Knowledge-Based Systems 71 (2014) 290–302

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Robust non-convex least squares loss function for regression with outliers

Kuaini Wang, Ping Zhong *

College of Science, China Agricultural University, Beijing 100083, China

ARTICLE INFO

Article history:
Received 16 January 2014
Received in revised form 5 August 2014
Accepted 5 August 2014
Available online 12 August 2014

Keywords:
Least squares support vector regression
Robust
Iterative strategy
Loss function
DC program

ABSTRACT

In this paper, we propose a robust scheme for least squares support vector regression (LS-SVR), termed as RLS-SVR, which employs non-convex least squares loss function to overcome the limitation of LS-SVR that it is sensitive to outliers. Non-convex loss gives a constant penalty for any large outliers. The proposed loss function can be expressed by a difference of convex functions (DC). The resultant optimization is a DC program. It can be solved by utilizing the Concave–Convex Procedure (CCCP). RLS-SVR iteratively builds the regression function by solving a set of linear equations at one time. The proposed RLS-SVR includes the classical LS-SVR as its special case. Numerical experiments on both artificial datasets and benchmark datasets confirm the promising results of the proposed algorithm.

1. Introduction

Support vector machine (SVM), introduced by Vapnik and colleagues [1–3], has been a powerful machine learning technique for classification and regression estimation. It is based on the Vapnik–Chervonenkis (VC) dimensional theory and statistical learning theory. The central idea of SVM is to construct two parallel hyperplanes that separate the two classes with maximum margin. Over the past few decades, many efficient learning algorithms and models to SVM have emerged. Most of the above algorithms or models determine two parallel hyperplanes. Recently, some non-parallel hyperplane classifiers have been proposed, named twin SVMs [4–7]. Twin SVM generates two nonparallel hyperplanes by two smaller and related SVM-type problems, in which each hyperplane is closer to one class and as far as possible from the other. SVMs have been successfully used in many fields including pattern recognition [8], text categorization [9], time series prediction [10–12].

As for the regression estimation problem, one is given the training samples of input vectors $\{\mathbf{x}_i\}_{i=1}^n$ along with the corresponding targets $\{y_i\}_{i=1}^n$, and the task is to find a regression function that best represents the relation between input vectors and their targets. A nonlinear regressor makes predictions by $f(\mathbf{x}) = w^\top \phi(\mathbf{x}) + b$, where $\phi(\cdot)$ is a mapping which maps the input data into a high-dimensional feature space, w represents the model complexity, and b is the bias. The weight vector w and the bias b are determined by minimizing the regularized risk function $R_{\mathrm{emp}}[f] + \lambda||w||^2$, where $R_{\mathrm{emp}}[f]$ is the empirical risk of a specific loss function and $\lambda > 0$ is the regularization parameter. A successful method for regression estimation is least squares support vector regression (LS-SVR) introduced in [13,14], which tries to minimize the least squares errors on the training samples while simultaneously escaping from overfitting. LS-SVR is different from the classical SVR which solves a quadratic programming problem (QPP). LS-SVR replaces the QPP in SVR with a set of linear equations by using a squared loss function and leads to an extremely fast training speed.

In real applications, the samples obtained may be subject to outliers. Outliers occur for various reasons, such as erroneous samplings and measurements or noisy samples with the heavy-tailed noise distribution. Traditionally, although LS-SVR obtains fast training speed and comparable generalization, it still exits one obvious limitation that its solution suffers from lack of robustness. In LS-SVR, the squared errors result in bad robustness. LS-SVR is only optimal if the error variables follow a Gaussian distribution because it attempts to minimize the sum of squared error loss of these samples.

To improve the robustness of LS-SVR, many researchers have made much effort in recent years. The commonly used approach adopts the weight setting strategies to reduce the influence of outliers. Since the training samples that include outliers are aggregately regarded in the training process, it is important to consider the concept of robust statistic. For this reason, Suykens et al. proposed a weighted LS-SVR in which different weighting factors are put on the error variables, such that the less important

* Corresponding author.
E-mail address: zping@cau.edu.cn (P. Zhong).

http://dx.doi.org/10.1016/j.knosys.2014.08.003
0950-7051/© 2014 Elsevier B.V. All rights reserved.

samples or outliers have small weights [14]. Another alternative calculation rule of weights was that the samples which had large distances from other samples should be assigned smaller weights to reduce their impact [15,16]. In [17], the authors compared four different types of weighting function, including Huber, Hampel, Logistic and Myriad, and gained the conclusion that Logistic and Myriad weighting functions owned better robustness over the other two functions in most cases. In essence, the weighted LS-SVR [14] is an LS-SVR with Hampel weighting function [17]. However, whether these weighting strategies are the optimal choice with respect to dataset is unclear. Another kind of methods enhances LS-SVR by outlier elimination [18,19].

Recently, various works focused on non-convex loss functions have been proposed as they have shown superiority to convex ones in generalization performance and robustness [20–26]. Xu et al. [20] studied training algorithms for SVMs with the ramp loss and solved the non-convex optimization by utilizing semidefinite program and convex relaxation techniques. Wu and Liu [23] proposed a robust SVM with truncated hinge loss, which was illustrated to be more robust to outliers and derived more accuracy classifiers. Collobert et al. [21,22] pointed out the scalability advantages of non-convex approaches and used the CCCP [29] for non-convex optimization to achieve faster batch SVMs and Transductive SVMs. Motivated by the recent interest in solving SVM in the primal [27,28], Wang et al. [24] gave robust support vector machine with smooth ramp loss in the primal. Zhao and Sun [25] extended the similar idea to regression estimation. Zhong [26] presented a smooth non-convex loss function for robust SVR. These works only focus on the classical support vector classification or regression, whereas there are few papers focus on the least squares version of SVR.

