Expert Systems with Applications 37 (2010) 3495–3498

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Support vector machine for classification based on fuzzy training data [☆]

Ai-bing Ji [a,b,*], Jia-hong Pang [b], Hong-jie Qiu [b]

[a] Department of Computer Science and Engineering, Nanjing University of Aeronautics and Astronautics, 210016 Nanjing, PR China
[b] College of Medicine, Hebei University, Baoding 071000, Hebei, PR China

ARTICLE INFO

Keywords:
Support vector machine
Fuzzy training examples
Possibility measure
Fuzzy linear separable example
Fuzzy chance constraint programming

ABSTRACT

Support vector machines (SVM) have been very successful in pattern recognition and function estimation problems, but in the support vector machines for classification, the training example is non-fuzzy input and output is $y = \pm 1$; In this paper, we introduce the support vector machine which the training examples are fuzzy input, and give some solving procedure of the Support vector machine with fuzzy training data.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

The support vector machine (SVM) is a training algorithm for learning classification and regression rules from data, SVMS were first introduced by Vapnik (1995, 1998) and Cortes and Vapnik (1995) in the 1990s for classification and have recently become an area of intense research owing to developments in the techniques and theory coupled with extensions to regression and density estimation. SVM is based on the structural risk minimization principle, this principle incorporates capacity control to prevent over-fitting and thus is a partial solution to the bias-variance trade-off dilemma.

In the support vector machine for classification, the training example is non-fuzzy input and the output is $y = \pm 1$. Considering the noisy in the training example set, fuzzy membership was introduced in classification by Chen and Chen (2002), Tsujinishi and Abe (2003), Kikuchi and Abe (2005) and Lin and Wang (2004) introduced the fuzzy support vector machine, it used the membership function to express the membership grade of an example to positive class or negative class. But in nature, it is still a common support vector machine of Vapnik.

In fact, because the noisy and error of measurement, the training examples are usually uncertain (Jeng, Chuang, & Su, 2003) or fuzzy. Then the study of support vector machine with fuzzy training data is very significant.

In this paper, we first give some preliminary knowledge, then for fuzzy training data, introduce the concept of fuzzy linear separable and approximately fuzzy linear separable. At last, we systematically study the support vector machine for two-class classification with fuzzy training data.

2. Preliminary

Here we focus on SVM for two-class classification, for the training sample $(x_1,y_1),(x_2,y_2),\ldots,(x_k,y_k) \in R^n \times \{\pm 1\}$, $y_i = +1, -1$ represent positive class and negative class respectively. The geometrical interpretation of support vector classification (SVC) is that the algorithm searches for the optimal separating hyperplane, SVM is outlined first for the linearly separable case.

The training data are linearly separable, if there exists a pair $(\mathbf{w},b)$ such that

$$\begin{aligned} &\mathbf{w}^T\mathbf{x}_i + b \geqslant 1, \ \text{for all } y_i = +1 \\ &\mathbf{w}^T\mathbf{x}_i + b \leqslant -1, \ \text{for all } y_i = -1 \end{aligned} \tag{1}$$

with the decision rule given by

$$f_{\mathbf{w},b}(\mathbf{x}) = \operatorname{sgn}(\mathbf{w}^T\mathbf{x} + b). \tag{2}$$

$\mathbf{w}$ is termed the weight vector and b is the bias (or $-b$ is termed the threshold). The inequality constraints (1) can be combined to give

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geqslant 1, \tag{3}$$

The learning problem is hence reformulated as the convex quadratic programming (QP) problem

$$\begin{aligned} &\underset{\mathbf{w},b}{\text{Minimize}}\ \Phi(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 \\ &\text{s.t. } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geqslant 1, \quad i = 1,\ldots,l. \end{aligned} \tag{4}$$

This problem has a global optimum, and its dual problem is to maximize the objective function

☆ This work was supported by a grant from National Natural Science Foundation of China (60773062) and Research Foundation of College of Medicine in Hebei University.

* Corresponding author. Address: Department of Computer Science and Engineering, Nanjing University of Aeronautics and Astronautics, 210016 Nanjing, PR China.
E-mail address: jabpjh@163.com (A.-b. Ji).

