Routledge
Taylor & Francis Group

TECHNOLOGICAL AND ECONOMIC DEVELOPMENT OF ECONOMY
ISSN 2029-4913 print/ISSN 2029-4921 online

2012 Volume 18(1): 5–33
doi:10.3846/20294913.2012.661205

Review

RECENT ADVANCES ON SUPPORT VECTOR MACHINES RESEARCH

Yingjie Tian[1], Yong Shi[2], Xiaohui Liu[3]

[1]Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, No. 80 Zhongguancun East Road, Haidian District, Beijing 100190, China
[2]College of Information Science and Technology, University of Nebraska at Omaha, Omaha, NE 68182, USA
[3]School of Information Systems, Computing and Mathematics, Brunel University, Uxbridge, Middlesex, UK
E-mails: [1]tyj@gucas.ac.cn (corresponding author); [2]yshi@gucas.ac.cn; [3]xiaohui.liu@brunel.ac.uk

Received 05 September 2011; accepted 19 December 2011

Abstract. Support vector machines (SVMs), with their roots in Statistical Learning Theory (SLT) and optimization methods, have become powerful tools for problem solution in machine learning. SVMs reduce most machine learning problems to optimization problems and optimization lies at the heart of SVMs. Lots of SVM algorithms involve solving not only convex problems, such as linear programming, quadratic programming, second order cone programming, semi-definite programming, but also non-convex and more general optimization problems, such as integer programming, semi-infinite programming, bi-level programming and so on. The purpose of this paper is to understand SVM from the optimization point of view, review several representative optimization models in SVMs, their applications in economics, in order to promote the research interests in both optimization-based SVMs theory and economics applications. This paper starts with summarizing and explaining the nature of SVMs. It then proceeds to discuss optimization models for SVM following three major themes. First, least squares SVM, twin SVM, AUC Maximizing SVM, and fuzzy SVM are discussed for standard problems. Second, support vector ordinal machine, semi-supervised SVM, Universum SVM, robust SVM, knowledge based SVM and multi-instance SVM are then presented for nonstandard problems. Third, we explore other important issues such as l_p-norm SVM for feature selection, LOOSVM based on minimizing LOO error bound, probabilistic outputs for SVM, and rule extraction from SVM. At last, several applications of SVMs to financial forecasting, bankruptcy prediction, credit risk analysis are introduced.

Keywords: support vector machines (SVMs), optimization, machine learning (ML), data mining (DM), financial forecasting, bankruptcy prediction, credit risk analysis.

Reference to this paper should be made as follows: Tian, Y.; Shi, Y.; Liu, X. 2012. Recent advances on support vector machines research, *Technological and Economic Development of Economy* 18(1): 5–33.

JEL Classification: C02, C38, C53, C61.

1. Introduction

Support vector machines (SVMs), which were introduced by Vapnik and his coworkers in the early 1990's (Cortes, Vapnik 1995; Vapnik 1996, 1998), are proved to be effective and promising techniques for data mining (Peng *et al.* 2008; Yang, Wu 2006). There are three essential elements making SVMs so successful: the principle of maximal margin, dual theory, and kernel trick. SVMs, unlike traditional methods (e.g. Neural Networks), having their roots in Statistical Learning Theory (SLT) and optimization methods, become powerful tools to solve the problems of machine learning with finite training points and overcome some traditional difficulties such as the "curse of dimensionality", "over-fitting" and etc. SVMs' theoretical foundation and implementation techniques have been established and SVMs are gaining quick development and popularity due to a number of their attractive features: nice mathematical representations, geometrical explanations, good generalization abilities and promising empirical performance (Cristianini, Shawe-Taylor 2000; Deng, Tian 2004, 2009; Deng *et al.* 2012; Herbrich 2002; Schölkopf, Smola 2002). They have been successfully applied in many fields ranging from text categorization (Joachims 1999a; Lodhi *et al.* 2000), face detection, verification, and recognition (Jonsson *et al.* 2002; Lu *et al.* 2001; Tefas *et al.* 2001), speech recognition (Ganapathiraju *et al.* 2004; Ma *et al.* 2001), to bioinformatics (Guyon *et al.* 2001; Zhou, Tuck 2006), bankruptcy prediction (Shin *et al.* 2005), remote sensing image analysis (Melgani, Bruzzone 2004), time series forecasting (Kim 2003; Tay, Cao 2001), information and image retrieval (Druker *et al.* 2001; Liu *et al.* 2007; Tian *et al.* 2000), information security (Mukkamala *et al.* 2002) and etc. (Adankon, Cheriet 2009; Ancona *et al.* 2001; Azimi-Sadjadi, Zekavat 2000; Borgwardt 2011; Gutta *et al.* 2000; Peng *et al.* 2009; Schweikert *et al.* 2009; Yao *et al.* 2002).

In recent years, the fields of machine learning and mathematical programming are increasingly intertwined (Bennett, Parrado-Hernández 2006), in which SVMs are the typical representatives. SVMs reduce most machine learning problems to optimization problems, optimization lies at the heart of SVMs, especially the convex optimization problem plays an important role in SVMs. Since convex problems are much more tractable algorithmically and theoretically, lots of SVM algorithms involves solving convex problems, such as linear programming (Nash, Sofer 1996; Vanderbei 2001), convex quadratic programming (Nash, Sofer 1996), second order cone programming (Alizadeh, Goldfarb 2003; Boyd, Vandenberghe 2004; Goldfarb, Iyengar 2003), semi-definite programming (Klerk 2002) and etc. However, there are also non-convex and more general optimization problems appeared in SVMs: integer or discrete optimization considers non-convex problems with integer constraints, semi-infinite programming (Goberna, López 1998), bi-level optimization (Bennett *et al.* 2006) and so on. Especially in the process of model construction, these optimization problems may be solved many times. The research area of mathematical programming intersects with SVMs closely through these core optimization problems.

Generally speaking, there are three majors themes in the interplay of SVMs and mathematical programming. The first theme contains the development of under-lying models for standard classification or regression problems. Novel methods are developed by making some changes to the standard SVM models that enable the development of powerful new

algorithms, including ν-SVM (Schölkopf, Smola 2002; Vapnik 1998), linear programming SVM (Deng, Tian 2009; Deng *et al.* 2012, Weston *et al.* 1999), least squares SVM (LSSVM) (Johan *et al.* 2002), proximal SVM (PSVM) (Fung, Mangasarian 2001), twin SVM (TWSVM) (Khemchandani, Chandra 2007; Shao *et al.* 2011), multi-kernel SVM (Sonnenburg *et al.* 2006; Wu *et al.* 2007), AUC maximizing SVM (Ataman, Street 2005; Brefeld, Scheffer 2005), localized SVM (Segata, Blanzieri 2009), cost sensitive SVM (Akbani *et al.* 2004), fuzzy SVM (Lin, Wang 2002), Crammer-Singer SVM (Crammer, Singer 2001), K-support vector classification regression (K-SVCR) (Angulo, Català 2000) and etc., are developed. The second theme concerns the well-known optimization methods extended to new SVM models and paradigms. A wide range of programming methods is used to create novel optimization models in order to deal with different practical problems such as ordinal regression (Herbrich *et al.* 1999), robust classification (Goldfarb, Iyengar 2003; Yang 2007; Zhong, Fukushima 2007), semi-supervised and unsupervised classification (Xu, Schuurmans 2005; Zhao *et al.* 2006, 2007), transductive classification (Joachims 1999b), knowledge based classification (Fung *et al.* 2001, 2003; Mangasarian, Wild 2006), Universum classification (Vapnik 2006), priviledged classification (Vapnik, Vashist 2009), multi-instance classification (Mangasarian, Wild 2008), multi-label classification (Tsoumakas, Katakis 2007; Tsoumakas *et al.* 2010), multi-view classification (Farquhar *et al.* 2005), structured output classification (Tsochantaridis *et al.* 2005) and etc. The third theme considers the important issues in constructing and solving SVM optimization problems. On the one hand, several methods are developed for constructing optimization problems in order to enforce feature selection (Chen, Tian 2010; Tan *et al.* 2010), model selection (Bennett *et al.* 2006; Kunapuli *et al.* 2008), probabilistic outputs (Platt 2000), rule extraction from SVMs (Martens *et al.* 2008) and so on. On the other hand existing SVM optimization models are aimed at being solved more efficiently for the large scale data set, in which the key point is creating algorithms that exploit the structure of the optimization problem and pay careful attention to algorithmic and numeric issue, such as SMO (Platt 1999), efficient methods for solving large-scale linear SVM (Chang *et al.* 2008; Hsieh *et al.* 2008; Joachims 2006; Keerthi *et al.* 2008), parallel methods for solving large-scale SVM (Zanghirati, Zanni 2003) and etc.

Considering the many variants of SVM core optimization problems, a systematic survey is needed and helpful to understand and use this family of data mining techniques more easily. The goal of this paper is to closely review SVMs from the optimization point of view. Section 2 of the paper takes standard C – SVM as an example to summarize and explain the nature of SVMs. Section 3 will describe SVM optimization models with different variations according to the above three major themes. Several applications of SVMs to financial forecasting, bankruptcy prediction, credit risk analysis are introduced in Section 4. Finally, Section 5 will provide remarks and future research directions.

2. The nature of *C*-Support vector machines

In this section, standard C-SVM (Deng, Tian 2004, 2009; Deng *et al.* 2012; Vapnik 1998) for binary classification is briefly summarized and understood from several points of view.

Definition 2.1. (Binary classification). For the given training set

$$T = \{(x_1, y_1), ..., (x_l, y_l)\} \in (R^n \times \mathbf{y})^l, \tag{1}$$

where $x_i \in R^n$, $y_i \in \mathbf{y} = \{1, -1\}$, $i = 1, \cdots, l$, the goal is to find a real function $g(x)$ in R^n and derive the value of y for any x by the decision function

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

C – SVM formulates the problem as a convex quadratic programming

$$\min_{w,b,\xi} \quad \frac{1}{2}||w||^2 + C\sum_{i=1}^{l}\xi_i, \tag{3}$$

$$s.t. \quad y_i((w \cdot x_i) + b) \geq 1 - \xi_i, \ \ i = 1, \cdots, l, \tag{4}$$

$$\xi_i \geq 0, i = 1, \cdots, l, \tag{5}$$

where $\xi = (\xi_1, ..., \xi_l)^T$, and $C > 0$ is a penalty parameter. For this primal problem, C – SVM solves its Lagrangian dual problem

$$\min_{\alpha} \quad \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j K(x_i, x_j) - \sum_{j=1}^{l}\alpha_j, \tag{6}$$

$$\text{s.t.} \quad \sum_{i=1}^{l} y_i\alpha_i = 0, \tag{7}$$

$$0 \leq \alpha_i \leq C, \ \ i = 1, \cdots, l, \tag{8}$$

where $K(x, x')$ is the kernel function, which is also a convex quadratic problem and then construct the decision function.

As we all know, the principal of Structural Risk Minimization (SRM) is embodied in SVM, the confidential interval and the empirical risk should be considered at the same time. The two terms in the objective function (3) indicate that we not only minimize $||w||^2$ (maximize the margin), but also minimize $\sum_{i=1}^{l}\xi_i$, which is a measurement of violation of the constraints $y_i((w \cdot x_i) + b) \geq 1, i = 1, \cdots, l$. Here the parameter C determines the weighting between the two terms, the larger the value of C, the larger the punishment on empirical risk.

