TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 08/16 pp190-197
Volume 15, Number 2, April 2010

One-Class Support Vector Machine with Relative Comparisons*

GU Hong (顾 弘)**, ZHAO Guangzhou (赵光宙), QIU Jun (裘 君)†

College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China;
† Ningbo Institute of Technology, Zhejiang University, Ningbo 315000, China

Abstract: One-class support vector machines (one-class SVMs) are powerful tools that are widely used in many applications. This paper describes a semi-supervised one-class SVM that uses supervision in terms of relative comparisons. The analysis uses a hypersphere version of one-class SVMs with a penalty term appended to the objective function. The method simultaneously finds the minimum sphere in the feature space that encloses most of the target points and considers the relative comparisons. The result is a standard convex quadratic programming problem, which can be solved by adapting standard methods for SVM training, i.e., sequential minimal optimization. This one-class SVM can be applied to semi-supervised clustering and multi-classification problems. Tests show that this method achieves higher accuracy and better generalization performance than previous SVMs.

Key words: one-class support vector machines; semi-supervised learning; relative comparisons; clustering; multiclass classification

Introduction

One-class support vector machines (one-class SVMs) were first proposed for novelty detection[1] and high dimensional density estimation[2]. The two kinds of one-class SVMs are the hypersphere version and the hyperplane version. The hypersphere model, also known as the support vector domain description (SVDD)[1,3], tries to find a minimum surrounding sphere in the feature space that encloses most of the given points. The hyperplane model[2], on the other hand, separates the given points from the origin with the maximum margin. Both mechanisms transform the data into the feature space corresponding to a given kernel and use relaxation parameters to control the outliers. Originally, these two methods both belonged to the unsupervised learning category, with the target class points provided only (or positive samples). The semi-supervised versions employ negative samples to improve the performance. One method closely related to the hyperplane one-class SVMs is the transductive SVMs (TSVMs)[4,5], which focuses on the labeled data. TSVMs try to develop a large margin hyperplane classifier using labeled training data, while simultaneously forcing the hyperplane to be far from the unlabeled data.

Labeled data points are normally used as inequality constraints like $y_i(\boldsymbol{x}_i \cdot \boldsymbol{w} + b) - 1 \geqslant 0$ in SVMs for classification problems or are transformed to pairwise constraints (must-links and cannot-links) for clustering. This paper presents a new kind of constraint for relative comparisons. A relative comparison constraint is expressed by a triplet $(\boldsymbol{x}^1, \boldsymbol{x}^2, \boldsymbol{a})$, which means $\boldsymbol{a}$ is closer to $\boldsymbol{x}^1$ than to $\boldsymbol{x}^2$. In many applications, relative comparisons are believed to be more easily obtained than other types of constraints[6,7], especially in applications with user feedback. Another advantage of these constraints, which can be used to regularize the

Received: 2009-11-16; revised: 2010-01-21
* Supported by the National Natural Science Foundation of China (No. 60872070)
** To whom correspondence should be addressed.
E-mail: ghong@zju.edu.cn; Tel: 86-13588705645

local density, is that they indicate which points are closer to the focus point, or the clustering center in one-class data. For example, relevance feedback of content-based image retrieval naturally asks the user which instance is similar to the query even when they do not belong to the same class. Relative comparisons have gained much attention in metric learning, but little research has been done related to instance-based learning theories. This analysis considers these constraints in a one-class SVM. This may be the first study to integrate relative constraints into a one-class SVM. This method appends a new penalty term to the learning objective function, which leads to a convex quadratic programming (QP) problem. The tests demonstrate the performance of this method for clustering and multi-classification problems.

1 Related Works

In essence, a one-class SVM can be regarded as a density estimator[2,3,8]. However, there are various applications of one-class SVMs. Schölkopf et al.[9] applied a one-class SVM to novelty detection. They trained a one-class SVM with normal target data and then used the SVM to detect novel data that deviated from the learned classifier. Manevitz and Yousef[10] compared the real performance of one-class SVMs for document classification problems. They showed that the one-class SVMs outperform all other methods except the neural network one.

Ben-Hur et al.[11] presented an unsupervised learning method called support vector clustering (SVC) that yields good results for arbitrary geometrical shapes. SVC is based on the observation that given a pair of data points that belong to different components (clusters), any path that connects them must exit from the sphere in the feature space. The learned hypersphere, when mapped back to the data space, can separate into several components, each enclosing a separate cluster of points. The disadvantage of SVC is that it can hardly get the prototype of each cluster. Recent studies have thus preferred using the multiple sphere representation[12-14]. The common basis of these approaches is to train a one-class SVM for each class or cluster. Inspired by metric learning, Wang et al.[15] proposed a structured one-class SVM training algorithm with constraints expressed by $(\boldsymbol{x}-\boldsymbol{a})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{a})\leqslant R$ rather than $\|(\boldsymbol{x}-\boldsymbol{a})\|^2\leqslant R$ where $\boldsymbol{\Sigma}$ is computed like RCA[16] in a cluster.