0957-4174/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2009.10.038

$$Q(\lambda) = \sum_{i=1}^{l} \lambda_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \lambda_i \lambda_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \quad (5)$$

subject to the constraints

$$\begin{cases} \sum_{i=1}^{l} \lambda_i y_i = 0 \\ \lambda_i \geqslant 0, \end{cases} \quad (6)$$

The decision function is then given by

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{l} y_i \lambda_i^* \mathbf{x}^T \mathbf{x}_i + b^* \right). \quad (7)$$

The SVM can be used to learn non-linear decision functions by first mapping the data to some higher dimensional *feature space* and constructing a separating hyperplane in this space. Denoting the mapping to feature space by

$$X \to H$$
$$\mathbf{x} \mapsto \phi(\mathbf{x})$$

$K(\mathbf{x},\mathbf{z}) \equiv \phi(\mathbf{x})^T \phi(\mathbf{z})$ is kernel function. To take account of the fact that some data points may be misclassified, we introduce a vector of slack variables $\xi = (\xi_1,\ldots,\xi_l)^T$ that measure the amount of violation of the constraints (3). The problem can then be written

$$\begin{aligned} &\underset{\mathbf{w},b,\Xi}{\text{Minimize}}\ \Phi(\mathbf{w}, b, \Xi) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l} \xi_i \\ &\text{s.t. } y_i(\mathbf{w}^T \phi(\mathbf{x}_i) + b) \geqslant 1 - \xi_i, \xi_i \geqslant 0, \quad i = 1, \ldots, l \end{aligned} \quad (8)$$

where C is specified beforehand. C is a regularization parameter that controls the trade-off between maximizing the margin and minimizing the training error term. If C is too small, then insufficient stress will be placed on fitting the training data. If C is too large, then the algorithm will over-fit the training data.

The dual problem of (8) is to maximize the objective function

$$Q(\lambda) = \sum_{i=1}^{l} \lambda_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \lambda_i \lambda_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \quad (9)$$

subject to the constraints

$$\begin{cases} \sum_{i=1}^{l} \lambda_i y_i = 0 \\ 0 \leqslant \lambda_i \leqslant C, \end{cases}$$

where C is a user-specified positive parameter. The decision functions become

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{l} y_i \lambda_i^* K(\mathbf{x}, \mathbf{x}_i) + b^* \right)$$

where the bias is given by

$$b^* = y_i - \mathbf{w}^{*T} \phi(\mathbf{x}_i) = y_i - \sum_{j=1}^{l} y_j \lambda_i^* K(\mathbf{x}_j, \mathbf{x}_i) \quad (10)$$

for any support vector $\mathbf{x}_i$.

3. Possibility measure and fuzzy chance constrained programming

Definition 3.1 (Dubois and Prade, 1988; Klir, 1999). Let X be a nonempty set, $P(X)$ be the class of all subsets of X, a mapping Pos: $P(X) \to [0,1]$ is called a possibility measure if it satisfies:

(1) $\mathrm{Pos}(\phi) = 0$
(2) $\mathrm{Pos}(X) = 1$
(3) $\mathrm{Pos}(\bigcup_{t\in T} A_t) = \underset{t\in T}{Sup}\,\mathrm{Pos}(A_t)$

Definition 3.2. Let $\tilde{a}$ be a fuzzy number, and its membership function is

$$\mu_{\tilde{a}}(x) = \begin{cases} \frac{x-r_1}{r_2-r_1}, & r_1 \leqslant x < r_2 \\ 1 & x = r_2 \\ \frac{x-r_3}{r_2-r_3}, & r_2 < x \leqslant r_3 \end{cases}$$

(where $r_1 \leqslant r_2 \leqslant r_3$, and r_1, r_2, r_3 are real numbers) $\tilde{a}$ is called a triangular fuzzy number, denoted by (r_1, r_2, r_3).

Definition 3.3 (Liu, 1998). let $\tilde{a}$ be a fuzzy number, then the possibility measure of fuzzy event $\tilde{a} \leqslant b$ is defined by $Pos(\tilde{a} \leqslant b) = Sup\{\mu_{\tilde{a}}(x) | x \in R, x \leqslant b\}$.

