سومین کنفرانس فناوری اطلاعات و دانش

۶ تا ۸ آذرماه ۱۳۸۶

A Neural Network-Based Training Scheme for Probabilistic Support Vector Machine

Sohrab Effati
Department of Mathematics
Tarbiat Moallem University of Sabzevar
Sabzevar, Iran
effati@sttu.ac.ir

Hadi Sadoghi Yazdi
Engineering Department
Tarbiat Moallem University of Sabzevar
Sabzevar, Iran
sadoghi@sttu.ac.ir

Zahra Saberi
Department of Mathematics
Tarbiat Moallem University of Sabzevar
Sabzevar, Iran
z_saberi2007@yahoo.com

Abstract— **In this paper a new scheme is proposed for training of the support vector machine (SVM) with probabilistic constraints. In the SVM, a pattern recognition problem is converted to a constraint quadratic programming. Each constraint is related to a training sample thereupon noisy data is caused appearance of incorrect support vector. Probabilistic constraints admit presence probably of samples in each class is applied based on a distribution function for determining suitable support vectors. In this way, it is possible noisy samples have low effect for finding support vectors. In the proposed method, SVM with the probabilistic constraints is converted to form of neural network model which can be described by the nonlinear dynamical system. A set of differential equations are defined for modeling of this dynamic and converges to optimal solution for the SVM with the probabilistic constraints. Another feature of the proposed method is solving both the primal and dual problem of SVM. Experimental results show the capability of the proposed method relative to conventional SVM.**

1. INTRODUCTION

Support vector machines (SVMs) as originally introduced by Vapnik within the area of statistical learning theory and structural risk minimization [1] and create a classifier with minimized VC dimension. In statistical learning theory, the support vector machine (SVM) has been developed for data classification and function estimation and it is used in wide range applications such as optical character recognition [2, and 3], text categorization [4], face detection in images [5], vehicle tracking in video sequence [6], nonlinear equalization in communication systems [7], and generating of fuzzy rule based system using SVM framework [8, and 9]. This paper emphasis over noisy learning data and learning scheme. Following text proceeds to literature about these problems.

As shown in [11, 12], SVM is very sensitive to outliers or noises since the penalty term of SVM treats every data point equally in the training process. This may result in the occurrence of over fitting problem if one or few data points have relatively very large values of slack variable. The fuzzy SVM (FSVM) to deal with the over fitting problem. FSVM is an extension of SVM that takes into account the different significance of the training samples. For FSVM, each training sample is associated with a fuzzy membership value. The membership value reflects the fidelity of the data; in other words, how confident we are about the actual class information of the data. The higher value show the more confident about that class. The optimization problem of the FSVM is formulated in [10, and 14] and have used in works such as [13, 15, and 16]. In this method slack variable is scaled by the membership value. The fuzzy membership values are used to weight the soft penalty term in the cost function of SVM. The weighted soft penalty term reflects the relative fidelity of the training samples during training. Important samples with larger membership values will have more impact in the FSVM training than others.

In addition SVM is a constraint quadratic programming problem can be solved by using existing methods. Traditional algorithms for digital computers may not be efficient since the computing time required for a solution is greatly dependent on the dimension and structure of the problems. One possible and reliable approach to real-time optimization is using artificial neural networks. Because of the inherent massive parallelism, the neural network approach can solve optimization problems in running time at the orders of magnitude much faster than the most popular optimization algorithms executed on general-purpose digital computers.

In 1985 and 1986 Hopfield and Tank [18,19] proposed a neural network for solving linear programming problems. Their main work has inspired many researchers to investigate alternative neural networks for solving linear and nonlinear programming problems. In 1987, Kennedy and Chua [20] proposed an improved model that always converges to an approximation of the optimal solution. In