In fact, the parameter C has another meaningful interpretation (Deng, Tian 2009; Deng *et al.* 2012). Consider the binary classification problem, select a decision function candidate set $\mathcal{F}(t)$ depending on a real parameter t:

$$\mathcal{F}(t) = \{f(x) = \text{sgn}((w \cdot x) + b) \,|\, ||w|| \leq t, \quad t \in [0, \infty)\}, \tag{9}$$

and suppose that the loss function to be the soft margin loss function defined by

$$c(x, y, f(x)) = \max\{0, 1 - yg(x)\}, \quad \text{where} \ \ g(x) = (w \cdot x) + b. \tag{10}$$

Thus structural risk minimization is implemented by solving the following convex programming for an appropriate parameter t:

$$\min_{w,b,\xi} \ \sum_{i=1}^{l}\xi_i \ , \tag{11}$$

$$\text{s.t.} \quad y_i((w \cdot x_i) + b) \geq 1 - \xi_i, \ \ i = 1, \cdots, l, \tag{12}$$

$$\xi_i \geq 0, i = 1, \cdots, l, \tag{13}$$

$$|| w || \leq t. \tag{14}$$

An interesting point is proved that when the parameters C and t are chosen satisfying $t = \psi(C)$, where $\psi(\cdot)$ is nondecreasing in the interval $(0, +\infty)$, problem(3)~(5) and problem (11)~(14) will get the same decision function (Zhang *et al.* 2010). Hence the very interesting and important meaning of the parameter C is proposed: C corresponds to the size of the decision function candidate set in the principle of SRM: the larger the value of C, the larger the decision function candidate set.

Now we can summarize and understand C – SVM from following points of view: (i) Construct a decision function by selecting a proper size of the decision function candidate set via adjusting the parameter C; (ii) Construct a decision function by selecting the weighting between the margin of the decision function and the deviation of the decision function measured by the soft-margin loss function via adjusting the parameter C; (iii) Another understanding about C – SVM can also be seen in the literatures (Deng *et al.* 2012): Construct a decision function by selecting the weighting between flatness of the decision function and the deviation of the decision function measured by the soft-margin loss function via adjusting the parameter C.

3. Optimization models of support vector machines

In this section, several representative and important SVM optimization models with different variations are described and analyzed. These models can be divided into three categories: models for standard problems, models for nonstandard learning problems, and models combining SVMs with other issues in machine learning.

3.1. Models for standard problems

For the standard classification or regression problems, lot of methods are developed based on standard SVM models to be the powerful new algorithms. Here we briefly introduce several basic and efficient models, lots of developments of these models are omitted here.

3.1.1. Least squares support vector machine

Just like the standard C – SVM the starting point of least squares SVM (LSSVM) (Johan *et al.* 2002) is also to find a separating hyperplane, but with different primal problem. In fact, introducing the transformation $x = \Phi(x)$ and the corresponding kernel $K(x, x') = (\Phi(x) \cdot \Phi(x'))$, the primal problem becomes the convex quadratic programming

$$\min_{w, \eta, b} \ \frac{1}{2}||w||^2 + \frac{C}{2}\sum_{i=1}^{l}\eta_i^2, \tag{15}$$

$$\text{s.t.} \quad y_i((w \cdot \Phi(x_i)) + b) = 1 - \eta_i, \quad i = 1, \cdots, l. \tag{16}$$

The geometric interpretation of the above problem with $x \in R^2$ is shown in Figure 1, where minimizing $\frac{1}{2}\| w \|^2$ realizes the maximal margin between the straight lines

$$(w \cdot x) + b = 1 \text{ and } (w \cdot x) + b = -1, \tag{17}$$

while minimizing $\sum_{i=1}^{l} \eta_i^2$ implies making the straight lines (17) be proximal to all inputs of positive points and negative points respectively.

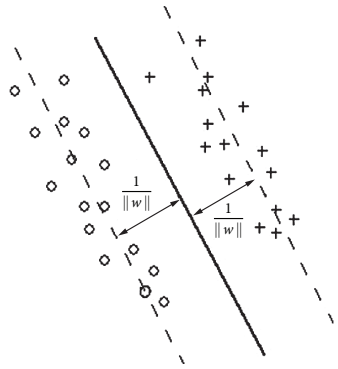

Fig. 1. Geometric interpretation of LSSVM

Its dual problem to be solved in LSSVM is also a convex quadratic programming

$$\max_{\alpha} \quad -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j (K(x_i, x_j) + \frac{\delta_{ij}}{C}) + \sum_{i=1}^{l}\alpha_i, \tag{18}$$

$$\text{s.t.} \quad \sum_{i=1}^{l}\alpha_i y_i = 0, \tag{19}$$

where

$$\delta_{ij} = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases} \tag{20}$$

In C – SVM, the error is measured by the soft margin loss function, this leads to the fact that the decision function is decided only by the support vectors. While in LSSVM, almost all training points contribute to the decision function, which makes it lose the sparseness. However, LSSVM needs to solve a quadratic programming with only equality constraints, or equivalently a linear system of equations. Therefore, it is simpler and faster than C – SVM.

3.1.2. Twin support vector machine

Twin support vector machine (TWSVM) is a binary classifier that perform classification using two nonparallel hyperplanes instead of a single hyperplane as in the case of conventional SVMs (Shao *et al.* 2011). Suppose the two non-parallel hyperplanes are the positive hyperplane

$$(w_+ \cdot x) + b_+ = 0, \tag{21}$$

and the negative hyperplane

$$(w_- \cdot x) + b_- = 0. \tag{22}$$

The primal problems for finding these two hyperplanes are two convex quadratic programming problems (Shao *et al.* 2011)

$$\min_{w_+, b_+, \xi_-} \quad \frac{1}{2} c_1 (\| w_+ \|^2 + b_+^2) + \frac{1}{2} \sum_{i=1}^{p} ((w_+ \cdot x_i) + b_+)^2 + c_2 \sum_{j=p+1}^{p+q} \xi_j, \tag{23}$$

$$\text{s.t.} \quad (w_+ \cdot x_j) + b_+ \le -1 + \xi_j, \; j = p+1, \cdots, p+q, \tag{24}$$

$$\xi_j \ge 0, \; j = p+1, \cdots, p+q \tag{25}$$

and

$$\min_{w_-, b_-, \xi_+} \quad \frac{1}{2} c_3 (\| w_- \|^2 + b_-^2) + \frac{1}{2} \sum_{i=p+1}^{p+q} ((w_- \cdot x_i) + b_-)^2 + c_4 \sum_{j=1}^{p} \xi_j, \tag{26}$$

$$\text{s.t.} \quad (w_- \cdot x_j) + b_- \ge 1 - \xi_j, \; j = 1, \cdots, p, \tag{27}$$

$$\xi_j \ge 0, \; j = 1, \cdots, p, \tag{28}$$

where $x_i, i = 1, ..., p$ are positive inputs, and $x_i, i = p+1, ..., p+q$ are negative inputs, $c_1 > 0, c_2 > 0, c_3 > 0, c_4 > 0$ are parameters, $\xi_- = (\xi_{p+1}, ..., \xi_{p+q})^{\mathrm{T}}$, $\xi_+ = (\xi_1, ..., \xi_p)^{\mathrm{T}}$.

For both of the above primal problems an interpretation can be offered in the same way. The geometric interpretation of the problem (23)~(25) with $x \in R^2$ is shown in Figure 2, where minimizing the second term $\sum_{i=1}^{p} ((w_+ \cdot x_i) + b_+)^2$ makes the positive hyperplane (blue solid line in Fig. 2) to be proximal to all positive inputs, minimizing the third term $\sum_{j=p+1}^{p+q} \xi_j$ with the constraints (24) and (25) requires the positive hyperplane to be at a distance from the negative inputs by pushing the negative inputs to the other side of the bounding hyperplane

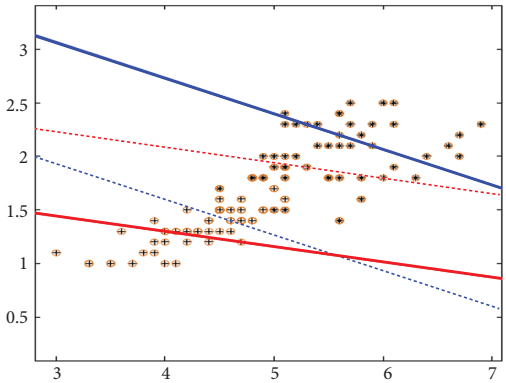

Fig. 2. Geometric interpretation of TWSVM

(blue dotted line in Fig. 2), where a set ξ of variables is used to measure the error whenever the positive hyperplane is close to the negative inputs. Minimizing the first term $\frac{1}{2}(\| w_+ \|^2 + b_+^2)$ realizes the maximal margin between the positive hyperplane $(w_+ \cdot x) + b_+ = 0$ and the bounding hyperplane $(w_+ \cdot x) + b_+ = -1$ in R^{n+1} space.

TWSVM is established based on solving two dual problems of the above primal problems separately. The generalization of TWSVM has been shown to be significantly better than standard SVM for both linear and nonlinear kernels. It has become one of the popular methods in machine learning because of its low computational complexity, since it solves above two smaller sized convex quadratic programming problems. On average, it is about four times faster than the standard SVMs.

3.1.3. AUC maximizing support vector machine

Nowadays the area under the receiver operating characteristics (ROC) curve, which corresponds to the Wilcoxon-Mann-Whitney test statistic, is increasingly used as a performance measure for classification systems, especially when one often has to deal with imbalanced class priors or misclassification costs. The area of that curve is the probability that a randomly drawn positive example has a higher decision function value than a random negative example; it is called the AUC (area under ROC curve). When the goal of a learning problem is to find a decision function with high AUC value, then it is natural to use a learning algorithm that directly maximizes this criterion. Over the last years, AUC maximizing SVMs (AUCSVM) have been developed (Ataman, Street 2005; Brefeld, Scheffer 2005), in which one kind of primary problem to be solved is a convex problem

$$\min_{w,\xi} \quad \frac{1}{2}\| w \|^2 + C\sum_{i=1}^{l^+}\sum_{i=1}^{l^-}\xi_{ij}, \tag{29}$$

$$\text{s.t.} \quad (w \cdot (x_i^+ - x_j^-)) \geq 1 - \xi_{ij}, i = 1, \cdots, l^+, \; j = 1, \cdots, l^-, \tag{30}$$

$$\xi_{ij} \geq 0, i = 1, \cdots, l^+, \; j = 1, \cdots, l^-, \tag{31}$$

where $x_i^+, i = 1, \cdots, l^+$, and $x_j^-, \; j = 1, \cdots, l^-$ are positive and negative inputs separately. It's dual problem is also a convex quadratic programming problem.

However, the existing algorithms all have the serious drawback that the number of constraints is quadratic in the number of training points, so they become very large even for small training set. To cope with this, different strategies can be constructed, in one of which a Fast and Exact k – Means (FEKM) (Goswami *et al.* 2004) algorithm is applied to approximate the problem by representing the l^+l^- many pairs $(x_i^+ - x_j^-)$ by only $l^+ - l^-$ cluster centers and thereby reduce the number of constraints and parameters. The approximate k – Means AUCSVM is more effective at maximizing the AUC than the SVM for linear kernels. Its execution time is quadratic in the sample size.