Similarly, $Pos(\tilde{a} < b) = Sup\{\mu_{\tilde{a}}(x).| x \in R, x < b\}, Pos(\tilde{a} = b) = \mu_{\tilde{a}}(b)$.

If $\tilde{x}_i (i = 1, 2, \ldots, n)$ are all fuzzy numbers, then $\widetilde{X} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n)$ is called a fuzzy number vector, the class of all fuzzy number vectors is denoted by $F^n(R)$, especially when $\tilde{x}_i (i = 1, 2, \ldots, n)$ are all triangular fuzzy numbers, then $\widetilde{X} = (\tilde{x}_1, \tilde{x}_2, \ldots \tilde{x}_n)$ is called a triangular fuzzy number vector, all the triangular fuzzy number vectors is denoted by $T^n(R)$.

Following from the Zadeh extension principle (Zadeh, 1978), then for function $f : R^n \to R, \tilde{y} = f(\tilde{x}_1, \tilde{x}_2, \ldots \tilde{x}_n)$ is a fuzzy number, and its membership function is:

$$\mu_{\tilde{y}}(v) = \sup_{u_1, u_2, \ldots, u_n \in R} \left\{ \min_{1 \leqslant i \leqslant n} \mu_{\tilde{x}_i}(u_i) \,|\, v = f(u_1, u_2, \ldots, u_n) \right\}$$

Especially, when $\tilde{a}, \tilde{b}$ are two fuzzy numbers, we can similarly define $\tilde{c} = f(\tilde{a}, \tilde{b})$ and we can easily obtain:

Theorem 3.1 (Liu & Liu, 2003). *Let $\tilde{a} = (r_1, r_2, r_3), \tilde{b} = (t_1, t_2, t_3)$ be two triangular fuzzy numbers, ρ be a real number, then*

(1) $\tilde{a} + \tilde{b} = (r_1 + t_1, r_2 + t_2, r_3 + t_3)$;
(2) $\rho\tilde{a} = \begin{cases} (\rho r_1, \rho r_2, \rho r_3), & \rho \geqslant 0 \\ (\rho r_3, \rho r_2, \rho r_1), & \rho < 0 \end{cases}$

Theorem 3.2. *Let $\tilde{a} = (r_1, r_2, r_3)$ be a triangular fuzzy number, then*

$$Pos\{\tilde{a} \leqslant 0\} = \begin{cases} 1 & r_2 \leqslant 0 \\ \frac{r_1}{r_1 - r_2}, & r_1 \leqslant 0, r_2 > 0 \\ 0, & r_1 > 0 \end{cases}$$

Theorem 3.3 (Liu, 2000, 2001a). *Let $\tilde{a} = (r_1, r_2, r_3)$ be a triangular fuzzy number, then for any given level $\lambda (0 < \lambda \leqslant 1), Pos\{\tilde{a} \leqslant 0\} \geqslant \lambda$ is equivalent to: $(1 - \lambda)r_1 + \lambda r_2 \leqslant 0$*

Proof. If $Pos\{\tilde{a} \leqslant 0\} \geqslant \lambda$, then either $r_2 \leqslant 0$ or $\frac{r_1}{r_1 - r_2} \geqslant \lambda$. When $r_2 \leqslant 0$, then $r_1 \leqslant r_2 \leqslant 0$, therefore $(1 - \lambda)r_1 + \lambda r_2 \leqslant 0$. When $\frac{r_1}{r_1 - r_2} \geqslant \lambda$, then $r_1 \leqslant \lambda(r_1 - r_2)$, that is: $(1 - \lambda)r_1 + \lambda r_2 \leqslant 0$.

If $(1 - \lambda)r_1 + \lambda r_2 \leqslant 0$, when $r_2 \leqslant 0$, we have $Pos\{\tilde{a} \leqslant 0\} = 1 \geqslant \lambda$; when $r_2 \geqslant 0$, then $r_1 - r_2 < 0$, by $(1 - \lambda)r_1 + \lambda r_2 \leqslant 0$, we have $\frac{r_1}{r_1 - r_2} \geqslant \lambda$, that is $Pos\{\tilde{a} \leqslant 0\} \geqslant \lambda$.

