Expert Systems with Applications 37 (2010) 7781–7787

Contents lists available at ScienceDirect

Expert Systems with Applications

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

A reduced data set method for support vector regression

Horng-Lin Shieh *, Cheng-Chien Kuo

Department of Electrical Engineering, Saint John's University, Taipei, Taiwan

ARTICLE INFO

Keywords:
Support vector regression
Outlier
Fuzzy clustering
Robust fuzzy *c*-means

ABSTRACT

Support vector regression (SVR) has been very successful in pattern recognition, text categorization, and function approximation. The theory of SVR is based on the idea of structural risk minimization. In real application systems, data domain often suffers from noise and outliers. When there is noise and/or outliers exist in sampling data, the SVR may try to fit those improper data, and obtained systems may have the phenomenon of overfitting. In addition, the memory space for storing the kernel matrix of SVR will be increment with $O(N^2)$, where N is the number of training data. Hence, for a large training data set, the kernel matrix cannot be saved in the memory. In this paper, a reduced support vector regression is proposed for nonlinear function approximation problems with noise and outliers. The core idea of this approach is to adopt fuzzy clustering and a robust fuzzy *c*-means (RFCM) algorithm to reduce the computational time of SVR and greatly mitigates the influence of data noise and outliers.

1. Introduction

The theory of support vector machines (SVM) developed by Vapnik (1995) in 1995 is gaining in popularity due to its many attractive features. The SVM is based on the idea of structural risk minimization (SRM) and has been shown to be superior to traditional empirical risk minimization (ERM) principles employed by conventional neural networks (Gunn, 1998). SVM has been successfully applied to a number of applications, such as classification, time predictions, pattern recognition, and regression (Burges, 1998; Jair, Xiaoou, Wen, & Kang, 2008; Kamruzzaman & Begg, 2006; Kumar, Kulkarni, Jayaraman, & Kulkarni, 2004; Lijuan, 2003; Wong & Hsu, 2006; Zhou, Zhang, & Jiao, 2002). In many intelligent systems, SVM has been shown to provide higher performance than traditional learning machines, and has thus been adopted as a tool for solving classification issues (Lin & Wang, 2002). Over the past few years, a lot of researchers of neural networks and machine learning fields are attracted to devoting themselves to research on SVM (Wang & Xu, 2004).

The SVM is systematic and properly motivated by the statistical learning theory (Vapnik, 1998). Training of the SVM involves optimization of a convex cost function and globally minimizes to complete the learning process (Campbell, 2002). In addition, SVM can handle large input, and can automatically identify a small subset consisting of informative points, namely *support vectors* (Gustavo et al., 2004). The SVM can also be applied to regression problems by the introduction of an alternative loss function (Gunn, 1998). Such approaches are often called support vector regression (SVR).

SVM maps the input data into a high-dimensional feature space, and searches a separate hyperplane that maximizes the margin between two classes. SVM adopts quadratic programming (QP) to maximize the margin as computing tasks become very challenging when the number of data is beyond a few thousand (Hu & Song, 2004). For example, in Fig. 1, there are 500 sampling data generated from a *sin* wave with Gaussian noise $N(0,0.447)$. The SVR algorithm is adopted to construct this function. The entries of the kernel matrix of SVR are floating-points numbers, and each floating-point number requires 4 bytes for storing. Therefore, the total memory required is $500 * 500 * 4 = 1000{,}000$ bytes. The SVR algorithm is performed on a Pentium 4, 1.8 GHz with 128 MB of memory running Windows XP. The total execution time of the simulation is 21941 s (above 6 h). This execution time is very long and the memory requirements are very large for real applications of science.