3.1.4. Fuzzy support vector machine

In standard SVMs, each sample is treated equally; i.e., each input point is fully assigned to one of the two classes. However, in many applications, some input points, such as the outliers, may not be exactly assigned to one of these two classes, and each point does not have the same meaning to the decision surface. To solve this problem, each data point in the training data set is assigned with a membership, if one data point is detected as an outlier, it is assigned with a low membership, so its contribution to total error term decreases. Unlike the equal treatment in standard SVMs, this kind of SVM fuzzifies the penalty term in order to reduce the sensitivity of less important data points. Fuzzy SVM (FSVM) construct its primal problem as (Lin, Wang 2002)

$$\min_{w,b,\xi} \ \tfrac{1}{2}|\ w|^2 + C\sum_{i=1}^{l} s_i\xi_i, \tag{32}$$

$$\text{s.t.} \quad y_i((w\cdot x_i)+b) \geq 1-\xi_i, \ \ i=1,...,l, \tag{33}$$

$$\xi_i \geq 0, \ \ i=1,\cdots,l, \tag{34}$$

where s_i is the membership generalized by some outlier-detecting methods. Its dual problem is similarly deduced as C – SVM to be a convex quadratic programming

$$\min_{\alpha} \ \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j K(x_i,x_j) - \sum_{j=1}^{l}\alpha_j, \tag{35}$$

$$\text{s.t.} \quad \sum_{i=1}^{l} y_i\alpha_i = 0, \tag{36}$$

$$0 \leq \alpha_i \leq Cs_i, i=1,\cdots,l. \tag{37}$$

Model (32)~(34) is also the general formulation of the cost sensitive SVM (Akbani *et al.* 2004) solving the imbalanced problem, in which different error costs are used for the positive (C_+) and negative (C_-) classes

$$\min_{w,b,\xi} \ \tfrac{1}{2}\|w\|^2 + C_+\sum_{y_i=1}\xi_i + C_-\sum_{y_{i=-1}} x_i, \tag{38}$$

$$\text{s.t.} \quad y_i((w\cdot x_i)+b) \geq 1-\xi_i, \ \ i=1,\cdots,l, \tag{39}$$

$$\xi_i \geq 0, \ \ i=1,\cdots,l. \tag{40}$$

3.2. Models for nonstandard problems

For the nonstandard problems appeared in different practical applications, a wide range of programming methods are used to build novel optimization models. Here we present several important and interesting models to show the interplay of SVMs and optimization.

3.2.1. Support vector ordinal regression

Support vector ordinal regression (SVOR) (Herbrich *et al.* 1999) is a method to solve a specialization of the multi-class classification problem: ordinal regression problem. The problem

of ordinal regression arises in many fields, e.g., information retrieval, econometric models, and classical statistics. It is complementary to the classification problem and metric regression problem due to its discrete and ordered outcome space.

Definition 3.1. (Ordinal regression problem). Given a training set

$$T = \{x_i^j\}_{i=1,\cdots,l^j}^{j=1,\cdots,M}, \tag{41}$$

where x_i^j is an input of a training point, the supscript $j = 1,\cdots,M$ denotes the corresponding class number, $i = 1,\cdots,l^j$ is the index within each class, and l^j is the number of the training points in class *j*. Find $M-1$ parallel hyperplanes in R^n

$$(w \cdot x) - b_r = 0, \quad r = 1,\cdots,M-1, \tag{42}$$

where $w \in R^n, b_1 \le b_2 \le \cdots \le b_{M-1}, b_0 = -\infty, b_M = +\infty$, such that the class number for any *x* can be predicted by

$$f(x) = \arg \min_{r\in\{1,\cdots,M\}} \{r : (w \cdot x) - b_r < 0\}. \tag{43}$$

SVOR constructs the primal problem as

$$\min_{w,b,\xi^{(*)}} \quad \frac{1}{2}\| w \|^2 + C\sum_{j=1}^{M}\sum_{i=1}^{l^j}(\xi_i^j + \xi_i^{*j}), \tag{44}$$

$$\text{s.t.} \quad (w \cdot x_i^j) - b_j \le -1 + \xi_i^j, j = 1,\cdots,M, i = 1,\cdots,l^i, \tag{45}$$

$$(w \cdot x_i^j) - b_{j-1} \ge 1 - \xi_i^{*j}, \quad j = 1,\cdots,M, i = 1,\cdots,l^i, \tag{46}$$

$$\xi_i^j \ge 0, \ \xi_i^{*j} \ge 0, \ \ j = 1,\cdots,M, i = 1,\cdots,l^j, \tag{47}$$

where $b = (b_1,\cdots,b_{M-1})^{\mathrm{T}}, b_0 = -\infty, b_M = +\infty$. Its dual problem is the following convex quadratic programming

$$\min_{\alpha^{(*)}} \ \frac{1}{2}\sum_{j,i}\sum_{j',i'}(\alpha_i^{*j} - \alpha_i^j)(\alpha_{i'}^{*j'} - \alpha_{i'}^{j'})(x_i^j \cdot x_{i'}^{j'}) - \sum_{j,i}(\alpha_i^j + \alpha_i^{*j}), \tag{48}$$

$$\text{s.t.} \ \sum_{i=1}^{l^j}\alpha_i^j = \sum_{i=1}^{l^{j+1}}\alpha_i^{*j+1}, \ \ j = 1,\cdots,M-1, \tag{49}$$

$$0 \le \alpha_i^j, \alpha_i^{*j} \le C, \ j = 1,\cdots,M, \ \ i = 1,\cdots,l^j, \tag{50}$$

$$\alpha_i^{*1} = 0, i = 1,\cdots,l^1, \tag{51}$$

$$\alpha_i^M = 0, \quad i = 1,\cdots,l^M. \tag{52}$$

Though SVOR is a method to solve a specialization of the multi-class classification problem and has many applications itself (Herbrich *et al.* 1999), it is also used in the context of solving general multi-class classification problem (Deng, Tian 2009; Deng *et al.* 2012; Yang 2007; Yang *et al.* 2005), in which the SVOR is used as a basic classifier and used several times instead of only once, just as the binary classifiers for multi-class classification. There are many choices

since any p – class SVOR with different order can be candidate, where $p = 2, 3, \ldots, M$. When $p = 2$, this approach reduces to the approach based on binary classifiers.

3.2.2. Semi-supervised support vector machine

In practice, labeled instances are often difficult, expensive, or time consuming to obtain, meanwhile unlabeled instance may be relatively easy to collect. Different with standard SVMs using only labeled training points, lots of semi-supervised SVMs (S^3VM) use large amount of unlabeled data, together with the labeled data, to build better classifiers. Transductive support vector machine (TSVM) (Joachims 1999b) is such an efficient method finding a labeling of the unlabeled data, so that a linear boundary has the maximum margin on both the original labeled data and the (now labeled) unlabeled data. The decision function has the smallest generalization error bound on unlabeled data.

For a training set given by

$$T = \{(x_1, y_1), \cdots, (x_l, y_l)\} \bigcup \{x_{l+1}, \cdots, x_{l+q}\}, \tag{53}$$

where $x_i \in R^n$, $y_i \in \{-1, 1\}$, $i = 1, \cdots, l$, $x_i \in R^n$, $i = l+1, \cdots, l+q$, and the set $\{x_{l+1}, \cdots, x_{l+q}\}$ is a collection of unlabeled inputs. The primal problem in TSVM is constructed as the following (partly) combinational optimization problem

$$\min_{w,b,\xi,y^*} \quad \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{l} \xi_i + C^* \sum_{i=1}^{l} \xi_i^*, \tag{54}$$

$$\text{s.t.} \quad y_i((w \cdot x_i) + b) \geq 1 - \xi_i, \ i = 1, \cdots, l, \tag{55}$$

$$y_i^* ((w \cdot x_i^*) + b) \geq 1 - \xi_i^*, \ i = l+1, \cdots, l+q, \tag{56}$$

$$\xi_i \geq 0, \ i = 1, \cdots, l, \tag{57}$$

$$\xi_i^* \geq 0, \ i = l+1, \cdots, l+q, \tag{58}$$

where $y^* = (y_{l+1}^*, \cdots, y_{l+q}^*)$, $C > 0, C^* > 0$ are parameters. However, finding the exact solution to this problem is NP-hard. Major effort has focused on efficient approximation algorithms. The SVM-light is the first widely used software (Joachims 1999b).

In the approximation algorithms, several relax the above TSVM training problem to semi-definite programming (SDP) (Xu, Schuurmans 2005; Zhao *et al.* 2006, 2007). The basic idea is to work with the binary label matrix of rank 1, and relax it by a positive semi-definite matrix without the rank constraint. However the computational cost of SDP is still expensive for large scale problems.

3.2.3. Universum support vector machine

Different with semi-supervised SVM leveraging unlabeled data from the same distribution, Universum support vector machine (USVM) use the the additional data not belonging to either class of interest. Universum contains data belonging to the same domain as the prob-

lem of interest and is expected to represent meaningful information related to the pattern recognition task at hand. Universum classification problem can be formulated as follows:

Definition 3.2. (Universum classification problem). Given a training set

$$T = \{(x_1, y_1), \cdots, (x_l, y_l)\} \bigcup \{x_1^*, \cdots, x_u^*\}, \tag{59}$$

where $x_i \in R^n$, $y_i \in \{-1,1\}$, $i = 1, \cdots, l$, $x_j^* \in R^n$, $j = 1, \cdots, u$, and the set

$$U = \{x_1^*, \cdots, x_u^*\} \tag{60}$$

is a collection of unlabeled inputs known not to belong to either class, find a real function $g(x)$ in R^n such that the value of y for any $x$can be predicted by the decision function

$$f(x) = \text{sgn}(g(x)). \tag{61}$$

Universum SVM constructs the following primal problem

$$\min_{w,b,\xi,\psi(*)} \quad \frac{1}{2}\| w \|_2^2 + C_t \sum_{i=1}^{l} \xi_i + C_u \sum_{s=1}^{u} (\psi_s + \psi_s^*), \tag{62}$$

$$\text{s.t.} \quad y_i((w \cdot x_i) + b) \geq 1 - \xi_i, \ \xi_i \geq 0, i = 1, \cdots, l, \tag{63}$$

$$-\varepsilon - \psi_s^* \leq (w \cdot x_s^*) + b \leq \varepsilon + \psi_s, s = 1, \cdots, u, \tag{64}$$

$$\psi_s, \psi_s^* \geq 0, s = 1, \cdots, u, \tag{65}$$

where $\psi^{(*)} = (\psi_1, \psi_1^*, \cdots, \psi_u, \psi_u^*)^T$ and $C_t > 0, C_u > 0, \varepsilon > 0$ are parameters. Its goal is to find a separating hyperplane $(w \cdot x) + b = 0$ such that, on the one hand, it separates the inputs $\{x_1, \cdots, x_l\}$ with maximal margin, and on the other hand, it approximates to the inputs $\{x_1^*, \cdots, x_u^*\}$ .We can also get its dual problem and introduce kernel function for dealing with nonlinear classification.

It is natural to consider the relationship between USVM and some 3-class classification. In fact, it can be shown that, under some assumptions, USVM is equivalent to K-SVCR (Angulo, Català 2000), and is also equivalent to the SVOR with $M = 3$ with slight modification (Gao 2008). USVM's performance depends on the quality of the Universum, methodology of choosing the appropriate Universum is the subject of future research.