Similarly, we can prove: for a triangular fuzzy number $\tilde{a} = (r_1, r_2, r_3)$ and any given level $\lambda (0 < \lambda \leqslant 1), Pos\{\tilde{a} \geqslant 0\} \geqslant \lambda$ is equivalent to: $(1 - \lambda)r_3 + \lambda r_2 \geqslant 0$. □

4. Support vector machine for classification based on fuzzy training data

Consider the fuzzy training sample set S=$\{(\widetilde{X}_1, y_1), (\widetilde{X}_2, y_2), \ldots, (\widetilde{X}_l, y_l)\}$, where $\widetilde{X}_j \in T^n(R), y_j \in \{-1, 1\}, j = 1, 2, \ldots l$, when $y_i = 1$, *then*$(\widetilde{X}_i, y_i)$ is called a positive class; when $y_i = -1$, *then*$(\widetilde{X}_i, y_i)$ is called a negative class. The classification based on the fuzzy training set $S = \{(\widetilde{X}_1, y_1), (\widetilde{X}_2, y_2), \ldots, (\widetilde{X}_l, y_l)\}$ is to find a decision function $g(\widetilde{X})$, such that the positive class and the negative class can separated with the low classification error and good generalization performance.

4.1. Support vector machine for fuzzy linear separable training examples

Definition 4.1. For the fuzzy training sample $S = \{(\widetilde{X}_1, y_1), (\widetilde{X}_2, y_2), \ldots, (\widetilde{X}_l, y_l)\}$, if for a given level $\lambda (0 < \lambda \leqslant 1)$, there exist $w \in R^n, b \in R$, such that

$$Pos\{y_i(w \cdot \widetilde{X}_i + b) \geqslant 1\} \geqslant \lambda, \quad i = 1, 2, \ldots, l \tag{11}$$

then the fuzzy training sample set $S = \{(\widetilde{X}_1, y_1), (\widetilde{X}_2, y_2), \ldots, (\widetilde{X}_l, y_l)\}$ is called fuzzy linear separable with respect to the level λ.

Theorem 4.1. *If the fuzzy training example set* $S = \{(\widetilde{X}_1, y_1), (\widetilde{X}_2, y_2), \ldots, (\widetilde{X}_l, y_l)\}$ *is fuzzy linear separable with respect to level* λ, *where* $\widetilde{X}_i = (\tilde{x}_{i1}, \tilde{x}_{i2}, \ldots, \tilde{x}_{in})$ *and* $\tilde{x}_{ij} = (l_{ij}, m_{ij}, r_{ij})$ *is a triangular fuzzy number, then for the level* $\lambda (0 < \lambda \leqslant 1)$, *the inequality (11) is equivalent to:*

$$\begin{cases} l_{ij}(1-\lambda) + \lambda m_{ij} \leqslant t_{ij} \leqslant \lambda m_{ij} + r_{ij}(1-\lambda) \\ \quad (j = 1, 2, \ldots, n;\ i = 1, 2, \ldots, l) \\ y_i(w_1 t_{i1} + w_2 t_{i2} + \cdots + w_n t_{in} + b) \geqslant 1 \quad (i = 1, 2, \ldots, l) \end{cases} \tag{12}$$

Proof. Since

$$\begin{aligned} Pos\{y_i(w \cdot \widetilde{X}_i + b) \geqslant 1\} &= Pos\{y_i(w_1\tilde{x}_{i1} + w_2\tilde{x}_{i2} + \cdots + w_n\tilde{x}_{in} + b) \geqslant 1\} \\ &= \sup_{t_{i1}, t_{i2}, \ldots, t_{in} \in R} \left\{ \min_{1 \leqslant j \leqslant n} \mu_{\tilde{x}_{ij}}(t_{ij}) | y_i((w_1 t_{i1} + w_2 t_{i2} + \cdots + w_n t_{in} + b) \geqslant 1) \right\} \\ &\geqslant \lambda \end{aligned}$$