Relative comparisons were first used to study the Mahalanobis distance[7] and for SVaD measures[6]. Both results show that their methods take advantage of the relative comparisons and outperform state-of-art methods using pairwise constraints. Frome et al.[17] proposed an attractive large-margin distance learning algorithm based on relative comparisons. They trained the local distance functions in an SVM-like optimization framework. Another work related to Frome et al.[17] is the CLAD measures proposed by Krishna et al.[18] Krishna et al.[18] used the linear accumulative error model and gradient-based search. The local image distance and CLAD measures were used as the asymmetric distance functions. The main difference between these two methods is the role of the relative comparisons. In Krishna et al.[18], the comparisons were used to generate the objective function which was the accumulative error sum. Frome et al.[17] used the comparisons as constraints in a convex optimization problem. The advantage of this method is the global-consistent of the trained distance functions.

The semi-supervised learning improves the learning results by incorporating some supervision information. Generally speaking, there are two approaches to achieve this goal. The first approach is to learn the metric or kernel functions[7,16-18]. The other approach uses constraint-based approaches[5,19,20]. One popular semi-supervised clustering algorithm that combines metric learning and constraint-based approach is the HMRF-KMeans proposed by Sugato et al.[20], where pairwise constraints are considered. Semi-supervised learning based on one-class SVMs has not been well studied. Tax and Duin[3] gives the original solution when there are negative samples in the training set. However, this semi-supervised form of the SVDD is rarely used in real applications, even in classification problems. As mentioned before, relative comparisons have been investigated in metric learning algorithms but not in other types of algorithms. This paper shows how these constraints can be integrated into one-class SVMs.

2 One-Class SVMs with Relative Comparisons

Given the training dataset $\boldsymbol{X}=\{\boldsymbol{x}_i\},i=1,\ldots,N$, first introduce the relative comparison triplet $\tau(\boldsymbol{x}^1,\boldsymbol{x}^2,\boldsymbol{a})$,

where $\boldsymbol{x}^1 \in \boldsymbol{X}$ and $\boldsymbol{x}^2 \in \boldsymbol{X}$ are points in the data space and $\boldsymbol{a}$ represents the hypersphere center of the target points. The triplet $\tau(\boldsymbol{x}^1, \boldsymbol{x}^2, \boldsymbol{a})$ means

$$\tau(\boldsymbol{x}^1, \boldsymbol{x}^2, \boldsymbol{a}) \rightarrow \|\boldsymbol{x}^1 - \boldsymbol{a}\|^2 \leqslant \|\boldsymbol{x}^2 - \boldsymbol{a}\|^2 \quad (1)$$

in the linear feature space. Assume a set of triplets $\boldsymbol{T} = \{\tau(\boldsymbol{x}^1, \boldsymbol{x}^2, \boldsymbol{a})_k\}, k = 1, \ldots, M$, then the objective is

$$\begin{aligned} &\min R^2 + C_\xi \sum_i \xi_i + C_\gamma \sum_k \gamma_k \\ &\text{s.t.} \quad \|\boldsymbol{x}_i - \boldsymbol{a}\|^2 \leqslant R^2 + \xi_i, \quad \forall i, \xi_i \geqslant 0, \\ &\quad \|\boldsymbol{x}_k^1 - \boldsymbol{a}\|^2 \leqslant \|\boldsymbol{x}_k^2 - \boldsymbol{a}\|^2 + \gamma_k, \ \forall \tau(\boldsymbol{x}^1, \boldsymbol{x}^2, \boldsymbol{a})_k, \quad \gamma_k \geqslant 0 \end{aligned} \quad (2)$$

where ξ_i and γ_k are slack variables and C_ξ and C_γ are penalty regularization parameters. The basic idea of the objective is similar to that of SVDD, to find a minimum sphere enclosing most of the target data points. There are two different kinds of constraints in Eq. (2). The first is the same as the original one in Tax and Duin[3]. The second kind, $\|\boldsymbol{x}_k^1 - \boldsymbol{a}\|^2 \leqslant \|\boldsymbol{x}_k^2 - \boldsymbol{a}\|^2 + \gamma_k$, corresponding to the penalty term $C_\gamma \sum_k \gamma_k$, is the additional relative comparison. The objective then tries to find the most feasible model that fulfills these conditions with $\boldsymbol{x}_k^1$ closer to the sphere center than $\boldsymbol{x}_k^2$. Note that the metric regularization form described in Wang et al.[15] can be easily extended to this framework.