3.2.4. Robust support vector machine

In standard SVMs, the parameters in the optimization problems are implicitly assumed to be known exactly. However, in practice, some uncertainty is often resent in many real-world problems, these parameters have perturbations since they are estimated from the training data which are usually corrupted by measurement noise. The solutions to the optimization problems are sensitive to parameter perturbations. So it is useful to explore formulations that can yield discriminants robust to such measurement errors. For example, when the inputs are subjected to measurement errors, it would be better to describe the inputs by uncertainty sets $\mathcal{X}_i \in R^n$, $i = 1, \cdots, l,$ since all we know is that the input belongs to the set $\mathcal{X}_i$. Therefore the standard problem turns to be the following robust classification problem.

Definition 3.3. (Robust classification problem). Given a training set

$$T = \{(\mathcal{X}_1, \mathcal{Y}_1), \cdots, (\mathcal{X}_l, \mathcal{Y}_l)\}, \tag{66}$$

where $\mathcal{X}_i$ is a set in R^n, $\mathcal{Y}_i \in \{-1,1\}$. Find a real function $g(x)$ in R^n, such that the value of y for any x can be predicted by the decision function

$$f(x) = \text{sgn}(g(x)). \tag{67}$$

The geometric interpretation of the robust problem with circle perturbations is shown in Figure 3, where the circles with "+" and "o" are positive and negative input sets respectively, the optimal separating hyperplane $(w^* \cdot x) + b^* = 0$ by the principle of maximal margin is constructed by robust SVM (RSVM). Now, the primal problem of RSVM for such case is a semi-infinite programming problem

$$\min_{w,b,\xi} \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{l} \xi_i, \tag{68}$$

$$\text{s.t.} \quad y_i((w \cdot (x_i \cdot r_i u_i)) + b) \geq 1 - \xi_i, \ \forall \| u_i \| \leq 1, i = 1, \cdots, l, \tag{69}$$

$$\xi_i \geq 0, i = 1, \cdots, l, \tag{70}$$

where the set $\mathcal{X}_i$ is a supersphere obtained from perturbation of a point x_i

$$\mathcal{X}_i = \{x \mid \| x - x_i \| \leq r_i\}. \tag{71}$$

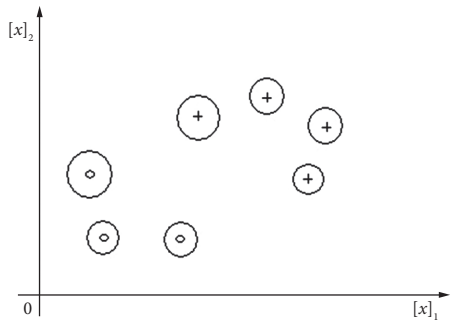

Fig. 3. Geometric interpretation of robust classification problem

This semi-infinite programming problem can be proved to be equivalent to the following second order cone programming (Goldfarb, Iyengar 2003; Yang 2007)

$$\min_{w,b,\xi,u,v,t} \quad \frac{1}{2}(u - v) + C \sum_{i=1}^{l} \xi_i, \tag{72}$$

$$\text{s.t.} \quad y_i((w \cdot x_i) + b) - r_i t \geq 1 - \xi_i, \ i = 1, \cdots, l, \tag{73}$$

$$\xi_i \geq 0, i = 1, \cdots, l, \tag{74}$$

$$u + v = 1, \tag{75}$$

$$\begin{pmatrix} u \\ t \\ v \end{pmatrix} \in L^3, \tag{76}$$

$$\begin{pmatrix} t \\ w \end{pmatrix} \in L^{n+1}, \tag{77}$$

its dual problem is also a second order cone programming

$$\max_{\alpha,\beta,\gamma,z_u,z_v} \quad \beta + \sum_{i=1}^{l} \alpha_i, \tag{78}$$

$$\text{s.t.} \quad \gamma \le \sum_{i=1}^{l} r_i \alpha_i - \sqrt{\sum_{i=1}^{l}\sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j K(x_i, x_j)}, \tag{79}$$

$$\beta + z_u = \frac{1}{2}, \tag{80}$$

$$\beta + z_v = -\frac{1}{2}, \tag{81}$$

$$\sum_{i=1}^{l} y_i \alpha_i = 0, \tag{82}$$

$$0 \le \alpha_i \le C, i = 1, \cdots, l, \tag{83}$$

$$\sqrt{\gamma^2 + z_v^2} \le z_u, \tag{84}$$

which can be efficiently solved by Self-Dual-Minimization (SeDuMi). SeDuMi is a tool for solving optimization problems. It can be used to solve linear programming, second-order cone programming and semi-definite programming, and is available at the web site http://sedumi.mcmaster.ca.

3.2.5. Knowledge based support vector machine

In many real-world problems, we are given not only the traditional training set, but also prior knowledge such as some advised classification rules. If appropriately used, prior knowledge can significantly improve the predictive accuracy of learning algorithms or reduce the amount of training data needed. Now the problem can be extended in the following way: the single input points in the training points are extended to input sets, called knowledge sets. If we consider the input sets restricted as polyhedrons, the problem is formulated mathematically as follows:

Definition 3.4. (Knowledge-based classification problem). Given a training set

$$T = \{(\mathcal{X}_1, y_1), \cdots, (\mathcal{X}_p, y_p), (\mathcal{X}_{p+1}, y_{p+1}), \cdots, (\mathcal{X}_{p+q}, y_{p+q})\}, \tag{85}$$

where $\mathcal{X}_i$ is a polyhedron in R_n defined by $\mathcal{X}_i = \{x \mid Q_i x \le d_i\}$, and $Q_i \in R^{l_i \times n}$, $d_i \in R^{l_i}$, $y_1 = \cdots = y_p = 1$, $y_{p+1} = \cdots = y_{p+q} = -1$. Find a real valued function $g(x)$ in R_n, such that the value of y for any x can be predicted by the decision function

$$f(x) = \operatorname{sgn}(g(x)). \tag{86}$$

Of course we can construct the primal problem to be the following semi-infinite programming problem

$$\min_{w,b,\xi} \quad \frac{1}{2}\| w \|^2 + C\sum_{i=1}^{p+q}\xi_i, \tag{87}$$

$$\text{s.t.} \quad (w\cdot x)+b \geq 1, \text{ for } x \in \mathcal{X}_i . i = 1,\cdots, p, \tag{88}$$

$$(w\cdot x)+b \leq -1, \text{ for } x \in \mathcal{X}_i . i = p+1,\cdots, p+q, \tag{89}$$

$$\xi_i \geq 0, i = 1,\cdots, p+q. \tag{90}$$

However, it was shown that the constraints (88)~(90) can be converted into a set of limited constraints and then the problem becomes a quadratic programming (Fung *et al.* 2001)

$$\min_{w,b,\xi,u,v,t} \quad \frac{1}{2}\| w \|^2 + C\sum_{i=1}^{p+q}((\sum_{j=1}^{n}\xi_{i_j})+\eta_i), \tag{91}$$

$$\text{s.t.} \quad -\xi_i \leq Q_i^{\mathrm{T}}u_i + w \leq \xi_i, i = 1,\cdots, p, \tag{92}$$

$$d_i^{\mathrm{T}}u_i - b + 1 \leq \eta_i, i = 1,\cdots, p, \tag{93}$$

$$-\xi_i \leq Q_i^{\mathrm{T}}u_i - w \leq \xi_i, \ i = p+1,\cdots, p+q, \tag{94}$$

$$d_i^{\mathrm{T}}u_i + b + 1 \leq \eta_i, \ i = p+1,\cdots, p+q, \tag{95}$$

$$\xi, \eta, u \geq 0. \tag{96}$$

This model considered the linear knowledge incorporated to linear SVM, while linear knowledge based nonlinear SVM and nonlinear knowledge based SVM were also proposed by Mangasarian and his co-workers (Fung *et al.* 2003; Mangasarian, Wild 2006). Handling prior knowledge is worthy of further study, especially when the training data may not be easily available whereas expert knowledge may be readily available in the form of knowledge sets. Another prior information such as some additional descriptions of the training points was also considered and a method called privileged SVM was proposed (Vapnik, Vashist 2009), which allows one to introduce human elements of teaching: teacher's remarks, explanations, analogy, and so on in the machine learning process.

3.2.6. Multi-instance support vector machine

Multi-instance problem was proposed in the application domain of drug activity prediction, and similar to both the robust and knowledge-based classification problems, it can be formulated as follows.

Definition 3.5. (Multi-instance classification problem). Suppose that there is a training set

$$T = \{(\mathcal{X}_1, \mathcal{Y}_1),\cdots,(\mathcal{X}_l, \mathcal{Y}_l)\}, \tag{97}$$

where $\mathcal{X}_i = \{x_{i1}, \cdots, x_{il_i}\}$, $x_{ij} \in R^n$, $j = 1, \cdots, l_i$, $\mathcal{Y}_l \in \{-1, 1\}$. Find a real function $g(x)$ in R^n, such that the label y for any instance x can be predicted by the decision function

$$f(x) = \text{sgn}(g(x)). \tag{98}$$

The set $\mathcal{X}_i$ is called a bag containing a number of instances. Note that the interesting point of this problem is that: the label of a bag is related with the labels of the instances in the bag and decided by the following way: a bag is positive if and only if there is at least one instance in the bag is positive; a bag is negative if and only if all instances in the bag are negative. A geometric interpretation of multi-instance classification problem is shown in Figure 4, where every enclosure stands for a bag; a bag with "+" is positive and a bag with "o" is negative, and both "+" and "o" stand for instances.

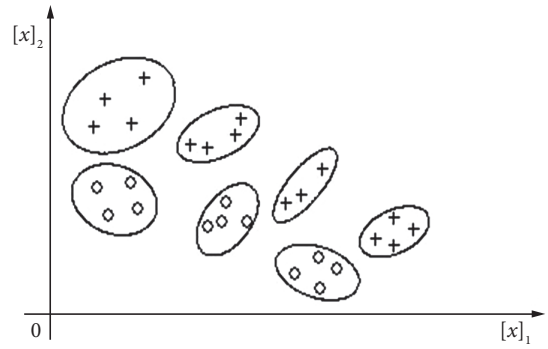

Fig. 4. Geometric interpretation of multi-instance classification problem

For a linear classifier, a positive bag is classified correctly if and only if some convex combination of points in the bag lies on the positive side of a separating plane. Thus the primal problem in the multi-instance SVM (MISVM) is constructed as the following nonlinear programming problem (Mangasarian, Wild 2008)

$$\min_{w,b,v,\xi} \quad \frac{1}{2}\| w \|^2 + C_1 \sum_{i=1}^{p} \xi_i + C_2 \sum_{i=r+1}^{r+s} \xi_i, \tag{99}$$

$$\text{s.t.} \quad (w \cdot \sum_{j \in I(i)} v_j^i x_j) + b \geq 1 - \xi_i, i = 1, \cdots, p, \tag{100}$$

$$(w \cdot x_i) + b \leq -1 + \xi_i, \quad i = r+1, \cdots, r+s, \tag{101}$$

$$\xi_i \geq 0, i = 1, \cdots, p, r+1, \cdots, r+s, \tag{102}$$

$$v_j^i \geq 0, j \in I(i), i = 1, \cdots, p, \tag{103}$$

$$\sum_{j \in I(i)} v_j^i = 1, i = 1, \cdots, p, \tag{104}$$

where *r* and *s* are respectively the number of the instances in all positive bags and all negative bags, and p is the number of positive bags.

