Applied Soft Computing 12 (2012) 2675–2682

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

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

# Evolution strategy based adaptive $L_q$ penalty support vector machines with Gauss kernel for credit risk analysis

Jianping Li[a,*], Gang Li[a,b], Dongxia Sun[a,b], Cheng-Few Lee[c]

[a] Institute of Policy and Management, Chinese Academy of Sciences, Beijing 100190, PR China
[b] Graduate University of Chinese Academy of Sciences, Beijing 100049, PR China
[c] Department of Finance and Economics, Rutgers Business School, Rutgers University, Piscataway, NJ 08854, USA

ARTICLE INFO

*Article history:*
Received 5 July 2011
Received in revised form 20 February 2012
Accepted 10 April 2012
Available online 25 April 2012

*Keywords:*
Adaptive penalty
Support vector machine
Credit risk classification
Evolution strategy

ABSTRACT

Credit risk analysis has long attracted great attention from both academic researchers and practitioners. However, the recent global financial crisis has made the issue even more important because of the need for further enhancement of accuracy of classification of borrowers. In this study an evolution strategy (ES) based adaptive $L_q$ SVM model with Gauss kernel (ES-A$L_q$G-SVM) is proposed for credit risk analysis. Support vector machine (SVM) is a classification method that has been extensively studied in recent years. Many improved SVM models have been proposed, with non-adaptive and pre-determined penalties. However, different credit data sets have different structures that are suitable for different penalty forms in real life. Moreover, the traditional parameter search methods, such as the grid search method, are time consuming. The proposed ES-based adaptive $L_q$ SVM model with Gauss kernel (ES-A$L_q$G-SVM) aims to solve these problems. The non-adaptive penalty is extended to $(0, 2]$ to fit different credit data structures, with the Gauss kernel, to improve classification accuracy.

For verification purpose, two UCI credit datasets and a real-life credit dataset are used to test our model. The experiment results show that the proposed approach performs better than See5, DT, MCCQP, SVM light and other popular algorithms listed in this study, and the computing speed is greatly improved, compared with the grid search method.



## 1. Introduction

Ever since the subprime mortgage crisis hit the United States and affected the whole world, deepening the global economic crisis further, credit risk has received increasing attention from both industry and academia. Numerous methodologies have been introduced for addressing the challenge and importance of credit risk analysis, risk evaluation, and risk management. Statistical and neural network based approaches are among the most popular paradigms [1–5].

Support vector machines (SVMs) have been used for credit risk analysis in the last few decades, as a promising credit risk analysis approach. It was first proposed by Vapnik [6,7]. It has been widely used in credit evaluation [8,9] and other fields, such as pattern classification, bioinformatics, and text categorization [10–12], due to its good generalization performance and strong theoretical foundations.

Although their strong theoretical foundations have been illustrated [6,7], the standard SVMs still have several drawbacks. For standard linearly separable problems, SVM attempts to optimize generalization performance, bound by separating data with a maximal margin classifier, which can be transformed into a constrained quadratic optimization problem. However, the data in real life are so complicated that most original credit data are not linearly separated and, therefore, the input space needs to be mapped into a higher dimensional feature space to make the data linearly separable [6,7,13]. As increasing attention is being put on learning kernel functions from data structures, kernels are introduced in the SVM for nonlinear transformations, in order to improve classification accuracy [13]. Many researchers have observed that kernels greatly improve the performance of SVM [14–16].

Computational complexity of credit risk analysis is another problem because it costs much time to solve the constrained quadratic programming problem, and to select the hyper-parameters and kernels for classification. Many modified versions of SVM have been proposed to deal with this drawback, such as LS-SVM and improved LS-SVM models [17,18,19,20,21]. However, these SVM and LS-SVM models fix the penalty forms ($L_2$) and cannot choose the optimal penalty adaptively, according to the dataset structures. If the dataset is sparse, an SVM with $L_2$ penalty cannot

* Corresponding author. Tel.: +86 105935 8805.
*E-mail addresses:* ljp@casipm.ac.cn (J. Li), ligang@casipm.ac.cn (G. Li), sdxnyb@163.com (D. Sun), lee@business.rutgers.edu (C.-F. Lee).

1568-4946/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2012.04.011

delete the noisy data, thus reducing the classification accuracy. As it is less sensitive to outliers of $L_1$, many researchers have chosen the $L_1$ penalty to improve the generalization performance, and have achieved better results on data sets having redundant samples or outliers [22,23]. But $L_1$ penalty SVM are not preferred if the data set is of a non-sparse structure. An adaptive $L_q$ SVM is proposed by Liu et al., which adaptively selects the optimal penalty, driven by the data [24]. However, that model adopted the linear kernel; consequently, the classification performance was not outstanding for the nonlinear separable credit datasets. Thus, the kernel is concerned in our study to improve the performance of this model for credit classification.

In addition, parameters optimization also needs to be considered in SVMs. Credit data are mass data; computation time increases as the size of training data increases, and performance of SVM deteriorates. Thus, some new methods need to be introduced to reduce the computational complexity of credit analysis. The Grid Search (GS) method is the most traditional and reliable optimization method for the parameter selection problem. To select optimal SVM parameters, GS is implemented by varying the SVM parameters through a wide range of values, with a fixed step size, assessing the performance of every parameter combination, based on a certain criterion, and finding the best setting in the end. However, the grid size will increase dramatically as the number of parameters increases. It is an exhaustive search method and very time-consuming. Therefore, a new method needs to be introduced to control computational costs. Evolutionary algorithms (EA) are iterative, direct, and randomized optimization methods inspired by principles of neo-Darwinian evolution theory. EA have higher computing efficiency than grid search, and can optimize parameters in non-differentiable functions. It has been proved to be suitable for hyper-parameter and feature selection for kernel-based learning algorithms [25,26]. Evolution strategies (ES), a branch of EA, were proposed by students at the Technical University of Berlin [27,28], which do not need coding and encoding processes and have good computing efficiency [29,30]. Especially, ES have self-adaptation search capacity, which quickly and nicely guides toward optimal points; some researchers have demonstrated that ES are suited to find the optimal hyper-parameters for SVMs and other multi-criteria programming models [31,32,33,34]. In this paper, we introduce an ES-based search method to find the optimal penalty and hyper-parameters simultaneously instead of Grid-based search method to save computing time.

