} \times \{0.1, 0.2, 0.4, 0.6, 1, 1.5\}$.
(3) Train the standard one-class SVM with S using the pair of parameters $(C,\sigma) \in \{1/(0.1n),\ 1/(0.3n),\ 1/(0.5n),\ 1/(0.7n),\ 1.0\} \times \{0.1, 0.2, 0.4, 0.6, 1, 1.5\}$, where n denotes the number of training points.
(4) Train the kernel Mean and kernel PCA with S using parameters $\sigma \in \{0.1, 0.2, 0.4, 0.6, 1, 1.5\}$.
(5) Rank all the data in the dataset, including S, according to the proximity measures defined for each method.
(6) Compute the average-precisions for each method in (19).

We did this procedure three times for each dataset, and took the pair of parameters for each method that produced the best average-precision. Table 1 shows the obtained parameters for each dataset and each method. In Table 1, LS and ST represent the LS one-class and the standard one-class SVM. The Mean and PCA denote the kernel versions.

With the selected parameters, we conducted two types of experiments, varying the number of training objects ranging from 10% to 70% of the population for each class. Due to the limit of space, we only show the results from 10% and 70% training objects. In the first type of experiment, after training, we ranked all the documents in the same dataset according to the proximity measures from the LS one-class SVM, the standard one-class SVM, the kernel Mean, and the kernel PCA, respectively. In the other type of experiment, we set aside the test dataset from the training dataset, and after training, we ranked the test dataset only. In both experiments, we computed the average-precision for each class and conducted the experiments four times and took the micro-average and macro-average (Chakrabarti, 2003) of the obtained average-precisions to compare the performances of the four different methods.

Tables 2 and 3 show the corresponding results from the first and second types of experiments with the three datasets, respectively. The best results are represented in boldface. Note that in case of 20NGs dataset the micro and macro-averages are equal since each class contains equal number of documents. In Table 2, one can see that the LS one-class SVM outperformed the other three methods. The standard one-class SVM outperformed the kernel Mean and the kernel PCA. It seems that the kernel PCA may not be suitable for the proximity measure problems. As the number of training objects increased, all the methods showed the performance improvement.

Table 3 shows the performances from the four approaches with the same conditions in Table 2 except the independent test dataset. As in the first experiment, the LS one-class SVM outperformed the three other methods. However, in general, the experiments with independent test dataset show relatively poor performances compared with the first experiments except the case of 70% training dataset with 20NGs and WebKB.

Table 1
Parameter selection for RBF kernel.

Data set	Parameter	LS	ST	Mean	PCA
20NGs	C	1	1/(0.3 N)	–	–
	σ	0.4	0.2	0.2	0.2
Reuter-21578	C	100	1/(0.5 N)	–	–
	σ	0.6	0.4	0.2	0.2
WebKB	C	100	1	–	–
	σ	0.4	0.4	0.4	0.2

Table 2
Micro and macro-average of average-precisions from testing whole dataset with the LS one-class, standard one-class SVM, kernel mean, and kernel PCA varying the number of training samples for 20ng, Reuter-21578, and WebKB data.

Data set	Measure	Train: 10%, Test: 100%				Train: 70%, Test: 100%			
		LS	ST	Mean	PCA	LS	ST	Mean	PCA
20NGs	Micro	**0.62**	0.55	0.54	0.53	**0.84**	0.80	0.76	0.74
	Macro	**0.62**	0.55	0.54	0.53	**0.84**	0.80	0.76	0.74
Reuter-21578	Micro	**0.82**	**0.82**	0.75	0.65	**0.93**	0.90	0.91	0.64
	Macro	**0.74**	0.73	0.63	0.49	**0.92**	0.88	0.89	0.49
WebKB	Micro	**0.47**	**0.47**	0.46	0.44	**0.87**	0.81	0.68	0.83
	Macro	**0.44**	0.43	0.43	0.40	***0.85***	0.80	0.69	0.82

Table 3
Micro and macro-average of average-precisions from testing independent dataset with the LS one-class, standard one-class SVM, kernel mean, and kernel PCA varying the number of training samples for 20ng, Reuter-21578, and WebKB data.

Data set	Measure	Train: 10%, Test: 90%				Train: 70%, Test: 30%			
		LS	ST	Mean	PCA	LS	ST	Mean	PCA
20NGs	Micro	**0.56**	0.48	0.47	0.46	**0.91**	0.88	0.82	0.82
	Macro	**0.56**	0.48	0.47	0.46	**0.91**	0.88	0.82	0.82
Reuter-21578	Micro	**0.73**	**0.73**	0.67	0.59	**0.81**	0.78	0.80	0.59
	Macro	**0.55**	0.54	0.48	0.38	**0.65**	0.62	0.64	0.39
WebKB	Micro	**0.37**	0.36	**0.37**	0.33	**0.99**	0.93	0.78	0.97
	Macro	**0.33**	0.32	**0.33**	0.29	**0.99**	0.94	0.81	0.97

In overall, one can see that the proposed LS one-class SVM outperformed the three other methods throughout the experiments. These experimental results indicate that the proposed LS one-class SVM is well suited for the proximity measure.