Though the above problem is nonlinear, it is easy to see that among its constraints, only the first one is nonlinear, and in fact is bilinear. Then a local solution to this problem is obtained by solving a succession of fast linear programs in a few iterations: Alternatively, hold one set of variables which constitute the bilinear terms constant while varying the other set. For a nonlinear classifier, a similar statement applies to the higher dimensional space induced by the kernel.

3.3. Other SVM issues

This section concerns some important issues of SVMs: feature selection, parameter (model) selection, probabilistic outputs, rule extraction, implements of algorithms and so on, in which the optimization models are also applied.

3.3.1. Feature selection via SVMs

Standard SVMs cannot get the importance features, while identifying a subset of features which contribute most to classification is also an important task in machine learning. The benefit of feature selection is twofold. It leads to parsimonious models that are often preferred in many scientific problems, and it is also crucial for achieving good classification accuracy in the presence of redundant features. We can combine SVM with various feature selection strategies, Some of them are "filters": general feature selection methods independent of SVMs. That is, these methods select important features first and then SVMs are applied. On the other hand, some are wrapper-type methods: modifications of SVMs which choose important features as well as conduct training/testing. In the machine learning literature, there are several proposals for feature selection to accomplish the goal of automatic feature selection in the SVM (Bradley, Mangasarian 1998; Guyon *et al.* 2001; Li *et al.* 2007; Weston *et al.* 2001; Zhu *et al.* 2004; Zou, Yuan 2008) via some optimization problems, in some of which they applied the l_0-norm, l_1-norm or l_∞-norm SVM and got competitive performance.

Naturally, we expect that using the l_p-norm $(0 < p < 1)$ in SVM can find more sparse solution than using l_1-norm and more algorithmic advantages. Through combining C-SVM and feature selection strategy by introducing the l_p-norm $(0 < p < 1)$, the primal problem in l_p-support vector machines (l_p-SVM) is (Chen, Tian 2010; Deng *et al.* 2012; Tian *et al.* 2010)

$$\min_{w,b,\xi} \quad \| w \|_p^p + C\sum_{i=1}^{l} \xi_i, \tag{105}$$

$$\text{s.t.} \quad y_i((w \cdot x_i) + b) \geq 1 - \xi_i, \ i = 1, \cdots, l, \tag{106}$$

$$\xi_i \geq 0, \ i = 1, \cdots, l, \tag{107}$$

where *p* is a nonnegative parameter, and

$$| w |_p^p = \sum_{i=1}^{n} | w_i |^p. \tag{108}$$

For the case of $p = 0, \| w \|_0$ represents the number of nonzero components of *w*, for the case of $p = 0$, the problem turns to be a linear programming, for the case of $p = 2$, a convex

quadratic programming, and for the case of $p = \infty$, the problem is proved to be equivalent to a linear programming problem (Zou, Yuan 2008).

However, solving this nonconvex, non-Lipschitz continuous minimization problem is very difficult. After equivalently transforming the problem to be

$$\min_{w,b,\xi,v} \sum_{i=1}^{n} v_i^p + C\sum_{i=1}^{l} \xi_i, \tag{109}$$

$$\text{s.t.} \quad y_i((w \cdot x_i) + b) \geq 1 - \xi_i, i = 1, \cdots, l, \tag{110}$$

$$\xi_i \geq 0, \quad i = 1, \cdots, l, \tag{111}$$

$$-v \leq w \leq v, \tag{112}$$

and introducing the first-order Taylor's expansion as the approximation of this nonlinear objective function, this problem can be solved by a successive linear approximation algorithm (Bradley *et al.* 1998; Deng *et al.* 2012). Furthermore, a lower bound for the absolute value of nonzero entries in every local optimal solution of l_p-SVM is developed (Tian *et al.* 2010), which reflects the relationship between sparsity of the solution and the choice of the parameters C and p.

3.3.2. LOO error bounds for SVMs

The success of SVMs depends on the tuning of their several parameters which affect the generalization error. An effective approach choosing these parameters which will generalize well is to estimate the generalization error and then search for parameters so that this estimator is minimized. This requires that the estimators are both effective and computationally efficient. Leave-one-out (LOO) method (Vapnik, Chapelle 2000) is the extreme case of cross-validation, and LOO error provides an almost unbiased estimate of the generalization error. However, one shortcoming of the LOO method is that it is highly time consuming when the number of training points l is very large thus methods are sought to speed up the process. An effective approach is to approximate the LOO error by its upper bound, that is computed by running a concrete classification algorithm only once on the original training set T of size l. This approach has successfully been developed for both support vector classification machine (Gretton *et al.* 2001; Jaakkola, Haussler 1998, 1999; Joachims 2000; Vapnik, Chapelle 2000), support vector regression machine (Chang, Lin 2005; Tian 2005; Tian, Deng 2005), and support vector ordinal regression (Yang *et al.* 2009). Then we can search for parameter so that this upper bound is minimized.

Furthermore, inspired by the LOO error bound, approaches were proposed by directly minimizing the expression given by the bound in an attempt to minimize leave-one-out error (Tian 2005; Weston 1999), and these approaches are called LOO support vector machines (LOOSVM). LOOSVMs also involve solving convex optimization problems, and one of which in such the algorithms is a linear programming problem

$$\min_{\alpha,\xi} \sum_{i=1}^{l} \xi_i, \tag{113}$$

$$\text{s.t.} \quad y_i f(x_i) \geq 1 - \xi_i + \alpha_i K(x_i, x_j), \; i = 1, \cdots, l, \tag{114}$$

$$\alpha_i \geq 0, \xi_i \geq 0, \ i = 1, \cdots, l, \tag{115}$$

where

$$f(x) = \text{sgn}(\sum_{i=1}^{l} \alpha_i y_i K(x_i, x_j)), \tag{116}$$

and $K(x, x')$ is the kernel function. LOOSVMs possess many of the same properties as SVMs. The main novelty of these algorithms is that apart from the choice of kernel, they are parameter less: the selection of the number of training errors is inherent in the algorithms and not chosen by an extra free parameter as in SVMs.

3.3.3. Probabilistic outputs for support vector machines

For a binary classification problem with the training set (1), standard *C*-SVM computes a decision function (2) such that it can be used to predict the label of any test input *x*. However, we cannot guarantee that the deduction is absolutely correct. So sometimes we hope to know how much confidence we have, i.e. the probability of the input x belonging to the positive class. To answer this question, investigate the information contained in $g(x)$. It is not difficult to imagine that the larger $g(x)$ is, the larger the probability is. So the value of $g(x)$ can be used to estimate the probability $P(y = 1 \mid g(x))$ of the input x belonging to the positive class. In fact, we only need to establish an appropriate monotonic function from $(-\infty, +\infty)$ where $g(x)$ takes value to the probability values interval [0,1], such as the sigmoid function is used (Platt 2000)

$$p(g) = \frac{1}{1 + \exp(c_1 g + c_2)}, \tag{117}$$

where $c_1 < 0$ and c_2 are two parameters to be found. In order to choose the optimal values c_1^* and c_2^*, an unconstrained optimization problem is constructed following the idea of maximum likelihood estimation

$$\max \prod_{y_i=1} p_i \prod_{y_i=-1} (1\text{-}p_i), \tag{118}$$

where

$$p_i = p_i(c_1, c_2) = \frac{1}{1 + \exp(c_1 g(x_i) + c_2)}, i = 1, \cdots, l. \tag{119}$$

This problem is a two-parameter maximization, hence it can be performed using any number of optimization algorithms, while Figure 5 shows a numerical results of the probabilistic outputs for a linear SVM on some data (Platt 2000).

For better implementation of solving problem (118), an improved algorithm that theoretically converges and avoids numerical difficulties was also proposed (Lin *et al.* 2007).

3.3.4. Rule extraction from support vector machines

Though SVMs are the state-of-the-art tools in data mining, their strength are also their main weakness, as the generated nonlinear models are typically regarded as incomprehensible black-box models. Therefore, opening the black-box or making SVMs explainable, i.e.

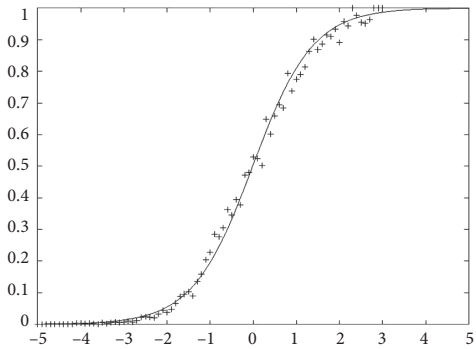

Fig. 5. Probabilistic outputs for a linear SVM on some data

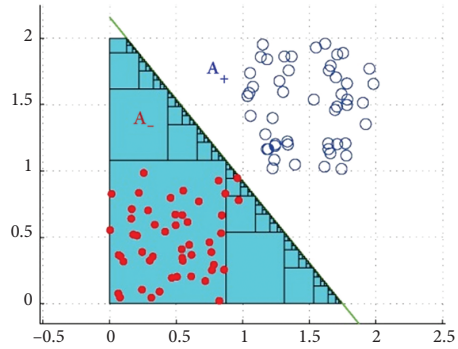

Fig. 6. Geometric interpretation of rule extraction from linear SVM

extracting rules from SVMs models to mimic their behavior and give comprehensibility to them became more important and necessary in areas such as medical diagnosis and credit evaluation (Martens *et al.* 2008).

There are several techniques to extract rules from SVMs so far, and one potential method of classifying these rule extraction techniques is in terms of the "translucency", which is of the view taken within the rule extraction method of the underlying classifier. Two main categories of rule extraction methods are known as pedagogical (Setiono *et al.* 2006) and decompositional (Fung *et al.* 2005; Núñez *et al.* 2002). Pedagogical algorithms consider the trained model as a black box and directly extract rules which relate the inputs and outputs of the SVMs. On the other hand, decompositional approach is closely related to the internal workings of the SVMs and their constructed hyperplane.

Fung *et al.* (2005) present an algorithm to extract propositional classification rules from linear SVMs. The method is considered to be decompositional because it is only applicable when the underlying model provides a linear decision boundary. The resulting rules are parallel with the axes and nonoverlapping, but only (asymptotically) exhaustive. The algorithm is iterative and extracts the rules by solving a constrained optimization problem that is computationally inexpensive to solve. Figure 6 shows execution of the algorithm for binary classification and only rules for the black squares are being extracted (Fung *et al.* 2005). Different optimal rules will be extracted according to different criteria, and maximizes the log of the volume of the region that the rules encloses is one kind of which, leads to solving the following optimization problem

$$\max_{x \in R^n} \log\left(\prod_{y_i=1} x_i\right), \tag{120}$$

$$\text{s.t.} \quad (w \cdot x) + b = 0, \tag{121}$$

$$0 \le x \le 1. \tag{122}$$

However, existing rule extracted algorithms have limitations in real applications especially when the problems are large scale with high dimensions. So the incorporation of the feature selection into the rule extraction problem is also a possibility to be explored, and there are already some papers considering this topic (Yang, Tian 2011).

4. Applications in economics

SVMs have been successfully applied in many fields including economics, finance and management. Some applications of SVMs to financial forecasting problems have been reported (Cao, Tay 2001, 2003; Kim 2003; Tay, Cao 2001, 2002). Tay and Cao (2002) proposed *C*-ascending SVMs by increasing the value of parameter C, this idea was based on the assumption that it was better to give more weights on recent data than distant data. Their results showed that *C*-ascending SVMs gave better performance than standard SVM in financial time series forecasting. Cao and Tay (2003) also compared SVMs with multilayer backpropagation (BP) neural network and the regularized radial basis function (RBF) neural network. Simulation results showed that SVMs with adaptive parameters outperform two other methods.