Osuna, Freund, and Girosi (1997) proposed a generalized decomposition strategy for the standard SVM, in which the original QP problem is replaced by a series of smaller sub-problems, which are proved able to converge to a global optimum point. However, it is well known that the decomposition process relies heavily on the selection of a good working set of the data, which normally starts with a random subset (Hu & Song, 2004). Lee and Huang (2007) proposed to restrict the number of support vectors by solving reduced support vector machines (RSVM). The main characteristic of this method is to reduce the matrix from $l \times l$ to $l \times m$, where m is the size of a randomly selected subset of training data that are considered as candidates of support vectors. The smaller matrix

* Corresponding author.
E-mail address: shieh@mail.sju.edu.tw (H.-L. Shieh).

0957-4174/$ - see front matter Crown Copyright © 2010 Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2010.04.062

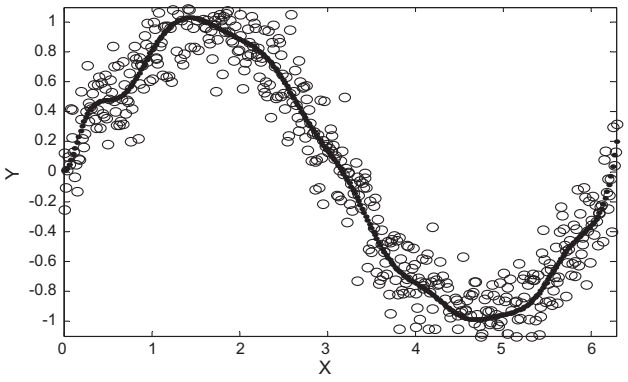

Fig. 1. Sin wave with noise $N(0, 0.447)$.

can easily be stored in memory, and then optimization algorithms, such as the Newton method, can be applied (Lin & Lin, 2003). However, as shown by Lin and Lin (2003), numerical experiments show that the accuracy of RSVM is usually lower than that of SVM.

Because support vectors can state the distributional features of all data, according to the characteristics of SVM, removing trivial data from the whole training set will not greatly affect the outcome, but will effectively increase the training process (Wang & Xu, 2004). A reduced set method based on the measurement of similarities between samples is developed by Wang and Xu (2004). In this paper, the samples similar to some data points will be discarded under a pre-established similarity threshold. In other words, these samples are so similar to the special data point that their influence on the prediction function can be ignored. According to this method, a large number of training vectors are discarded, and then a faster SVM training can be obtained without compromising the generalization capability of SVM. However, like the *K*-means clustering algorithm, the disadvantage of this algorithm is that the number of clusters must be predetermined, but in some real applications, there is no information to predefine the number of the clusters.

In real applications, data is bound to have noise and outliers, and algorithms utilized in engineering and scientific applications must be robust in order to process these data. In system modeling with noise and/or outliers existing in the sampling data, the system models may try to fit those improper data, and the output may have the phenomenon of overfitting (Chung, Su, & Hsiao, 2000; Shieh, Yang, Chang, & Jeng, 2009). SVR has been shown to have excellent performance for both the ε-insensitive and Huber's robust function for matching the correct type of noise in an application of time series prediction (Mukherjee, Osuna, & Girosi, 1997). However, in this SVR approach, outliers may possibly be taken as support vectors, and such an inclusion of outliers in support vectors may lead to serious overfitting phenomena (Chung, 2000).

In this paper, in order to overcome the above problems, a robust fuzzy clustering method is proposed to greatly mitigate the influence of noise and outliers in sampling data, and then the SVR method is used to construct the system models. Three experiments are illustrated, and their results have shown the proposed approach has better performance and less execution time than the original SVR method in various kinds of data domains with data noise and outliers.