5. Conclusions

In this paper, we proposed the LS one-class SVM by reformulating the hyperplane-based standard one-class SVM (Schölkopf et al., 2001). The proposed method extracts the hyperplane which most of training objects may lie close to, whereas the standard one-class SVM extracts the hyperplane beyond which most of training objects may reside. The LS one-class SVM can be reduced to a kernel ridge regression with a single target value if we set the bias in the hyperplane to a predefined target value as in a regression model. This relates the LS one-class SVM to the kernel ridge regression and the one-class kernel Fisher discriminant analysis, and the kernel PCA.

We applied the proposed LS one-class SVM for relevance ranking which requires a proximity measure to determine which objects resemble better than others. The series of experiments with real datasets showed that the LS one-class SVM is better suited for proximity measure compared with other traditional methods.

Acknowledgements

We would like to thank Dr. Raghu Krishnapuram in IBM Research-India for his kind review and comments. This work was supported by the GRRC program of Gyeonggi province [2008-12-013, Knowledge Discovery for Ambient Service].

References

Chakrabarti, S., Van Den Berg, M., Dom, B., 1999. A new approach to topic-specific web resource discovery. Comput. Networks 31, 1623–1640.

Chakrabarti, S., 2003. Mining the Web: Discovering Knowledge from Hypertext Data. Morgan Kaufmann Publishers, San Francisco.

Chen, Y., Zhou, X.S., Huang, T.S., 2001. One-class SVM for learning in image retrieval. In: Proc. Internat. Conf. on Image Process., vol. 1, pp. 34–37.

Cristianini, N., Shawe-Taylor, J., 2000. An introduction to Support Vector Machines. Cambridge University Press.

Deng, H., Xu, R., 2007. Model selection for anomaly detection in wireless ad hoc networks. In: Proc. IEEE Symp. on Comput. Intell. Data Mining (CIDM 2007), pp. 540–546.

Hastie, T., Buja, A., Tibshirani, R., 1995. Penalized discriminant analysis. Ann. Statist. 23, 73–102.

Kailath, T., 1980. Linear Systems. Prentice & Hall, Englewood Cliffs, NJ.

Kruif, B.J., Vries, T.J.A., 2003. Pruning error minimization in least squares support vector machines. IEEE Trans. Neural Networks 14 (3).

Kuh, A., De Wilde, P., 2007. Comments on pruning error minimization in least squares support vector machines. IEEE Trans. Neural Networks 18 (2).

Ma, J., Perkins, S., 2003. Time-series novelty detection using one-class support vector machines. In: Proc. Internat. Joint Conf. on Neural Networks (IJCNN), pp. 1741–1745.

Manevitz, L.M., Yousef, M., 2001. One-class SVMs for document classification. J. Mach. Learning Res. 2, 139–154.

Müller, K.-R., Mika, S., Rätsch, G., Tsuda, K., Schölkopf, B., 2001. An Introduction to kernel-based learning algorithms. IEEE Trans. Neural Networks 12 (2), 181–201.

Roth, V., 2004. Outlier detection with one-class kernel Fisher discriminants. Advances in Neural Information Processing Systems, vol. 17. MIT Press.

Saunders, C., Gammerman, A., Vovk, V., 1998. Ridge regression learning algorithm in dual variables. In: Proc. 15th Internat. Conf. on Mach. Learning, pp. 515–521.

Schölkopf, B., Smola, A., Müller, K.-R., 1999. Kernel Principal Component Analysis, Advances in Kernel Methods: Support Vector Learning. MIT Press, Cambridge, MA. pp. 327–352.

Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., 2001. Estimating the support of a high-dimensional distribution. Neural Comput. 13 (7), 1443–1471.

Setia, L., Ick, J., Burkhardt, H., 2005. SVM-based relevance feedback in image retrieval using invariant feature histograms. In: Proc. IAPR Workshop on Mach. Vision Appl. (MVA 2005), pp. 16–18.

Suykens, J.A.K., Vandewalle, J., 1999. Least squares support vector machine classifiers. Neural Process. Lett. 9 (3), 293–300.

Suykens, J.A.K., Van Gestel, T., De Brabanter, J., De Moor, B., Vandewalle, J., 2002. Least Squares Support Vector Machines. World Scientific, Singapore.

Suykens, J.A.K., Van Gestel, T., Vandewalle, J., De Moor, B., 2003. A support vector machine formulation to PCA analysis and its kernel version. IEEE Trans. Neural Networks (2), 447–450.

Tax, D.M.J., Duin, R.P.W., 1999. Support vector domain description. Pattern Recognition Lett. 20, 1191–1199.

Tax, D.M.J., Duin, R.P.W., 2004. Support vector data description. Mach. Learning 54, 45–66.

Van Gestel, T., Suykens, J.A.K., Baesens, B., Viaene, S., Vanthienen, J., Dedene, G., De Moor, B., Vandewalle, J., 2004. Benchmarking least squares support vector machine classifiers. Mach. Learning 54, 5–32.

Vapnik, V., 1998. Statistical Learning Theory. John Wiley & Sons, New York.