Bankruptcy prediction is an important and widely studied topic since it can have significant impact on bank lending decisions and profitability, SVMs were successfully adopted to this problem in recent years (Fan, Palaniswami 2000; Huang *et al.* 2004; Min, Lee 2005; Min *et al.* 2006; Shin *et al.* 2005). The results for different real world data sets demonstrated that SVMs outperform BP at the accuracy and generalization performance. The effect of the variability in performance with respect to various values of parameters in SVMs were also investigated.

Due to recent financial crises and regulatory concerns, credit risk assessment is an area that has seen a resurgence of interest from both the academic world and the business community. Since credit risk analysis or credit scoring is in fact a classification problem, so lots of classification techniques were applied to this field, and naturally competitive SVMs can be used (Stoenescu Cimpoeru 2011; Shi *et al.* 2005; Thomas *et al.* 2005; Van Gestel *et al.* 2003; Yu *et al.* 2009; Zhou *et al.* 2009). Additionally, combining genetic algorithms with SVMs, named hybrid GA-SVMs can simultaneously perform feature selection task and model parameters optimization (Huang *et al.* 2007). Because in credit scoring areas we usually cannot label one customer as absolutely good or bad, a fuzzy support vector machine different with model (32)~(34) was proposed to treat every inputs as both positive and negative classes, but with different memberships (Wang *et al.* 2005),

$$\min_{w,b,\xi} \quad \frac{1}{2}\| w \|^2 + C\sum_{i=1}^{l}[m_i\xi_i + (1-m_i)\eta_i], \tag{123}$$

$$\text{s.t.} \quad (w \cdot x_i) + b \geq 1 - \xi_i, i = 1, \cdots, l, \tag{124}$$

$$(w \cdot x_i) + b \leq -1 + \eta_i, i = 1, \cdots, l, \tag{125}$$

$$\xi_i \geq 0, \eta_i \geq 0, \ i = 1, \cdots, l, \tag{126}$$

where m_i is the membership for the ith inputs to the class y_i.

Other applications in economics, including motor insurance fraud management (Furlan *et al.* 2011), environmental risk assessment (Kochanek, Tynan 2010), e-banking website quality assessment (Kaya, Kahraman 2011) and etc., can also be explored by SVMs.

5. Remarks and future directions

This paper has offered an extensive review of optimization models of SVMs, including least squares SVM, twin SVM, AUC Maximizing SVM, and fuzzy SVM for standard problems; support vector ordinal machine, semi-supervised SVM, Universum SVM, robust SVM, knowledge based SVM, and multi-instance SVM for nonstandard problems, as well as l_p-norm SVM for feature selection, LOOSVM based on minimizing LOO error bound, probabilistic outputs for SVM, and rule extraction from SVM. These models have already been used in many real-life applications, such as text categorization, bio-informatics, bankruptcy prediction, remote sensing image analysis, network intrusion and detection, information security, and credit assessment management. Some applications to financial forecasting, bankruptcy prediction, credit risk analysis are also reviewed in this paper. Researchers and engineers in data mining, especially in SVMs can benefit from this survey in better understanding the essence of the relation between SVMs and optimization. In addition, it can also serve as a reference repertory of such approaches.

Research in SVMs and research in optimization have become increasingly coupled. In this paper, we can see optimization models including linear, nonlinear, second order cone, and semi-definite, integer or discrete, semi-infinite programming models are used. Of course, there are still many optimization models of SVMs not discussed here, and new practical problems remaining to be explored present new challenges to SVM to construct new optimization models. These models should also have the same desirable properties as the models in this paper including (Bennett *et al.* 2006): good generalization, scalability, simple and easy implementation of algorithm, robustness, as well as theoretically known convergence and complexity.

Acknowledgments

This work has been partially supported by grants from National Natural Science Foundation of China (No. 70921061, No. 10601064), the CAS/SAFEA International Partnership Program for Creative Research Teams, Major International (Regional) Joint Research Project (No. 71110107026), the President Fund of GUCAS, and the National Technology Support Program 2009BAH42B02.

References

Adankon, M. M.; Cheriet, M. 2009. Model selection for the LS-SVM application to handwriting recognition, *Pattern Recognition* 42(12): 3264–3270. http://dx.doi.org/10.1016/j.patcog.2008.10.023

Akbani, R.; Kwek1, S.; Japkowicz, N. 2004. Applying support vector machines to imbalanced datasets, in *Proceedings of European Conference on Machine Learning, Lecture Notes in Computer Science* 3201: 39–50.

Alizadeh, F.; Goldfarb, D. 2003. Second-order cone programming, *Mathematical Programming, Series B* 95: 3–51. http://dx.doi.org/10.1007/s10107-002-0339-5

Ancona, N.; Cicirelli, G.; Branca, A.; Distante, A. 2001. Goal detection in football by using support vector machines for classification, in *Proceedings of International Joint Conference on Neural Networks* 1: 611–616.

Angulo, C.; Català, A. 2000. K-SVCR, a multi-class support vector machine, in *Proceedings of European Conference on Machine Learning, Lecture Notes in Computer Science* 1810: 31–38. http://dx.doi.org/10.1007/3-540-45164-1_4

Ataman, K.; Street, W. N. 2005. Optimizing area under the ROC curve using ranking SVMs, in *Proceedings of International Conference on Knowledge Discovery in Data Mining.* Available from Internet: http://dollar.biz.uiowa.edu/ street/research/kdd05kaan.pdf

Azimi-Sadjadi, M. R.; Zekavat, S. A. 2000. Cloud classification using support vector machines, in *Proceedings of IEEE Geoscience and Remote Sensing Symposium* 2: 669–671.

Bennett, K.; Ji, X.; Hu, J.; Kunapuli, G.; Pang, J. S. 2006. Model selection via bilevel optimization, in *Proceedings of IEEE World Congress on Computational Intelligence*, 1922–1929.

Bennett, K.; Parrado-Hernández, E. 2006. The interplay of optimization and machine learning research, *Journal of Machine Learning Research* 7: 1265–1281.

Borgwardt, K. M. 2011. *Kernel Methods in Bioinformatics*. Handbook of Statistical Bioinformatics. Part 3, 317–334.

Boyd, S.; Vandenberghe, L. 2004. *Convex Optimization*. Cambridge University Press.

Bradley, P. S.; Mangasarian, O. L.; Street, W. N. 1998. Feature selection via mathematical programming, *INFORMS Journal on Computing Spring* 10(2): 209–217.

Bradley, P.; Mangasarian, O. 1998. Feature selection via concave minimization and support vector machines, in *Proceedings of International Conference on Machine Learning*, Morgan Kaufmann, 82–90.

Brefeld, U.; Scheffer, T. 2005. Auc maximizing support vector learning, in *Proceedings of the 22nd International Conference on Machine Learning, Workshop on ROC Analysis in Machine Learning*. Available from Internet: http://users.dsic.upv.es/~flip/ROCML2005/papers/brefeldCRC.pdf

Cao, L. J.; Tay, F. 2001. Financial forecasting using support vector machines, *Neural Computing Applications* 10: 184–192. http://dx.doi.org/10.1007/s005210170010

Cao, L. J.; Tay, F. 2003. Support vector machine with adaptive parameters in financial time series forecasting, *IEEE Transactions on Neural Networks* 14(6): 1506–1518. http://dx.doi.org/10.1109/TNN.2003.820556

Chang, K. W.; Hsieh, C. J.; Lin, C. J. 2008. Coordinate descent method for large-scale L2-loss linear SVM, *Journal of Machine Learning Research* 9: 1369–1398.

Chang, M. W.; Lin, C. J. 2005. Leave-one-out bounds for support vector regression model selection, *Neural Computation* 17(5): 1188–1222. http://dx.doi.org/10.1162/0899766053491869

Chen, W. J.; Tian, Y. J. 2010. l_p-norm proximal support vector machine and its applications, *Procedia Computer Science* 1(1): 2417–2423. http://dx.doi.org/10.1016/j.procs.2010.04.272

Stoenescu Cimpoeru, S. 2011. Neural networks and their application in credit risk assessment. Evidence from the Romanian market, *Technological and Economic Development of Economy* 17(3): 519–534. http://dx.doi.org/10.3846/20294913.2011.606339

Cortes, C.; Vapnik, V. 1995. Support vector networks, in *Proceedings of Machine Learning* 20: 273–297.

Crammer, K.; Singer, Y. 2001. On the algorithmic implementation of multi-class kernel based vector machines, *Journal of Machine Learning Research* 2: 265–292.

Cristianini, N.; Shawe-Taylor, J. 2000. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press.

Deng, N. Y.; Tian, Y. J. 2004. *New Method in Data Mining: Support Vector Machines*. Science Press, Beijing, China.

Deng, N. Y.; Tian, Y. J. 2009. *Support Vector Machines: Theory, Algorithms and Extensions*. Science Press, Beijing, China.

Deng, N. Y.; Tian, Y. J.; Zhang, C. H. 2012. *Support Vector Machines: Optimization Based Theory, Algorithms and Extensions.* CRC Press (in press).

Druker, H.; Shahrary, B.; Gibbon, D. C. 2001. Support vector machines: relevance feedback and information retrieval, *Information Processing and Management* 38(3): 305–323.

Fan, A.; Palaniswami, M. 2000. Selecting bankruptcy predictors using a support vector machine approach, in *Proceedings of International Joint Conference on Neural Net-works (IJCNN'00)* 6: 354–359.

Farquhar, J. D. R.; Hardoon, D. R.; Meng, H. Y.; Taylor, J. S.; Szedmák, S. 2005. Two view learning: SVM-2K, theory and practice, *Advances in Neural Information Processing Systems* 18: 355–362.

Fung, G.; Mangasarian, O. L. 2001. Proximal support vector machine classifiers, in *Proceedings of International Conference of Knowledge Discovery and Data Mining*, 77–86.

Fung, G.; Mangasarian, O. L.; Shavlik, J. 2001. Knowledge-based support vector machine classifiers, *Advances in Neural Information Processing Systems* 15: 537–544.

Fung, G.; Mangasarian, O. L.; Shavlik, J. 2003. Knowledge-based nonlinear kernel classifiers, *Learning Theory and Kernel Machines, Lecture Notes in Computer Science* 2777: 102–113. http://dx.doi.org/10.1007/978-3-540-45167-9_9

Fung, G.; Sandilya, S.; Rao, R. B. 2005. Rule extraction from linear support vector machines, in *Proceedings of International Conference on Knowledge Discovery in Data Mining*, 32–40.

Furlan, S.; Vasilecas, O.; Bajec, M. 2011. Method for selection of motor insurance fraud management system components based on business performance, *Technological and Economic Development of Economy* 17(3): 535–561. http://dx.doi.org/10.3846/20294913.2011.602440

Ganapathiraju, A.; Hamaker, J.; Picone, J. 2004. Applications of support vector machines to speech recognition, *IEEE Transaction on Signal Process* 52(8): 2348–2355. http://dx.doi.org/10.1109/TSP.2004.831018

Gao, T. T. 2008. *U-support Vector Machine and Its Applications:* Master Thesis. China Agricultural University.