Therefore there exists $T_i = (t_{i1}, t_{i2}, \ldots, t_{in}) \in R^n$, such that for $1 \leqslant j \leqslant n, \mu_{\tilde{x}_{ij}}(t_{ij}) \geqslant \lambda$ and $y_i(w \cdot T_i + b) = y_i(w_1 t_{i1} + w_2 t_{i2} + \cdots + w_n t_{in} + b) \geqslant 1, \quad (i = 1, 2, \ldots, l)$. By $\mu_{\tilde{x}_{ij}}(t_{ij}) \geqslant \lambda$, we have: $l_{ij}(1-\lambda) + \lambda m_{ij} \leqslant t_{ij} \leqslant \lambda\ m_{ij} + r_{ij}(1-\lambda)$, then

$$\begin{cases} l_{ij}(1-\lambda) + \lambda m_{ij} \leqslant t_{ij} \leqslant \lambda m_{ij} + r_{ij}(1-\lambda), \\ (j = 1, 2, \ldots, n;\ i = 1, 2, \ldots, l) \\ y_i(w_1 t_{i1} + w_2 t_{i2} + \cdots + w_n t_{in} + b) \geqslant 1 \\ (i = 1, 2, \ldots, l) \end{cases}$$

The support vector machine for fuzzy linear separable training examples is to solve the fuzzy chance constrained programming (Liu, 2001b, 2002):

$$\begin{aligned} &\min \frac{1}{2}\|w\|^2 \\ &\text{s.t.} \\ &Pos\{y_i(w \cdot \widetilde{X}_i + b) \geqslant 1\} \geqslant \lambda, \quad i = 1, 2, \ldots l \end{aligned} \tag{13}$$

We can solve the fuzzy chance constrained programming using the hybrid intelligent algorithm (Liu, 1999, 2002). On the other hand, this fuzzy chance constrained programming is equivalent to the following classical convex quadratic programming (QP) problem:

$$\begin{aligned} &\min \tfrac{1}{2}\|w\|^2 \\ &\text{s.t.} \\ &\begin{cases} l_{ij}(1-\lambda) + \lambda m_{ij} \leqslant t_{ij} \leqslant \lambda m_{ij} + r_{ij}(1-\lambda), \\ \quad (j = 1, 2, \ldots, n;\ i = 1, 2, \ldots, l) \\ y_i(w \cdot T_i + b) \geqslant 1 \\ \quad (i = 1, 2, \ldots, l) \end{cases} \end{aligned} \tag{14}$$

its dual problem is to maximize the objective function

$$\begin{aligned} &Q(\alpha) = \sum_{i=1}^{l} \alpha_i - \tfrac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j T_i^T T_j \\ &\text{s.t.} \\ &\begin{cases} \sum_{i=1}^{l} \alpha_i y_i = 0 \\ \quad \alpha_i \geqslant 0, \\ l_{ij}(1-\lambda) + \lambda m_{ij} \leqslant t_{ij} \leqslant \lambda m_{ij} + r_{ij}(1-\lambda), \\ \quad (j = 1, 2, \ldots, n;\ i = 1, 2, \ldots, l) \end{cases} \end{aligned} \tag{15}$$

where $w = \sum_{i=1}^{l} \alpha_i y_i T_i, \alpha_i, T_i$ is the solution of programming (15). □

4.2. Support vector machine for approximately fuzzy linear separable training examples

To take account of the fact that some data points may be misclassified(not fuzzy linear separable with respect to the level λ), we introduce a vector of slack variables $\xi = (\xi_1, \ldots, \xi_l)^T$ that measure the amount of violation of the constraints, then the fuzzy training sample $S = \{(\widetilde{X}_1, y_1), (\widetilde{X}_2, y_2), \ldots, (\widetilde{X}_l, y_l)\}$ is called approximately fuzzy linear separable with respect to the level λ, if for a given level $\lambda (0 < \lambda \leqslant 1)$, there exist $w \in R^n, b \in R$, such that $Pos\{y_i(w \cdot \widetilde{X}_i + b) + \xi_i \geqslant 1\} \geqslant \lambda, \xi_i \geqslant 0, i = 1, 2, \ldots, l$.