Motivated by the aforementioned studies, in this paper, we propose a non-convex least squares loss function by setting constant penalty for any large outliers to reduce the danger from these samples, and derive a robust LS-SVR model (RLS-SVR). Due to the fact that the proposed loss function is not convex, the classical optimization method cannot be employed directly to solve the RLS-SVR. First, we decompose the non-convex loss function into a difference of convex functions. The resultant optimization problem is a DC program [30,31]. Second, the CCCP is used to transform the DC program into a sequences of convex optimization problem. RLS-SVR iteratively builds the regression function by solving a set of linear equations at one time. Numerical experiments on both artificial datasets and benchmark datasets reveal the efficiency of the proposed method.

The rest of this paper is organized as follows. In Section 2, we present a background on LS-SVR. Section 3 formulates a robust scheme for LS-SVR with non-convex loss. We propose a non-convex least squares loss function and derive RLS-SVR, and give an iterative algorithm for the proposed RLS-SVR. Section 4 performs experiments on artificial datasets and benchmark datasets to investigate the effectiveness of the RLS-SVR. The last section concludes the paper.

2. LS-SVR

In this section, we concisely present the basic principles of LS-SVR. For more detail, the reader can refer to [13,14]. To derive a nonlinear regressor, LS-SVR solves the following optimization problem:

$$\min_{w,b,e_i} \quad \frac{1}{2}\|w\|^2 + \frac{C}{2}\sum_{i=1}^{n} e_i^2 \tag{1}$$

$$\text{s.t.} \quad y_i = w^\top \phi(\mathbf{x}_i) + b + e_i, \quad i = 1,\ldots,n \tag{2}$$

where e_i represents the error variables, and $C > 0$ is the regularization parameter that balances the model complexity and empirical risk. We introduce Lagrangian multipliers and construct a Lagrangian function to solve the optimization problem (1) and (2). Utilizing the Karush–Kuhn–Tucker (KKT) conditions, the dual problem can be obtained as:

$$\begin{bmatrix} 0 & \mathbf{1}^\top \\ \mathbf{1} & K + \frac{1}{C} I_n \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \tag{3}$$

where $\mathbf{1} = (1,1,\ldots,1)^\top$, $y = (y_1, y_2, \ldots, y_n)^\top$, I_n denotes $n \times n$ identity matrix, $K = (K_{ij})_{n\times n}$ is the kernel matrix with $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$, and $k(\cdot,\cdot)$ is the kernel function which can be expressed as the inner product calculation in high dimensional feature space. For a new sample $\mathbf{x}$, we can predict its target by

$$f(\mathbf{x}) = w^\top \phi(\mathbf{x}) + b = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x}) + b \tag{4}$$

where α and b are the solutions of (3).

3. Robust least squares support vector regression (RLS-SVR)

3.1. Non-convex least squares loss function

The optimization problem of LS-SVR (1) and (2) can be rewritten as a unconstrained regularized model:

$$\min_{w,b} \quad \frac{1}{2}\|w\|^2 + \frac{C}{2}\sum_{i=1}^{n} l_1(y_i - f(\mathbf{x}_i)) \tag{5}$$

where

$$l_1(r) = r^2 \tag{6}$$

is squared loss function, and $\sum_{i=1}^{n} l_1(y_i - f(\mathbf{x}_i))$ expresses the empirical risk.

As mentioned, LS-SVR is sensitive to outliers and noises with the $l_1(r) = r^2$. When there exist outliers which are markedly far away from the rest of samples, large errors will dominate the sum of squared error and the decision hyperplane of LS-SVR will severely deviate from the original position and thus deteriorate the generalization performance of LS-SVR. By setting a constant penalty θ^2 for any large outlier, we propose a non-convex least squares loss function (see Fig. 1):

$$l_\theta(r) = \begin{cases} r^2, & \text{if } |r| \leqslant \theta \\ \theta^2, & \text{if } |r| > \theta \end{cases} \tag{7}$$

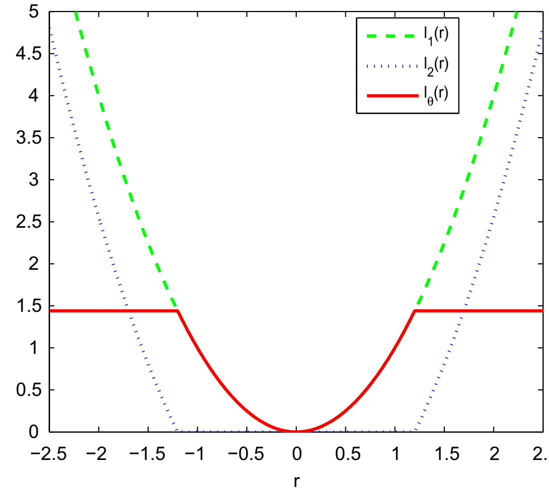

Fig. 1. Non-convex least squares loss function $l_\theta(r)$ with $\theta = 1.2$, squared loss function $l_1(r)$ and $l_2(r)$.