Goberna, M. A.; López, M. A. 1998. *Linear Semi-Infinite Optimization*. New York: John Wiley.

Goldfarb, D.; Iyengar, G. 2003. Robust convex quadratically constrained programs, *Mathematical Programming, Series B* 97: 495–515. http://dx.doi.org/10.1007/s10107-003-0425-3

Goswami, A.; Jin, R.; Agrawal, G. 2004. Fast and exact out-of-core k-means clustering, in *Proceedings of the IEEE International Conference on Data Mining* 10: 17–40.

Gretton, A.; Herbrich, R.; Chapelle, O. 2001. *Estimating the leave-one-out error for classification learning with SVMs*. Available from Internet: http://www.kyb.tuebingen.mpg.de/publications/pss/ps1854.ps

Gutta, S.; Huang, J. R. J.; Jonathon, P.; Wechsler, H. 2000. Mixture of experts for classification of gender, ethnic origin, and pose of human, *IEEE Transaction on Neural Networks* 11(4): 948–960. http://dx.doi.org/10.1109/72.857774

Guyon, I.; Weston, J.; Barnhill, S.; Vapnik, V. 2001. Gene selection for cancer classification using support vector machines, *Machine Learning* 46: 389–422. http://dx.doi.org/10.1023/A:1012487302797

Herbrich, R. 2002. *Learning Kernel Classifiers: Theory and Algorithms*. The MIT Press.

Herbrich, R.; Graepel, T.; Obermayer, K. 1999. Support vector learning for ordinal regression, in *Proceedings of the 9th International Conference on Artifical Neural Networks*, 97–102. http://dx.doi.org/10.1049/cp:19991091

Hsieh, C. J.; Chang, K. W.; Lin, C. J.; Keerthi, S. S.; Sundararajan, S. 2008. A dual coordinate descent method for large-scale linear SVM, in *Proceedings of the 25th International Conference on Machine Learning (ICML'08)*, 408–415.

Huang, C. L.; Chen, M. C.; Wang, C. J. 2007. Credit scoring with a data mining approach based on support vector machines, *Expert Systems with Applications* 33(4): 847–856. http://dx.doi.org/10.1016/j.eswa.2006.07.007

Huang, W.; Lai, K. K.; Nakamori, Y.; Wang, S. Y. 2004. Forecasting foreign exchange rates with artificial neural networks: a review, *International Journal of Information Technology and Decision Making* 3(1): 145–165. http://dx.doi.org/10.1142/S0219622004000969

Jaakkola, T. S.; Haussler, D. 1998. Exploiting generative models in discriminative classifiers, *Advances in Neural Information Processing Systems* 11. MIT Press.

Jaakkola, T. S.; Haussler, D. 1999. Probabilistic Kernel regression models, in *Proceedings of the 1999 Conference on AI and Statistics*. Morgan Kaufmann.

Joachims, T. 1999a. Text categorization with support vector machines: learning with many relevant features, in *Proceedings of 10th European Conference on Machine Learning*, 137–142.

Joachims, T. 1999b. Transductive inference for text classification using support vector machines, in *Proceedings of 16th International Conference on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 200–209.

Joachims, T. 2000. Estimating the generalization performance of an SVM efficiently, in *Proceedings of the 17th International Conference on Machine Learning*. Morgan Kaufmann, San Franscisco, California, 431–438.

Joachims, T. 2006. Training linear SVMs in linear time, in *Proceedings of International Conference on Knowledge Discovery in Data Mining*, 217–226.

Johan, A. K. S.; Tony, V. G.; Jos, D. B.; Bart, D. M.; Joos, V. 2002. *Least Squares Support Vector Machines.* World Scientific.

Jonsson, K.; Kittler, J.; Matas, Y. P. 2002. Support vector machines for face authentication, *Journal of Image and Vision Computing* 20(5): 369–375. http://dx.doi.org/10.1016/S0262-8856(02)00009-4

Kaya, T.; Kahraman, C. 2011. A fuzzy approach to e-banking website quality assessment based on an integrated AHP-ELECTRE method, *Technological and Economic Development of Economy* 17(2): 313–334. http://dx.doi.org/10.3846/20294913.2011.583727

Keerthi, S. S.; Sundararajan, S.; Chang, K. W.; Hsieh, C. J.; Lin, C. J. 2008. A sequential dual method for large scale multi-class linear SVMs, in *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, 408–416.

Khemchandani, J. R.; Chandra, S. 2007. Twin support vector machines for pattern classification, *IEEE Transaction on Pattern Analysis and Machine Intelligence* 29(5): 905–910. http://dx.doi.org/10.1109/TPAMI.2007.1068

Kim, K. J. 2003. Financial time series forecasting using support vector machines, *Neurocomputing* 55(1): 307–319. http://dx.doi.org/10.1016/S0925-2312(03)00372-2

Klerk, E. 2002. *Aspects of Semidefinite Programming.* Kluwer Academic Publishers, Dordrecht.

Kochanek, K.; Tynan, S. 2010. The environmental risk assessment for decision support system for water management in the vicinity of open cast mines (DS WMVOC), *Technological and Economic Development of Economy* (16)3: 414–431. http://dx.doi.org/10.3846/tede.2010.26

Kunapuli, G.; Bennett, K.; Hu, J.; Pang, J. S. 2008. Bilevel model selection for support vector machines, *CRM Proceedings and Lecture Notes* 45: 129–158.

Li, J. P.; Chen, Z. Y.; Wei, L. W.; Xu, W. X.; Kou, G. 2007. Feature selection via least squares support feature machine, *International Journal of Information Technology and Decision Making* 6(4): 671–686. http://dx.doi.org/10.1142/S0219622007002733

Lin, C. F.; Wang, S. D. 2002. Fuzzy support vector machine, *IEEE Transaction on Neural Network* 13(2): 464–471. http://dx.doi.org/10.1109/72.991432

Lin, H. T.; Lin, C. J.; Weng, R. C. 2007. A note on Platts probabilistic outputs for support vector machines, *Machine Learning* 68: 267–276. http://dx.doi.org/10.1007/s10994-007-5018-6

Liu, Y.; Zhang, D.; Lu, G.; Ma, W. Y. 2007. A survey of content-based image retrieval with high-level semantics, *Pattern Recognition* 40(1): 262–282. http://dx.doi.org/10.1016/j.patcog.2006.04.045

Lodhi, H.; Cristianini, N.; Shawe-Taylor, J.; Watkins, C. 2000. Text classification using string kernels, *Advances in Neural Information Processing Systems* 13: 563–569.

Lu, J. W.; Plataniotis, K. N.; Ventesanopoulos, A. N. 2001. Face recognition using feature optimization and υ-support vector machine, in *Proceedings of the 2001 IEEE Signal Processing Society Workshop*, 373–382.

Ma, C.; Randolph, M. A.; Drish, J. 2001. A support vector machines-based rejection technique for speech recognition, in *Proceedings of IEEE International. Conference on Acoustics, Speech, and Signal Processing* 1: 381–384.

Mangasarian, O. L.; Wild, E. W. 2006. Nonlinear knowledge-based classifiers, *IEEE Transactions on Neural Networks* 19(10): 1826–1832. http://dx.doi.org/10.1109/TNN.2008.2005188

Mangasarian, O. L.; Wild, E. W. 2008. Multiple instance classification via successive linear programming, *Journal of Optimization Theory and Application* 137(1): 555–568. http://dx.doi.org/10.1007/s10957-007-9343-5

Martens, D.; Huysmans, J.; Setiono, R.; Vanthienen, J.; Baesens, B. 2008. Rule extraction from support vector machines: an overview of issues and application in credit scoring, *Studies in Computational Intelligence (SCI)* 80: 33–63. http://dx.doi.org/10.1007/978-3-540-75390-2_2

Melgani, F.; Bruzzone, L. 2004. Classification of hyperspectral remotesensing images with support vector machines, *IEEE Transactions on Geoscience and Remote Sensing* 42(8): 1778–1790. http://dx.doi.org/10.1109/TGRS.2004.831865

Min, J. H.; Lee, Y. C. 2005. Bankruptcy prediction using support vector machine with optimal choice of Kernel function parameters, *Expert Systems with Applications* 28(4): 603–614. http://dx.doi.org/10.1016/j.eswa.2004.12.008

Min, S. H.; Lee, J.; Han, I. 2006. Hybrid genetic algorithms and support vector machines for bankruptcy prediction, *Expert Systems with Applications* 31(3): 652–660. http://dx.doi.org/10.1016/j.eswa.2005.09.070

Mukkamala, S.; Janoski, G.; Sung, A. H. 2002. Intrusion detection using neural networks and support vector machines, in *Proceedings of IEEE International Joint Conference on Neural Network*, 1702–1707.

Nash, S. G.; Sofer, A. 1996. *Linear and Nonlinear Programming.* McGraw-Hill Companies, Inc. USA.

Núñez, H.; Angulo, C.; Català, A. 2002. Rule extraction from support vector machines, in *European Symposium on Artificial Neural Networks (ESANN)*, 107–112.

Peng, Y.; Kou, G.; Shi, Y.; Chen, Z. X. 2008. A descriptive framework for the field of data mining and knowledge discovery, *International Journal of Information Technology and Decision Making* 7(4): 639–682. http://dx.doi.org/10.1142/S0219622008003204

Peng, Y.; Kou, G.; Wang, G. X., *et al.* 2009. Empirical evaluation of classifiers for software risk management, *International Journal of Information Technology and Decision Making* 8(4): 749–767. http://dx.doi.org/10.1142/S0219622009003715

Platt, J. 1999. Fast training of support vector machines using sequential minimal optimization, in Schölkopf, B.; Burges, C. J. C.; Smola, A. J. (Eds.). *Advances in Kernel Methods Support Vector Learning*. Cambridge, MA: MIT Press, 185–208.

Platt, J. 2000. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods, in Smola, A.; Bartlett, P.; Schölkopf, B.; Schuurmans, D. (Eds.). *Advances in Large Margin Classifiers*. MIT Press, Cambridge, MA.

Schölkopf, B.; Smola, A. J. 2002. *Learning with Kernels–Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press.

Schweikert, G.; Zien, A.; Zeller, G.; Behr, J.; Dieterich, C., *et al.* 2009. mGene: accurate SVM-based gene finding with an application to nematode genomes, *Genome Research* 19: 2133–2143. http://dx.doi.org/10.1101/gr.090597.108

Segata, N.; Blanzieri, E. 2009. Fast local support vector machines for large datasets, *Machine Learning and Data Mining in Pattern Recognition, Lecture Notes in Computer Science* 5632: 295–310. http://dx.doi.org/10.1007/978-3-642-03070-3_22

Setiono, R.; Baesens, B.; Mues, C. 2006. Risk management and regulatory compliance: a data mining framework based on neural network rule extraction, in *Proceedings of the International Conference on Information Systems (ICIS'06)*. Available from Internet: http://www.springerlink.com/content/v837r344822815hr/fulltext.pdf

Shao, Y.; Zhang, C. H.; Wang, X. B.; Deng, N. Y. 2011. Improvements on twin support vector machines, *IEEE Transactions on Neural Networks* 22(6): 962–968. http://dx.doi.org/10.1109/TNN.2011.2130540

Shi, Y.; Peng, Y.; Kou, G.; Chen, Z. X. 2005. Classifying credit card accounts for business intelligence and decision making: a multiple-criteria quadratic programming approach, *International Journal of Information Technology and Decision Making* 4(4): 581–599. http://dx.doi.org/10.1142/S0219622005001775

Shin, K. S.; Lee, T. S.; Kim, H. J. 2005. An application of support vector machines in bankruptcy prediction model, *Expert Systems with Applications* 28(1): 127–135. http://dx.doi.org/10.1016/j.eswa.2004.08.009

Sonnenburg, S.; Rätsch, G.; Schäfer, C.; Schölkopf, B. 2006. Large scale multiple kernel learning, *Journal of Machine Learning Research* 7: 1–18.