Equation (2) is a standard convex quadratic programming problem which can be solved using the Lagrangian:

$$\begin{aligned} L(R, \boldsymbol{a}, \xi_i, \gamma_j, \alpha_i, \beta_i, \chi_k, \eta_k) &= R^2 + C_\xi \sum_i \xi_i + C_\gamma \sum_k \gamma_k - \\ &\sum_i \alpha_i \{R^2 + \xi_i - (\boldsymbol{x}_i \cdot \boldsymbol{x}_i - 2\boldsymbol{a} \cdot \boldsymbol{x}_i + \boldsymbol{a} \cdot \boldsymbol{a})\} - \sum_i \beta_i \xi_i - \\ &\sum_k \chi_k \{(\boldsymbol{x}_k^2 \cdot \boldsymbol{x}_k^2 - 2\boldsymbol{a} \cdot \boldsymbol{x}_k^2) + \gamma_k - (\boldsymbol{x}_k^1 \cdot \boldsymbol{x}_k^1 - 2\boldsymbol{a} \cdot \boldsymbol{x}_k^1)\} - \\ &\sum_k \eta_k \gamma_k \end{aligned} \quad (3)$$

For simplicity, the symbol $\sum_i$ is used here to traverse all points in $\boldsymbol{X}$ and $\sum_k$ is used to traverse all triplets in $\boldsymbol{T}$. $\alpha_i, \beta_i, \chi_k$, and η_k are Lagrangian multipliers associated with the variables in Eq. (2) which satisfies $\alpha_i \geqslant 0$, $\beta_i \geqslant 0$, $\chi_k \geqslant 0$, $\eta_k \geqslant 0$.

From the condition

$$\frac{\partial L}{\partial R} = \frac{\partial L}{\partial \boldsymbol{a}} = \frac{\partial L}{\partial \xi_i} = \frac{\partial L}{\partial \gamma_k} = 0 \quad (4)$$

we get

$$\sum_i \alpha_i = 1 \quad (5)$$

$$\boldsymbol{a} = \sum_i \alpha_i \boldsymbol{x}_i + \sum_k \chi_k (\boldsymbol{x}_k^1 - \boldsymbol{x}_k^2) \quad (6)$$

$$\alpha_i = C_\xi - \beta_i, \ \forall i \quad (7)$$

$$\chi_k = C_\gamma - \gamma_k, \ \forall k \quad (8)$$

Using Eqs. (5)-(8), the solution of Eq. (2) can be obtained by maximizing the Wolfe dual form of the Lagrangian in Eq. (3):

$$\begin{aligned} W = &\sum_i \alpha_i (\boldsymbol{x}_i \cdot \boldsymbol{x}_i) - \sum_i \sum_j \alpha_i \alpha_j (\boldsymbol{x}_i \cdot \boldsymbol{x}_j) + \\ &\sum_k \chi_k \{(\boldsymbol{x}_k^1 \cdot \boldsymbol{x}_k^1) - (\boldsymbol{x}_k^2 \cdot \boldsymbol{x}_k^2)\} - \\ &\sum_k \sum_s \chi_k \chi_s \{(\boldsymbol{x}_k^1 \cdot \boldsymbol{x}_s^1) - (\boldsymbol{x}_k^1 \cdot \boldsymbol{x}_s^2) - (\boldsymbol{x}_k^2 \cdot \boldsymbol{x}_s^1) + (\boldsymbol{x}_k^2 \cdot \boldsymbol{x}_s^2)\} - \\ &2 \sum_i \sum_k \alpha_i \chi_k \{(\boldsymbol{x}_k^1 \cdot \boldsymbol{x}_i) - (\boldsymbol{x}_k^2 \cdot \boldsymbol{x}_i)\} \end{aligned} \quad (9)$$

with the following constraints:

$$\begin{aligned} &\sum_i \alpha_i = 1, \\ &0 \leqslant \alpha_i \leqslant C_\xi, \ \forall i, \\ &0 \leqslant \chi_k \leqslant C_\gamma, \ \forall k \end{aligned} \quad (10)$$