In brief, in order to improve the performance of credit classification and control computational costs, this paper proposes an ES-based A$L_q$G-SVM approach. First, the Gauss kernel is applied to adaptive $L_q$ penalty support vector machines in this paper to better solve the nonlinear credit data problem. Second, evolution strategy is used to select optimal parameters instead of grid search in order to improve operation efficiency. The remainder of this paper is organized as follows. Section 2 reviews the relevant literature on SVM and adaptive $L_q$ SVM. Section 3 introduces the proposed ES-based A$L_q$G-SVM approach for improving the classification accuracy and the computing speed. Experimental results, compared with other popular models, are shown in Section 4. Conclusions are finally drawn in Section 5, with recommendations for future research.

## 2. Adaptive $L_q$ SVM

### 2.1. Support vector machine

The general form of support vector machine is used to solve binary classification problem. Given a set of credit data points $G = \{(\vec{x}_i, y_i)\}_{i=1}^{n}$, where $\vec{x}_i \in R^m$ is the $i$th input record and $y_i \in \{1, -1\}$ is the corresponding observed result, the main goal of SVM is to find an optimal separating hyper-plane, which can be represented as $\langle \vec{\varpi}, \vec{x} \rangle + b = 0$, and to maximize the separation margin (the distance between it and the nearest data point of each class), and to minimize the empirical classification error. Thus, the problem of seeking the optimal separating hyper-plane can be transformed into the following optimization problem:

$$\min \frac{1}{2}||\vec{\varpi}||^2 + C\sum_{i=1}^{n}\xi_i, \quad \text{s.t. } y_i(\vec{\varpi}^T\phi(x_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, n, \tag{1}$$

where variables $\xi_i$ are the non-negative slack variables, representing the classification error; regularized parameter $C$ is a constant denoting a trade-off between the maximum margin and the minimum experience risk.

The above quadratic optimization problem can be solved by transforming it into Lagrange function:

$$L(\vec{\varpi}, b, \xi, \alpha) = \frac{1}{2}||\vec{\varpi}||^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i(y_i(\vec{\varpi}^T\phi(\vec{x}_i) + b) - 1 + \xi_i) - \sum_{i=1}^{n}\theta_i\xi_i, \tag{2}$$

where $\alpha_i$, $\theta_i$ denote Lagrange multipliers; $\alpha_i \geq 0$ and $\theta_i \geq 0$.

At the optimal point, we have the following saddle point equations:

$\partial L/\partial\vec{\varpi} = 0$, $\partial L/\partial b = 0$ and $\partial L/\partial \xi_i = 0$, which translate into:

$$\vec{\varpi} = \sum_{i=1}^{n}\alpha_i y_i \phi(\vec{x}_i), \sum_{i=1}^{n}\alpha_i y_i = 0 \quad \text{and} \quad \alpha_i = C - \theta_i \quad i = 1, \ldots, n \tag{3}$$

By substituting (3) into (2), and by replacing $(\phi(\vec{x}_i) \cdot \phi(\vec{x}_j))$ with kernel functions $k(\vec{x}_i, \vec{x}_j)$, we get the dual optimization problem:

$$\max \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j k(\vec{x}_i, \vec{x}_j), \quad \text{s.t.} \sum_{i=1}^{n}\alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, n. \tag{4}$$

Commonly used kernel functions include polynomial, radial basis function (RBF) and sigmoid kernel, which are shown in (5)–(7), respectively.

$$k(\vec{x}, \vec{y}) = (\vec{x} \cdot \vec{y} + c)^d, \quad \text{for } d \in N, c \geq 0, \tag{5}$$

$$k(\vec{x}, \vec{y}) = \exp(-\gamma||\vec{x} - \vec{y}||^2), \quad \text{for } \gamma > 0, \tag{6}$$

$$k(\vec{x}, \vec{y}) = \tanh(\gamma(\vec{x} \cdot \vec{y}) + c), \quad \text{for } \gamma > 0, c \geq 0. \tag{7}$$

On solving the dual optimization problem, solution $\alpha_i$ determines the parameters of the optimal hyper-plane. This leads to the decision function, expressed as:

$$f(\vec{x}) = sgn\left(\sum_{i=1}^{n}y_i\alpha_i k(\vec{x}_i, \vec{x}) + b\right), \tag{8}$$

where $b = y_j - \sum_{i=1}^{n}y_i\alpha_i k(x_i, x_j)$, for some $j$, $\alpha_i \leq C$.

Usually, there is only a small subset of Lagrange multipliers $\alpha_i$ that is greater than zero in a classification problem. Training vectors having nonzero $\alpha_i^*$ are called support vectors; the optimal decision hyper-plane depends on them exclusively:

$$f(\vec{x}) = sgn\left(\sum_{i \in SV}y_i\alpha_i^* k(\vec{x}_i, \vec{x}) + b^*\right). \tag{9}$$