National Seminar on Control and Optimization
Nov 13-14, 2019
سومین سمینار ملی کنترل و بهینه‌سازی
۲۲-۲۳ آبان ۱۳۹۸

CLASSIFYING RANDOM VARIABLES BASED ON SUPPORT VECTOR MACHINE BY NEURAL NETWORK

AMIR FEIZI [1]*, ALI REZA NAZEMI [1], AND MOHAMMAD REZA RABIEE[2]

[1] *Department of Mathematics, School of Mathematical Sciences, Shahrood University of Technology, P.O. Box 3619995161-316, Shahrood, Iran. Tel./Fax: +98 23 32300235 .*
amirfeizi93@gmail.com; nazemi20042003@gmail.com
rabie.mr87@gmail.com

Abstract. In this paper, a version of Support Vector Machine (SVM) is proposed which any of training samples are considered the random variables. Hence, in order to achieve robustness, the constraint in SVM must be replaced with probability of constraint. In this model by applying the nonparametric statistical methods, we obtain the optimal separating hyperplane by using a quadratic optimization problem and solve it with a neural network.

1. Introduction

Support vector machines (SVMs) are powerful tools for data classifcation and regression. In the recent years, many fast algorithms for SVMs have been developed. In many engineering and military applications, the demands on real-time data processing is often needed, such as classifcation in complex electromagnetic environments, recognition in medical diagnostics radar object recognition in strong background clutter, etc. However, the conventional numerical methods require much more computation time and cannot satisfy real-time requirement. One possible and promising approach to train SVMs in real-time is to employ recurrent neural networks based on circuit implementation. The standard SVM suppose that the training data are known exactly. However in practical classification problems such as sensor database, location database, biometric information systems, the samples can not be observed exactly because of sampling errors, modeling errors and/or measurement errors. In this case, standard SVM may not be successful.
The main motivation of this paper relies on random variables and probabilistic constraint to construct optimal separating hyperplane by using the SVM and solving the optimal problem by neural network.

Key words and phrases. Neural Network, Classifying, Random Variables, Support Vector Machine .
* Speaker.

2. Support Vector Machine for Classification

This section appropriates to a brief introduction on SVM. Let $S = \{(\mathbf{x}_i, y_i); i = 1, ..., n\}$ be a set of n training samples, where $\mathbf{x}_i = [x_i^1, ..., x_i^m]^T$ is a m-dimensional sample in the input space and $y_i \in \{-1, 1\}$ is the class label of $\mathbf{x}_i$. SVM finds the optimal separating hyperplane with the minimal classification errors. Let $\mathbf{w}_0$ and b_0 denote the optimum values of the weight vector and bias respectively. The hyperplane can be represented as $\mathbf{w}_0^T\mathbf{x} + b = 0$, where $\mathbf{w}_0 = [w_{01}, ..., w_{0m}]^T$ and $\mathbf{x} = [x^1, ..., x^m]^T$, $\mathbf{w}_0$ is the normal vector of the hyperplane and b_0 is the bias that is a scalar. The optimal hyperplane can be obtained by solving the following optimization problem

$$\begin{array}{ll} Minimize & \frac{1}{2}\|\mathbf{w}\ \|^2 + C\sum_{i=1}^{n}\zeta_i \\ subject\ \ to & \left\{ \begin{array}{l} y_i(\mathbf{w}^T\mathbf{x}_i - b) \geq 1 - \zeta_i, \\ \zeta_i \geq 0. \quad i = 1, ..., n. \end{array} \right. \end{array} \tag{2.1}$$

It follows from the saddle point condition that the partial derivatives of Lagrange function with respect to the primal variables $\mathbf{w}$,b ,ζ_i have to vanish for optimality.

3. The Proposed Probabilistic Constraints SVM in Linear Case

Frequently in practical classification problems, training data cannot be observed precisely because of sampling errors, modeling errors or measurement errors. In this section we investigate the SVM with uncertain input data in a stochastic framework. Suppose that $\{(X_i, y_i) : i = 1, ..., n\}$ be a set of n training samples. Let $X_i = [X_i^1, ...X_i^m]^T$ be ith training sample, where is a random vector in the input space with mathematical expectation $E(X_i) = [E(X_i^1), ..., E(X_i^m)]^T$ and $y_i \in \{-1, 1\}$ is the class label of X_i. In order to achieve robustness, the constraints in 2.1 must be replaced with probability constraints. Probabilistic constraints SVM finds the optimal separating hyperplane with the minimal classification errors.
optimal separating hyperplane can be obtained by solving the following optimization problem:

$$\begin{array}{ll} Minimize & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\zeta_i \\ subject\ \ to & \left\{ \begin{array}{l} P_r(y_i(\mathbf{w}^T\mathbf{X}_i - b) \geq 1 - \zeta_i) \geq \delta_i, \\ \zeta_i \geq 0. \quad i = 1, ..., n. \end{array} \right. \end{array} \tag{3.1}$$

where $\delta_i \in [0, 1]$ is value of effect of ith sample determination of optimal hyperplane position.
we now derive sufficient condition for this constraint and convert the optimization problem into a solvable QP.