Finally, the distance from a point to the learned data center is given by

$$\begin{aligned} D^2(\boldsymbol{x}) = &\|\boldsymbol{x} - \boldsymbol{a}\|^2 = \\ &(\boldsymbol{x} \cdot \boldsymbol{x}) - 2 \sum_i \alpha_i (\boldsymbol{x} \cdot \boldsymbol{x}_i) + \sum_i \sum_j \alpha_i \alpha_j (\boldsymbol{x}_i \cdot \boldsymbol{x}_j) - \\ &2 \sum_k \chi_k \{(\boldsymbol{x} \cdot \boldsymbol{x}_k^1) - (\boldsymbol{x} \cdot \boldsymbol{x}_k^2)\} + \sum_k \sum_s \chi_k \chi_s \{(\boldsymbol{x}_k^1 \cdot \boldsymbol{x}_s^1) - \\ &(\boldsymbol{x}_k^1 \cdot \boldsymbol{x}_s^2) - (\boldsymbol{x}_k^2 \cdot \boldsymbol{x}_s^1) + (\boldsymbol{x}_k^2 \cdot \boldsymbol{x}_s^2)\} \end{aligned} \quad (11)$$

where i and j are used to traverse all points in $\boldsymbol{X}$ and k and s are used to traverse all triplets in $\boldsymbol{T}$. The radius of the hypersphere is given by $R = D(\boldsymbol{x}^{\text{sv}})$ for any point belonging to the spherical support vectors.

The one-class SVM works in the feature space by replacing the dot product $(\boldsymbol{x}_i \cdot \boldsymbol{x}_j)$ with a kernel function $g(\cdot, \cdot)$:

$$g(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Phi(\boldsymbol{x}_i) \cdot \Phi(\boldsymbol{x}_j) \quad (12)$$

Typically, when the Gaussian kernel is used, the one-class SVM can model arbitrary cluster shapes of the data.

3 Training the One-Class SVMs

Sequential minimal optimization (SMO)[21] was used to solve the quadratic optimization problem (QP). SMO breaks up the constrained minimization of Eq. (9) into a set of local optimization steps. In this setting,

both α and χ need to be optimized. The basic optimizing step is as follows:

$$\Delta=\alpha_1^{\text{new}}+\alpha_2^{\text{new}},\alpha_2^{\text{new}}=\alpha_2+\frac{D^2(\boldsymbol{x}_2)-D^2(\boldsymbol{x}_1)}{2\Gamma(\boldsymbol{x}_1,\boldsymbol{x}_2)},$$

$$\alpha_1^{\text{new}}=\Delta-\alpha_2^{\text{new}} \tag{13}$$

$$\chi_k^{\text{new}}=\chi_k+\frac{D^2(\boldsymbol{x}_k^1)-D^2(\boldsymbol{x}_k^2)}{2\Gamma(\boldsymbol{x}_k^1,\boldsymbol{x}_k^2)} \tag{14}$$

where $\Gamma(\boldsymbol{x}_1,\boldsymbol{x}_2)=\boldsymbol{x}_1\cdot\boldsymbol{x}_1+\boldsymbol{x}_2\cdot\boldsymbol{x}_2-2\boldsymbol{x}_1\cdot\boldsymbol{x}_2$. In the training process, any point that deviates from the Karush-Kuhn-Tucker (KKT) conditions of QP in Eq. (2) is selected and optimized. The KKT for Eq. (2) is

$$\begin{cases}\alpha_i\{R^2+\xi_i-(\boldsymbol{x}_i\cdot\boldsymbol{x}_i-2\boldsymbol{a}\cdot\boldsymbol{x}_i+\boldsymbol{a}\cdot\boldsymbol{a})\}=0 \ \ \forall i,\\ \beta_i\xi_i=0 \ \ \forall i,\\ \chi_k\{(\boldsymbol{x}_k^2\cdot\boldsymbol{x}_k^2-2\boldsymbol{a}\cdot\boldsymbol{x}_k^2)+\gamma_k-\\ \qquad(\boldsymbol{x}_k^1\cdot\boldsymbol{x}_k^1-2\boldsymbol{a}\cdot\boldsymbol{x}_k^1)\}=0 \ \ \forall k,\\ \eta_k\gamma_k=0 \ \ \forall k\end{cases} \tag{15}$$

Points corresponding to $0<\alpha_i<C_\xi$ are called support vectors (SVs) during the optimization. They lie on the spherical surface of the hypersphere, so can be referred to as spherical SVs as well. Points corresponding to $\alpha_i=0$ and $\alpha_i=C_\xi$ are called inner points and boundary support vectors (BSVs). Analogously, the relative comparison triplets with $\chi_k=0$ are called legal triplets while the triplets with $0<\chi_k<C_\gamma$ are called critical support triplets (CSTs) and triplets with $\chi_k=C_\gamma$ are deviation support triplets (DSTs). After the optimization, the inner points and legal triplets are not needed for the classifier since only the terms with $\alpha_i\neq 0$ and $\chi_k\neq 0$ need to be restored in Eq. (11). During the iterations, the value of α is reduced if it violates Eq. (10) or Eq. (13). The general strategy is to first ensure the optimization of α first. Once all the α satisfy the KKT, χ is optimized.