To find w,b, we may solve the following fuzzy chance constrained programming

$$\begin{aligned} &\min \tfrac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} \xi_i \\ &\text{s.t.} \\ &Pos\{y_i(w \cdot \widetilde{X}_i + b) + \xi_i \geqslant 1\} \geqslant \lambda, \quad i = 1, 2, \ldots, l \\ &\xi_i \geqslant 0 \end{aligned} \tag{16}$$

Similarly we can obtain the equivalent classical convex quadratic programming (QP) problem:

Table 1
The data of diastolic pressure and plasma cholesterol of the patient of coronary and healthy people.

i	$\tilde{x}_{i1}$ (KPa)	$\tilde{x}_{i2}$ (mmol/L)	y_i	i	$\tilde{x}_{i1}$	$\tilde{x}_{i2}$	y_i
1	(9.84,9.86,9.88)	(5.17,5.18,5.19)	1	13	(10.62,10.66,10.70)	(2.06,2.07,2.08)	−1
2	(13.31,13.33,13.35)	(3.72,3.73,3.74)	1	14	(12.51,12.53,12.55)	(4.44,4.45,4.46)	−1
3	(14.63,14.66,14.69)	(3.87,3.89,3.91)	1	15	(13.30,13.33,13.36)	(3.04,3.06,3.08)	−1
4	(9.32,9.33,9.34)	(7.08,7.10,7.12)	1	16	(9.32,9.33,9.34)	(3.90,3.94,3.98)	−1
5	(12.87,12.80,12.83)	(5.47,5.49,5.51)	1	17	(10.64,10.66,10.68)	(4.43,4.45,4.47)	−1
6	(10.64,10.66,10.68)	(4.06,4.09,4.12)	1	18	(10.64,10.66,10.68)	(4.89,4.92,4.95)	−1
7	(10.65,10.66,10.67)	(4.43,4.45,4.47)	1	19	(9.31,9.33,9.35)	(3.66,3.68,3.70)	−1
8	(13.31,13.33,13.35)	(3.60,3.63,3.66)	1	20	(10.64,10.66,10.68)	(3.20,3.21,3.22)	−1
9	(13.32,13.33,13.34)	(5.68,5.70,5.72)	1	21	(10.37,10.40,10.43)	(3.92,3.94,3.96)	−1
10	(11.97,12.00,12.03)	(6.17,6.19,6.21)	1	22	(9.31,9.33,9.35)	(4.90,4.92,4.94)	−1
11	(14.64,14.66,14.68)	(4.00,4.01,4.02)	1	23	(11.19,11.20,11.21)	(3.40,3.42,3.44)	−1
12	(13.31,13.33,13.35)	(3.99,4.01,4.03)	1	24	(9.31,9.33,9.35)	(3.62,3.63,3.64)	−1

$$\min \tfrac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i$$
s.t.
$$\begin{cases} l_{ij}(1-\lambda)+\lambda m_{ij} \leqslant t_{ij} \leqslant \lambda m_{ij} + r_{ij}(1-\lambda), \\ \quad (j=1,2,\ldots,n;\ i=1,2,\ldots,l) \\ y_i(w\cdot T_i + b) + \xi_i \geqslant 1 \\ \quad (i=1,2,\ldots,l) \\ \xi_i \geqslant 0, \\ \quad (i=1,2,\ldots,l) \end{cases} \tag{17}$$

(where C is a given parameter which control the trade-off of complexity of machine and the number of non-separable points).

The dual problem is to maximize the objective function

$$Q(\alpha) = \sum_{i=1}^{l}\alpha_i - \tfrac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j T_i^T T_j$$
s.t.
$$\begin{cases} \sum_{i=1}^{l}\alpha_i y_i = 0 \\ 0 \leqslant \alpha_i \leqslant C, \\ l_{ij}(1-\lambda)+\lambda m_{ij} \leqslant t_{ij} \leqslant \lambda m_{ij} + r_{ij}(1-\lambda), \\ \quad (j=1,2,\ldots,n;\ i=1,2,\ldots,l) \end{cases} \tag{18}$$

For a fuzzy example which unknown class $\widetilde{X} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n)$, the decision rule is: for a given confidence level $\lambda(0 < \lambda \leqslant 1)$, If $Pos\{(w_0 \cdot \widetilde{X} + b) \geqslant 0\} \geqslant \lambda$, then $\widetilde{X} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n)$ is a positive example; If $Pos\{(w_0 \cdot \widetilde{X} + b) \leqslant 0\} \geqslant \lambda$, then $\widetilde{X} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n)$ is a negative example.