2. Support vector regression

The model of learning from examples can be considered a general statistic framework of minimizing expected loss using sampling data. Suppose there are n random independent identically distributed (i.i.d.) data $(\mathbf{x}_1, y_1)$, $(\mathbf{x}_2, y_2)$, …, $(\mathbf{x}_n, y_n)$, where $\mathbf{x}_i \in R^d$, $y_i \in R$, $i = 1, 2, \ldots, n$ drawn according to the uniform probability distribution function $P(\mathbf{x}, y) = P(\mathbf{x})P(y|\mathbf{x})$. Given a set of functions $f(x, \alpha)$, $\alpha \in \Lambda$, where Λ is a parameter set, from which the goal of the learning process is to choose a function $f(x, \alpha_0)$ that can obtain the best relationship between input and output pairs. Consider a measure of the *loss* $L(y, f(x, \alpha_0))$ between the output y of the sampling data to a given input x, and the response $f(x, \alpha_0)$, provided by the learning machine. In order to obtain $f(x, \alpha_0)$, one has to minimize the expected *risk functional*

$$R(\alpha) = \int L(y, f(x, \alpha_0)) dP(x, y), \tag{1}$$

A common choice for the loss function is L_2-norm; i.e., $L(e) = e^2 = (y - f(x, \alpha_0))^2$. However, because $P(\mathbf{x}, y)$ is unknown, $R(\alpha)$ cannot be directly evaluated from Eq. (1). In general, the expected risk function is replaced by the *empirical risk functional*

$$R_{emp}(\alpha) = \frac{1}{n} \sum_{i=1}^{n} L(y, f(x, \alpha_0)). \tag{2}$$

There is no probability distribution in Eq. (2). However, in real application systems, data domains often suffer from noise and outliers. When there is noise and/or outliers exist in sampling data, Eq. (2) may try to fit those improper data and obtained systems may have the phenomenon of overfitting.

Let the sampling data be represented as $\{(\mathbf{x}_i, y_i) | \mathbf{x}_i \in R^d, y_i \in \{-1, 1\}\}$, $i = 1, 2, \ldots n$. In the SVR method, the regression function is approximated by the following function as:

$$f = \sum_{i=1}^{n} \mathbf{w}_i \boldsymbol{\varphi}(\mathbf{x}_i) + b, \tag{3}$$

where $\{\boldsymbol{\varphi}(\mathbf{x}_i)\}_{i=1}^{n}$ are the features of inputs, $\{\mathbf{w}_i\}_{i=1}^{n}$ and b are coefficients. The coefficients are estimated by minimizing the regularized risk function (Wang & Xu, 2004)

$$R(C) = C \frac{1}{n} \sum_{i=1}^{n} L(y, f) + \frac{1}{2} \|\mathbf{w}\|^2, \tag{4}$$

where $L(y, f)$ adopt the ε-insensitive loss function, and is defined as follows:

$$L(y, f) = \begin{cases} |y - f| - \varepsilon, & |y - f| \geqslant \varepsilon, \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

and $\varepsilon \geqslant 0$ is a predefined parameter.

In Eq. (4), the second term, $\frac{1}{2}\|\mathbf{w}\|^2$, is used for the flatness measurement of function (3), and C is a regular constant determining the tradeoff between the training error and the model flatness. SVR introduces slack variables ξ, ξ^* and leads Eq. (4) to the following constrained function (Wang & Xu, 2004):

minimize

$$R(\mathbf{w}, \xi^*) = C^* \sum_{i=1}^{n} (\xi_i + \xi_i^*) + \frac{1}{2} \|\mathbf{w}\|^2, \tag{6}$$

subject to

$$\begin{aligned} &\mathbf{w}\boldsymbol{\varphi}(\mathbf{x}_i) + b - y_i \leqslant \varepsilon + \xi_i^*, \\ &y_i - \mathbf{w}\boldsymbol{\varphi}(x_i) - b_i \leqslant \varepsilon + \xi_i, \\ &\xi, \xi^* \geqslant 0, \end{aligned} \tag{7}$$

where ξ, ξ^* are slack variables representing upper and lower constraints on the outputs of the system. Thus, function (3) becomes the explicit form:

$$f(\mathbf{x}, \alpha_i, \alpha_i^*) = \sum_{i=1}^{n} \mathbf{w}_i \boldsymbol{\varphi}(\mathbf{x}_i) + b = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) \boldsymbol{\varphi}(\mathbf{x}_i)^T \boldsymbol{\varphi}(\mathbf{x}_i) + b \tag{8}$$