4 Application: Semi-Supervised Clustering

Semi-supervised clustering aims to improve the clustering results using limited supervision. This has become a topic of significant interest in the recent machine learning literature. Existing methods take advantage of metric learning and background or feedback constraints. Sugato et al.[20] proposed a probabilistic framework based on hidden Markov random fields (HMRFs) for semi-supervised clustering that combines the constraint-based and distance-based approaches. An extended kernel version of HMRF-KMeans, known as SS-KERNEL-KMeans was presented by Kulis et al.[19] HMRF-KMeans and SS-KERNEL-KMeans use pairwise constraints. SSSVaD[6] is another recently proposed semi-supervised clustering algorithm. SSSVaD learns the Mahalanobis distance using relative comparisons which gives better performance than HMRF-KMeans. Xing et al.[22] described a convex programming approach to learn the Mahalanobis distance for K-Means with side information. Other methods have also been used in Refs. [23-25].

Kernel Grower[13,26] is a new unsupervised clustering algorithm based on one-class SVMs and KMeans. The present method adopts a similar mechanism but uses a generative model and the semi-supervised one-class SVM. At the beginning of the clustering, some of the data points are selected to construct the initial clusters. For each cluster, a one-class SVM is trained with some relative comparisons generated from prior knowledge. Formally, let $\mathbf{CL}=\{\mathrm{CL}_1,\mathrm{CL}_2,\ldots,\mathrm{CL}_Z\}$ be the clusters and the data points $\boldsymbol{X}=\{\boldsymbol{x}_i\}$. The points reassignment rule is assumed to be $\arg\max_z p(\mathrm{CL}_z\,|\,\boldsymbol{x}_i)$, where

$$p(\mathrm{CL}_z\,|\,\boldsymbol{x}_i)=\frac{p(\boldsymbol{x}_i\,|\,\mathrm{CL}_z)p(\mathrm{CL}_z)}{p(\boldsymbol{x}_i)} \tag{16}$$

Using the Bayesian rule, Eq. (16) can be rewritten as

$$p(\mathrm{CL}_z\,|\,\boldsymbol{x}_i)=\frac{p(\boldsymbol{x}_i\,|\,\mathrm{CL}_z)p(\mathrm{CL}_z)}{\sum_{j=1,\ldots,Z}p(\boldsymbol{x}_i\,|\,\mathrm{CL}_j)p(\mathrm{CL}_j)} \tag{17}$$

For a given cluster, calculate $p(\boldsymbol{x}_i\,|\,\mathrm{CL}_z)$ by

$$p(\boldsymbol{x}_i\,|\,\mathrm{CL}_z)=\max(R_{\mathrm{CL}_z}^2-D_{\mathrm{CL}_z}^2(\boldsymbol{x}_i),0) \tag{18}$$

where R_{CL_z} denotes the radius of the learned z-th hypersphere and $D_{\mathrm{CL}_z}(\boldsymbol{x})$ denotes the distance from the sphere center to $\boldsymbol{x}$, Eq. (11). One issue in the clustering algorithm is the generation of the triplets set $\boldsymbol{T}=\{\tau(\boldsymbol{x}^1,\boldsymbol{x}^2,\boldsymbol{a})_k\},k=1,\ldots,M$. The format used here differs from ones described in Refs. [6,7]. Given a cluster CL_z, dynamically transform $(\boldsymbol{x}^1,\boldsymbol{x}^2,\boldsymbol{x}^3)$ to the present format. If $\boldsymbol{x}^1$ and $\boldsymbol{x}^2$ belong to CL_z, construct two triplets $\tau(\boldsymbol{x}^1,\boldsymbol{x}^3,\boldsymbol{a}_z)$ and $\tau(\boldsymbol{x}^2,\boldsymbol{x}^3,\boldsymbol{a}_z)$. A prior label does not need to be assigned to start the clustering, with this simple heuristic working well in the tests.

Two datasets were used to validate the system performance. Letters-IJL is a subset of the Letter Recognition Dataset (Letters) in the UCI data repository[27]. Letters contains 20 000 characters, each a 16-dimensional vector. A subset of 10% of the data points was chosen randomly from three classes $\{I, J, L\}$ of the Letters. The second dataset was Protein[22], which contains 20-dimensional feature vectors representing 116 proteins from 6 classes. The clustering results were evaluated using the standard normalized mutual information (NMI) defined as

$$\mathrm{NMI} = \frac{I(U,V)}{(H(U)+H(V))/2} \tag{19}$$

where U is a random variable denoting the cluster assignments of the points and V is a random variable denoting the underlying class labels of the points. $I(U,V) = H(U) - H(U|V)$ is the mutual information between the random variables U and V. $H(U)$ is the Shannon entropy of U and $H(U|V)$ is the conditional entropy of U given V. More information about the NMI measures can be found in Ref. [28].