5. Example

In the following, we shall apply the support vector machine for two-class classification with fuzzy training data in the diagnosis of Coronary. The data in Table 1 are the diastolic pressure $(\tilde{x}_{i1})$ and plasma cholesterol $(\tilde{x}_{i2})$ of 24 persons, where half of them are healthy ($y_i = 1$), the others are Coronary patients($y_i = -1$), $\tilde{x}_{i1}$ and $\tilde{x}_{i2}$ are triangular fuzzy numbers.

Based on the fuzzy training data given in Table 1, when parameter C=0.1, $\lambda = 0.65$, solving the programming (16), we can obtain w_0 = (0.415444,0.4792959),b = -0.6962587, then the decision rule is: for a given confidence level $\lambda = 0.65$, If $Pos\{(w_0 \cdot \widetilde{X} + b) \geqslant 0\} \geqslant 0.65$, then $\widetilde{X} = (\tilde{x}_1, \tilde{x}_2)$ is a positive example; If $Pos\{(w_0 \cdot \widetilde{X} + b) \leqslant 0\} \geqslant 0.65$, then $\widetilde{X} = (\tilde{x}_1, \tilde{x}_2)$ is a negative example. Using this decision rule to fit the data in Table1, only three fuzzy examples are misclassification.

6. Conclusions

This paper discusses the support vector machine which the training example is fuzzy input, and give some solution procedure of the Support vector machine with fuzzy training data, it is a generalization of support vector machine of V.N. Vapnik. In the further study, we are to discuss the support vector regression machine based on the fuzzy input and fuzzy output.

References

Chen, J.-H., & Chen, C.-S. (2002). Fuzzy kernel perceptron. *IEEE Transactions Neural Networks, 13*(6), 1364–1373.

Cortes, C., & Vapnik, V. N. (1995). Support vector networks. *Machine Learning, 20*(3), 273–297.

Dubois, D., & Prade, H. (1988). *Possibility theory*. New York: Plenum Press.

Jeng, J.-T., Chuang, C.-C., & Su, S.-F. (2003). Support vector interval regression networks for interval regression analysis. *Fuzzy Sets and Systems, 138*, 283–300.

Kikuchi, T., & Abe, S. (2005). Comparison between error correcting output codes and fuzzy support vector machines. *Pattern Recognition Letters, 26*, 1937–1945.

Klir, G. J. (1999). On fuzzy-set interpretation of possibility theory. *Fuzzy Sets andSystems, 108*, 263–373.

Lin, C., & Wang, S. (2004). Training algorithms for fuzzy support vector machines with noisy data. *Pattern Recognition Letters, 25*, 1647–1656.

Liu, B. (1998). Minimax chance constrained programming models for fuzzy decision systems. *Information Sciences, 112*, 25–38.

Liu, B. (1999). *Uncertain programming*. New York: Wiley.

Liu, B. (2000). Dependent-chance programming in fuzzy environments. *Fuzzy Sets and Systems, 109*(1), 97–106.

Liu, B. (2001a). Fuzzy random chance-constrained programming. *IEEE Transactions on Fuzzy Systems, 9*(5), 713–720.

Liu, B. (2001b). Fuzzy random dependent-chance programming. *IEEE Transactions on Fuzzy Systems, 9*(5), 721–726.

Liu, B. (2002). *Theory and practice of uncertain programming*. Heidelberg: Physica.

Liu, Y. K., & Liu, B. (2003). Fuzzy random variable: A scalar expected value operator. *Fuzzy Optimization and Decision Making, 2*(2), 143–160.

Tsujinishi, D., & Abe, S. (2003). Fuzzy least square SVM for multiclass problems. *Journal of Neural Networks, 16*, 785–792.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. New York: Springer-Verlag.

Vapnik, V. N. (1998). *Statistical learning theory*. New York: Wiley.

Zadeh, L. A. (1978). Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems, 1*, 3–28.