Tan, J. Y.; Zhang, C. H.; Deng, N. Y. 2010. Cancer related gene identification via p-norm support vector machine, in *Proceeding of International Conference on Computational Systems Biology*, 101–108.

Tay, F. E. H.; Cao, L. J. 2001. Application of support vector machines in financial time series forecasting, *Omega* 29(4): 309–317. http://dx.doi.org/10.1016/S0305-0483(01)00026-3

Tay, F. E. H.; Cao, L. J. 2001. Improved financial time series forecasting by combining support vector machines with self-organizing feature map, *Intelligent Data Analysis* 5(4): 339–354. http://dx.doi.org/10.1016/S0925-2312(01)00676-2

Tay, F. E. H.; Cao, L. J. 2002. Modified support vector machines in financial time series forecasting, *Neurocomputing* 48(1): 847–861.

Tefas, A.; Kotropoulos, C.; Pitas, I. 2001. Using support vector machines to enhance the performance of elastic graph matching for frontal face authentication, *IEEE Transaction on Pattern Analysis and Machine Intelligence* 23(7): 735–746. http://dx.doi.org/10.1109/34.935847

Thomas, L. C.; Oliver, R. W.; Hand, D. J. 2005. A survey of the issues in consumer credit modeling research, *Journal of the Operational Research Society* 56: 1006–1015. http://dx.doi.org/10.1057/palgrave.jors.2602018

Tian, Q.; Hong, P.; Huang, T. S. 2000. Update relevant image weights for content based image retrieval using support vector machines, in *Proceedings of IEEE International Conference on Multimedia and Expo* 2: 1199–1202.

Tian, Y. J. 2005. *Support Vector Regression and Its Applications*: PhD Thesis. China Agricultural University.

Tian, Y. J.; Deng, N. Y. 2005. Leave-one-out bounds for support vector regression, in *Proceedings of the 2005 International Conference on Computational Intelligence for Modelling, Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, 1061–1066. http://dx.doi.org/10.1109/FSKD.2010.5569345

Tian, Y. J.; Yu, J.; Chen, W. J. 2010. lp-norm support vector machine with CCCP, in *Proceedings of 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery*, 1560–1564.

Tsochantaridis, I.; Joachims, T.; Hofmann, T.; Altun, Y. 2005. Large margin methods for structured and interdependent output variables, *Journal of Machine Learning Research* 6: 1453–1484.

Tsoumakas, G.; Katakis, I. 2007. Multi-label classification: an overview, *International Journal of Data Warehousing and Mining* 3(3): 1–13. http://dx.doi.org/10.4018/jdwm.2007070101

Tsoumakas, G.; Katakis, I.; Vlahavas, I. 2010. Mining multi-label data, *Data Mining and Knowledge Discovery Handbook* 6: 667–685.

Van Gestel, T.; Baesens, B.; Garcia, J.; Van Dijcke, P. 2003. A support vector machine approach to credit scoring, *Bank en Financiewezen* 2: 73–82.

Vanderbei, R. J. 2001. *Linear Programming: Foundations and Extensions.* Second edition. Kluwer Academic Publishers.

Vapnik, V. N. 1996. *The Nature of Statistical Learning Theory.* Springer. New York.

Vapnik, V. N. 1998. *Statistical Learning Theory.* New York: John Wiley and Sons.

Vapnik, V. N. 2006. *Estimation of Dependences Based on Empirical Data.* 2nd edition. Springer .Verlag, Berlin.

Vapnik, V. N.; Chapelle, O. 2000. Bounds on error expectation for SVM, in *Advances in Large-Margin Classifiers, Neural Information Processing*. MIT Press, 261–280.

Vapnik, V. N.; Vashist, A. 2009. A new learning paradigm: learning using privileged information, *Neural Networks* 22(5): 544–577. http://dx.doi.org/10.1016/j.neunet.2009.06.042

Wang, Y. Q.; Wang, S. Y.; Lai, K. K. 2005. A new fuzzy support vector machine to evaluate credit risk, *IEEE Transactions on Fuzzy Systems* 13(6): 820–831. http://dx.doi.org/10.1109/TFUZZ.2005.859320

Weston, J. 1999. Leave-one-out support vector machines, in *Proceedings of the International Joint Conference on Artificial Intelligence*, 727–731.

Weston, J.; Gammerman, A.; Stitson, M. O.; Vapnik, V. N.; Vovk, V.; Watkins, C. 1999. Support vector density estimation, in *Advances in Kernel Methods–Support Vector Learning*. Cambridge. MA: MIT Press, 293–305.

Weston, J.; Mukherjee, S.; Vapnik, V. 2001. Feature selection for svms, *Advances in Neural Information Processing Systems* 13: 668–674.

Wu, Q.; Ying, Y.; Zhou, D. X. 2007. Multi-kernel regularized classifiers, *Journal of Complexity* 23(1): 108–134. http://dx.doi.org/10.1016/j.jco.2006.06.007

Xu, L.; Schuurmans, D. 2005. Unsupervised and semi-supervised multi-class support vector machines, in *Proceedings of the 20th National Conference on Artificial Intelligence.*

Yang, Q.; Wu, X. D. 2006. 10 Challenging problems in data mining research, *International Journal of Information Technology and Decision Making* 5(4): 567–604. http://dx.doi.org/10.1142/S0219622006002258

Yang, S. X.; Tian, Y. J. 2011. Rule extraction from support vector machines and its applications, in *Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, 221–224. http://dx.doi.org/10.1109/WI-IAT.2011.132

Yang, Z. X. 2007. *Support Vector Ordinal Regression and Multi-class Problems*: PhD Thesis. China Agricultural University.

Yang, Z. X.; Deng, N. Y.; Tian, Y. J. 2005. A multi-class classification algorithm based on ordinal regression machine, in *Proceedings of International Conference on CIMCA& IAWTIC* 2: 810–815.

Yang, Z. X.; Tian, Y. J.; Deng, N. Y. 2009. Leave-one-out bounds for support vector ordinal regression machine, *Neural Computing and Applications* 18(7): 731–748. http://dx.doi.org/10.1007/s00521-008-0217-z

Yao, Y.; Marcialis, G. L.; Pontil, M.; Frasconi, P.; Roli, F. 2002. Combining flat and structured representations for fingerprint classification with recursive neural networks and support vector machines, *Pattern Recognition* 36(2): 397–406. http://dx.doi.org/10.1016/S0031-3203(02)00039-0

Yu, L.; Wang, S. Y.; Cao, J. 2009. A modified least squares support vector machine classifier with application to credit risk analysis, *International Journal of Information Technology and Decision Making* 8(4): 697–710. http://dx.doi.org/10.1142/S0219622009003600

Zanghirati, G.; Zanni, L. 2003. A parallel solver for large quadratic programs in training support vector machines, *Parallel Computing* 29(4): 535–551. http://dx.doi.org/10.1016/S0167-8191(03)00021-8

Zhang, C. H.; Tian, Y. J.; Deng, N. Y. 2010. The new interpretation of support vector machines on statistical learning theory, *Science China Mathematics* 53(1): 151–164. http://dx.doi.org/10.1016/S0167-8191(03)00021-8

Zhao, K.; Tian, Y. J.; Deng, N. Y. 2007. Unsupervised and semi-supervised lagrangian support vector machines, in *Proceedings of the 7th International Conference on Computational Science Workshops, Lecture Notes in Computer Science* 4489: 882–889. http://dx.doi.org/10.1007/978-3-540-72588-6_140

Zhao, K.; Tian, Y. J.; Deng, N. Y. 2006. Unsupervised and semi-supervised two-class support vector machines, in *Proceedings of the 6th IEEE International Conference on Data Mining Workshops*, 813–817.

Zhong, P.; Fukushima, M. 2007. Second order cone programming formulations for robust multi-class classification, *Neural Computation* 19(1): 258–282. http://dx.doi.org/10.1162/neco.2007.19.1.258

Zhou, L.; Lai, K. K.; Yen, J. 2009. Credit scoring models with AUC maximization based on weighted SVM, *International Journal of Information Technology and Decision Making* 8(4): 677–696.

Zhou, X.; Tuck, D. P. 2006. MSVM-RFE: extensions of SVM-RFE for multiclass gene selection on DNA microarray data, *Bioinformatics* 23(9): 1106–1114. http://dx.doi.org/10.1142/S0219622009003582

Zhu, J.; Rosset, S.; Hastie, T.; Tibshirani, R. 2004. 1-norm support vector machines, *Advances in Neural Information Processing Systems* 16: 49–56.

Zou, H.; Yuan, M. 2008. The F_∞-norm support vector machine, *Statistica Sinica* 18: 379–398. http://dx.doi.org/10.1093/bioinformatics/btm036

Yingjie TIAN. Doctor. Associate Professor of Research Center on Fictitious Economy & Data Science, Chinese Academy of Sciences. First degree in mathematics (1994), Master in applied mathematics (1997), PhD in Management Science and Engineering. He has published 4 books (one of which has been cited over 700 times), and over 50 papers in various journals and numerous conferences/proceedings papers. Research interests: support vector machines, optimization theory and applications, data mining, intelligent knowledge management, risk management.

Yong SHI. Doctor. He currently serves as the Executive Deputy Director, Research Center on Fictitious Economy & Data Science, Chinese Academy of Sciences. He has been the Charles W. and Margre H. Durham Distinguished Professor of Information Science and Technology, College of Information Science and Technology, Peter Kiewit Institute, University of Nebraska, USA in 1999–2009. Dr. Shi's research interests include business intelligence, data mining, and multiple criteria decision making. He has published more than 17 books, over 200 papers in various journals and numerous conferences/proceedings papers. He is the Editor-in-Chief of International Journal of Information Technology and Decision Making (SCI), and a member of Editorial Board for a number of academic journals. Dr. Shi has received many distinguished awards including the Georg Cantor Award of the International Society on Multiple Criteria Decision Making (MCDM), 2009; Outstanding Young Scientist Award, National Natural Science Foundation of China, 2001; and Speaker of Distinguished Visitors Program (DVP) for 1997–2000, IEEE Computer Society. He has consulted or worked on business projects for a number of international companies in data mining and knowledge management.

Xiaohui LIU. Doctor. Professor of Computing at Brunel University in the UK where he directs the Centre for Intelligent Data Analysis, conducting interdisciplinary research concerned with the effective analysis of data. He was Honorary Pascal Professor at Leiden University (2004) and Visiting Researcher at Harvard Medical School (2005). Professor Liu is a Charted Engineer, Life Member of the Association for the Advancement of Artificial Intelligence, Fellow of the Royal Statistical Society and Fellow of the British Computer Society. He has given numerous invited and keynote talks, chaired several international conferences, and advised funding agencies on interdisciplinary research programs. Collaborating with many talented physical, clinical and life scientists, Professor Liu has over 250 publications in biomedical informatics, data mining, dynamic and intelligent systems.