The experiments used 2-fold cross validation. The data points were separated into two groups for training and for testing. Figures 1 and 2 show the average clustering results for Letters-IJL and Protein. The present method is referred as to the semi-supervised K multi-sphere support vector clustering (SS-KMSVC) method in the figures. The clustering results of some other algorithms are also given. All the KMeans-based algorithms found in the literature suffer from a dependence on the initialization algorithm. Therefore, to get a fair comparison, all the methods used the neighborhood inference algorithm provided by Sugato et al.[20] to initialize the clusters. The SS-KMSVC used the Gaussian kernel $g(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp(-q \cdot \| \boldsymbol{x}_1 - \boldsymbol{x}_2 \|^2)$. The width parameter q was adjusted to get the best performance, typically 1.2E-3 for Letters-IJL and 2.5E-3 for Protein. The penalty regularization parameters C_ξ and C_γ were both set to 1 in the present method. These two variables are robust to the clustering results. In the testing with very few relative comparison triplets are the DSTs where most χ_k were less than 0.1.

The semi-supervised clustering results obtained by the SSSVaD[6] algorithm consistently outperform the other methods when there is a small number of constraints as shown in Figs. 1 and 2. However, other

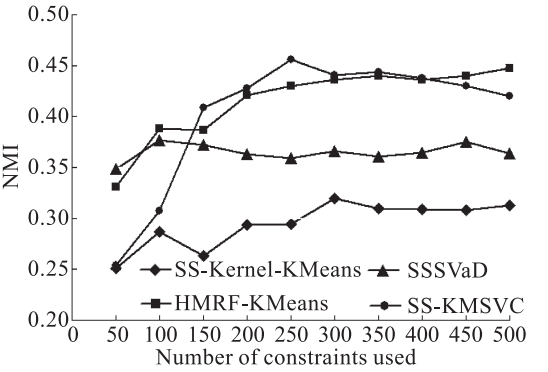

Fig. 1 Clustering results for Letters-IJL

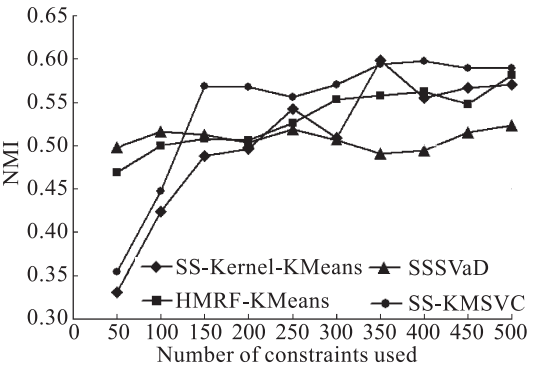

Fig. 2 Clustering results for Protein

methods had better performance than SSSVaD with a large number of constraints. The kernel methods, SS-KERNEL-KMeans did poorly on the Letters-IJL dataset, and yielded unstable results on the Protein dataset. The clustering results obtained by HMRF-KMeans are very good on both datasets, with better performance as the number of constraints increases. The present method achieves the best overall performance for a modest number of constraints. With the Protein dataset, SS-KMSVC did significantly better with more than 200 constraints. On the Letters-IJL dataset, the best clustering result occurred with about 250 constraints. The performance decreases slightly as the number of constraints increases due to the triplets conversion collisions and overfitting.

5 Application: Multi-Classification

This section applies the one-class SVMs for multi-classification problems. Generally speaking, the two basic strategies to solve multi-classification problem are to classify all the data at once based on a posterior probability and to use binary classification methods and then combine the binary classification results. A well known algorithm is the one-versus-one

method (also referred to as the pairwise method) based on SVMs. If there are z classes, the pairwise method takes $z(z-1)/2$ two-class SVMs to construct the final classification tree when the whole problem is decomposed to a set of binary classification problems. Another popular algorithm is the one-versus-all (OVA) method. A one-class classifier is trained by classifying points outside of this class as negative samples, thus z SVMs need be trained for the z class problem. Lee and Lee[14] proposed an OVA method based on the Bayesian optimal decision theory and SVDD. For multi-classification problems, they used the basic SVDD to train a one-class classifier for each class using only the target (positive) data points during the training. The final multiclass classifier was constructed by maximizing the pseudo posterior probability function:

$$\arg\max_{z=1,\ldots,Z} p(\mathrm{CL}_z)\cdot p(\boldsymbol{x}\,|\,\mathrm{CL}_z) \tag{20}$$

where $p(\mathrm{CL}_z)$ is the priori and $p(\boldsymbol{x}\,|\,\mathrm{CL}_z)$ is the density function obtained in Eq. (18).