Theorem 3.1. *Suppose that* $\boldsymbol{X}_i = [X_i^1, ..., X_i^m]^T$ *is a random vector with known mathematical expectation* $E(\boldsymbol{X}_i) = [E(X_i^1), ..., E(X_i^m)]^T$. *Then a sufficient condition for holding* $P_r(y_i(\boldsymbol{w}^T\boldsymbol{X}_i - b) \geq 1 - \zeta_i) \geq \delta_i$ *is* $y_i(\boldsymbol{w}^T E(\boldsymbol{X}_i) + b) \geq 2a\delta_i + 1 - \zeta_i$ *where* $a > 1$.

Proof. [1] □

Then optimal separating hyperplane can be obtained by solving the following optimization problem

$$\begin{aligned} &Minimize \qquad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\zeta_i \\ &subject \quad to \quad \begin{cases} y_i(\mathbf{w}^T E(\mathbf{X}_i) + b) \geq 2a\delta_i + 1 - \zeta_i, \\ \zeta_i \geq 0. \quad i = 1, ..., n. \end{cases} \end{aligned} \tag{3.2}$$

4. The Proposed Probabilistic Constraints SVM In Nonlinear Case

Nonlinear support vector machine is accomplished by fitting a linear classifier in a higher dimensional feature space. A nonlinear transformation ϕ is used to transform data points from the input space of dimension m into a feature space having a higher dimension m_1. The nonlinear mapping is denoted by $\phi : R^m \to R^{m_1}$. Similar to the linear case, optimal separating hyperplane can be obtained by solving the following optimization problem(see [1])

$$\begin{aligned} &Minimize \qquad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\zeta_i \\ &subject \quad to \quad \begin{cases} y_i(\mathbf{w}^T E(\phi(\mathbf{X}_i)) + b) \geq 2a\delta_i + 1 - \zeta_i, \\ \zeta_i \geq 0. \quad i = 1, ..., n. \end{cases} \end{aligned} \tag{4.1}$$

5. A neural network model

Consider the following optimization problem

$$\begin{aligned} min \quad & \frac{1}{2}\theta^T Q\theta + d^T\theta \\ s.t. \quad & A\theta - b \leq 0, \\ & e^T\theta = 0. \end{aligned} \tag{5.1}$$

It is shown in [2] that this problem is solved by the following neural network:

$$\begin{aligned} \frac{dy}{dt} &= \tau\psi(y), \\ y(t_0) = y_0 &= (\theta_0^T, \nu_0^T, \vartheta_0^T)^T, \tau > 0, \end{aligned} \tag{5.2}$$

where $y = (\theta^T, \nu^T, \vartheta^T)^T \in R^{3N+1}$, τ is a scale parameter, $(x)^+ = Max\{x, 0\}$ and

$$\psi(y) = \begin{pmatrix} -(Q\theta + d + A^T(\nu + A\theta - b)^+ + e\vartheta) \\ (\nu + A\theta - b)^+ - \nu \\ e^T\theta \end{pmatrix}$$

It should be noted that τ indicates the convergence rate of the neural network 5.2. The stability and convergence property of this neural network is proven in [2]

6. Numerical simulations

We consider thirty data as in figure 1

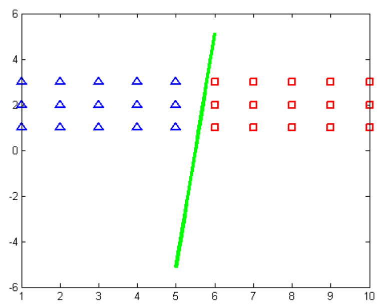

Figure 1. Results of support vector classyfing using the neural network Eqs 5.2

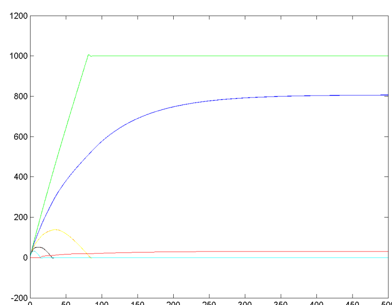

Figure 2. Transient behaviors $\theta_1, ..., \theta_{31}$ of the neural network Eqs 5.2

7. Acknowledgement

thankful to Dr. Ali Reza Nazemi and Dr. Mohammad Reza Rabiee for their very helpful guidances and comments.

References

[1] m. Abaszade , S. Effati, *A New Method for Classifying Random Variables Based on Support Vector Machine* , Journal of Classification **36** (2019) .

[2] Feizi A. and Nazemi A. R. (2017), An application of a practical neural network model for solving support vector regression problems , Intelligent Data Analysis, 21, 1443–1461 .