The current tests follow the tests in Lee and Lee[14]. The five datasets used in the tests were Iris, Vehicle, Vowel, Segment, and Letters. All these datasets can be freely downloaded from the UCI machine learning repository. The generation of relative comparisons is quite straightforward. Suppose that a one-class SVM classifier is trained for class A with m positive points in A and other n negative points (belong to other classes) to generate $m\times n$ triplets. This can be a very large number for some datasets; however, not all the triplets are needed. In general, we randomly select $(m+n)$ triplets as described in the following. The parameter q is set to the same value as in Lee and Lee[14] and C_ξ and C_γ are set to 1. The test results are shown in Table 1 which indicates that the current method outperforms that of Lee and Lee[14] on most datasets except for Segment, with a slightly higher error rate.

Table 1 Multi-classification test results for several datasets based on the error rate (%). The Vehicle dataset was normalized while the others were not. The parameter q for the five datasets was set to 3, 600, 0.9, 0.0037, and 0.0012. The boldface terms are the best results (the lowest error rates) for each dataset and the term in the parentheses is the differences of error rate between the present method and Lee and Lee[14]. LDA and QDA denote linear discriminant analysis and quadratic discriminant analysis[29].

Data	Error rate (%)					
	LDA	QDA	1-1 SVMs	1-all SVMs	Lee's[14]	Our method
Iris	**2**	**2**	6	10	**2**	**2** (↓0)
Vehicle	22.7	**20.21**	39.36	33.33	30.14	23.05 (↓7.09)
Vowel	55.63	52.81	38.74	34.85	37.23	**30.7** (↓6.53)
Segment	8.83	22.21	13.25	6.88	**4.94**	5.03 (↑0.09)
Letters	30.14	11.28	6.2	N/A	4.92	**4.76** (↓0.16)

As mentioned, many relative comparisons can be generated using all the original data points. The results in Fig. 3 show this effect for the Vehicle dataset. The error rate decreased for less than 800 relative comparisons and then increased slightly. The average number of CSTs (ANC) for each Vehicle class show that the class Opel had the most CSTs, about 10% of the number of generalized relative comparisons, with Saab having the next highest. Classes Van and Bus have small numbers of CSTs which shows good discrimination of the features. For the hard classification classes like Opel and Saab, the number of CSTs increases with the number of relative comparison triplets. However, with too many triplets the training process is time-consuming and does not bring the desired performance boosting, as shown in Table 2 which gives a detailed comparison of the training times needed for increasing numbers of relative comparisons. A compromise is to use about $(m+n)$ triplets in the training which gives better results without excessively long training times.

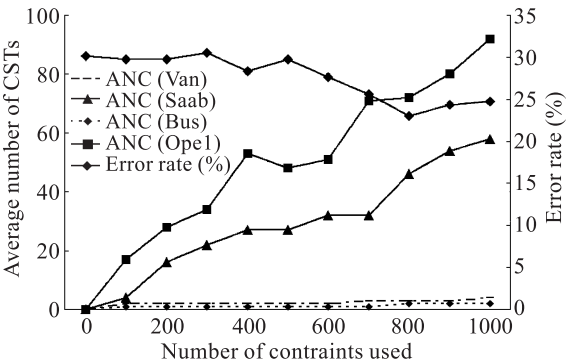

Fig. 3 Multi-classification results for the Vehicle dataset for various numbers of relative comparisons

Table 2 Training times for various numbers of relative comparisons on dataset Vehicle

Number of constraints used	Training time (ms)
0	1010
100	1964
500	10 511
800	22 798
1000	28 819

6 Conclusions

A one-class SVM was developed for relative comparisons as perhaps the first to incorporate this type of constraint into one-class SVMs. The SVM was then used for semi-supervised clustering and multiclass classification problems. The method yields better results than other state-of-the-art algorithms. The results shown in Section 4 suggest that the metric learning plays an important role for learning global attributes of feature vectors while the constraint-based methods are more suitable for regularizing the local density. Further work is needed to integrate metric learning algorithms into one-class SVMs.

References

[1] Tax D M J, Duin R P W. Support vector domain description. *Pattern Recognition Letters*, 1999, **20**(11-13): 1191-1199.

[2] Scholkopf B, Platt J C, Shawe-Taylor J, et al. Estimating the support of a high-dimensional distribution. *Neural Computation*, 2001, **13**(7): 1443-1471.

[3] Tax D M J, Duin R P W. Support vector data description. *Machine Learning*, 2004, **54**(1): 45-66.

[4] Vapnik V. The Nature of Statistical Learning Theory. Springer, 1995.

[5] Collobert R, Sinz F, Weston J, et al. Large scale transductive SVMs. *The Journal of Machine Learning Research*, 2006, **7**(1): 1712.

[6] Kumar N, Kummamuru K, Paranjpe D. Semisupervised clustering with metric learning using relative comparisons. *IEEE Transactions on Knowledge and Data Engineering*, 2008, **20**(4): 496-503.

[7] Schultz M, Joachims T. Learning a distance metric from relative comparisons. In: Proceeding of the Advances in Neural Information Processing Systems. Vancouver, Canada: The MIT Press, 2004: 41.

[8] Vert R, Vert J P. Consistency and convergence rates of one-class SVMs and related algorithms. *The Journal of Machine Learning Research*, 2006, **7**(1): 854.

[9] Scholkopf B, Williamson R C, Smola A J, et al. Support vector method for novelty detection. *Advances in Neural Information Processing Systems*, 2000, **12**(4): 582-588.

[10] Manevitz L M, Yousef M. One-class SVMs for document classification. *The Journal of Machine Learning Research*, 2002, **2**(1): 154.

[11] Ben-Hur A, Horn D, Siegelmann H T, et al. Support vector clustering. *The Journal of Machine Learning Research*, 2002, **2**: 125-137.

[12] Chiang J H, Hao P Y. A new kernel-based fuzzy clustering approach: Support vector clustering with cell growing. *IEEE Transactions on Fuzzy Systems*, 2003, **11**(4): 518-527.

[13] Camastra F, Verri A. A novel kernel method for clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005, **3**(1): 801-805.

[14] Lee D, Lee J. Domain described support vector classifier for multi-classification problems. *Pattern Recognition*, 2007, **40**(1): 41-51.

[15] Wang D, Yeung D S, Tsang E C C. Structured one-class classification. *IEEE Transactions on Systems, Man, and Cybernetics*, *Part B*, 2006, **36**(6): 1283-1295.

[16] Bar-Hillel A, Hertz T, Shental N, et al. Learning a mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research*, 2006, **6**(1): 937.

[17] Frome A, Singer Y, Sha F, et al. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In: Proceeding of the Eleventh IEEE International Conference on Computer Vision (ICCV' 07). Rio de Janeiro, Brazil: IEEE, 2007: 1255-1263.

[18] Krishna K, Raghu K, Rakesh A. On learning asymmetric dissimilarity measures. In: Proceedings of the Fifth IEEE International Conference on Data Mining. Washington, DC, USA, 2005: 697-700.

[19] Kulis B, Basu S, Dhillon I, et al. Semi-supervised graph clustering: A kernel approach. *Machine Learning*, 2009, **74**(1): 1-22.

[20] Sugato B, Mikhail B, Raymond J M. A probabilistic framework for semi-supervised clustering. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Seattle, WA, USA, 2004: 59-68.

[21] Platt J. Sequential minimal optimization: A fast algorithm for training support vector machines. *Advances in Kernel*

Methods-Support Vector Learning, 1999, **208**: 41-65.

[22] Xing E P, Ng A Y, Jordan M I, et al. Distance metric learning with application to clustering with side-information. *Advances in Neural Information Processing Systems*, 2003: 521-528.

[23] Cohn D, Caruana R, Mccallum A. Semi-supervised clustering with user feedback. *Constrained Clustering*: *Advances in Algorithms*, *Theory*, *and Applications*, 2003, **4**(1): 17.

[24] Chapelle O, Scholkopf B, Zien A. Semi-Supervised Learning. Cambridge, MA, USA: The MIT Press, 2006.

[25] Grira N, Crucianu M, Boujemaa N. Unsupervised and semi-supervised clustering: A brief survey. A Review of Machine Learning Techniques for Processing Multimedia Content. Tech. Rep. The MUSCLE European Network of Excellence (FP6), 2004.

[26] Chang L, Deng X M, Zheng S W, et al. Scaling up kernel grower clustering method for large data sets via core-sets. *Acta Automatica Sinica*, 2008, **34**(3): 376-382.

[27] Asuncion A, Newman D J. {UCI} Machine Learning Repository. http://www.ics.uci.edu/~mlearn/MLRepository.html. Irvine, CA: University of California, School of Information and Computer Science, 2007.

[28] Strehl A, Ghosh J, Mooney R. Impact of similarity measures on web-page clustering. In: Proceedings of Workshop on Artificial Intelligence for Web Search (AAAI 2000). AAAI, 2000: 58-64.

[29] Hastie T, Tibshirani R, Friedman J, et al. The Elements of Statistical Learning (2nd edition). The Mathematical Intelligencer, Springer-Verlag, 2008.