Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition Letters 29 (2008) 1175–1181
Pattern Recognition Letters
www.elsevier.com/locate/patrec

Fuzzy relevance vector machine for learning from unbalanced data and noise ☆

Ding-Fang Li *, Wen-Chao Hu, Wei Xiong, Jin-Bo Yang

School of Mathematics and Statistics, Wuhan University, Wuhan, 430072 Hubei, PR China

Received 20 June 2006; received in revised form 24 September 2007
Available online 26 January 2008

Communicated by A.M. Alimi

Abstract

Handing unbalanced data and noise are two important issues in the field of machine learning. This paper proposed a complete framework of fuzzy relevance vector machine by weighting the punishment terms of error in Bayesian inference process of relevance vector machine (RVM). Above problems can be learned within this framework with different kinds of fuzzy membership functions. Experiments on both synthetic data and real world data demonstrate that fuzzy relevance vector machine (FRVM) is effective in dealing with unbalanced data and reducing the effects of noises or outliers.

Keywords: Relevance vector machine; Unbalanced data; Noise; Fuzzy membership; Bayesian inference

1. Introduction

Relevance vector machine is a popular learning machine motivated by the statistical learning theory, and gaining popularity because of theoretically attractive features and profound empirical performance (Tipping, 2001a,b; Majumder et al., 2005; Bishop and Tipping, 2000). However, there are still some limitations of this theory. During the training procedure of RVM, all training points are treated uniformly, as a matter of fact, in many real world applications, the influence of the training points are different.

There are many researches which are focused on the following two major issues: learning from unbalanced data and noise (Murphey et al., 2004; Guo and Murphey, 2001; Tao et al., 2005; Fu Lin and Wang, 2005; Lin and Wang, 2002, 2004). In many application problems, the training data for each class is extremely unbalanced. To classify potential customers in ecommerce is a case in point. One thing in common in ecommerce is that 99% of netizen do not buy any product but only 1% buy some product. Most machine learning algorithms may not be robust enough and sometimes their performance could be affected severely with unbalanced data. This issue is caused by the overwhelming number of learning samples in one class input to the learning system partially undo the training effect on the small learning samples of a different class. The problem is more serious when data set has high level of noise.

In order to deal with above problems in the area of machine learning, Lin and Wang propose fuzzy support vector machine (FSVM) to eliminate the influence caused by unbalanced data and noise (Fu Lin and Wang, 2005; Lin and Wang, 2002, 2004). In this paper, a complete framework of FRVM is presented to address above problems with respect to RVM. By introducing Fuzzy mathematics, RVM is reformulated into FRVM. Specifically, a fuzzy membership is assigned to each input point such that different input points can make different influences in learning

☆ Supported by the National Natural Science Foundation of China (70771708).
* Corresponding author. Tel.: +86 027 687752957; fax: +86 027 68773568.
E-mail addresses: dfli@whu.edu.cn (D.-F. Li), wchu80@sina.com (W.-C. Hu), wxiongwhu@163.com (W. Xiong), yangjb1225@163.com (J.-B. Yang).

0167-8655/$ - see front matter © 2008 Published by Elsevier B.V.
doi:10.1016/j.patrec.2008.01.009

process. This is a natural way to make the learning algorithm more robust against unbalanced data and noise. Compared with FSVM, FRVM is based on full probabilistic framework rather than optimization theory.

The rest of this article is organized as follows. A brief review of relevance vector machine will be described in Section 2. Section 3 gives details on the architectures of fuzzy relevance vector machine. Different kinds of fuzzy membership functions are introduced in Section 4. The performance of the fuzzy relevance vector machine is presented and compared with the conventional RVM in Section 5. Some concluding remarks are included in Section 6.

2. Relevance vector machine

RVM is a probabilistic non-linear model with a prior distribution on the weights that enforces sparse solutions (Tipping, 2001a). It is reported that RVM can yield nearly identical performance to, if not better than, that of SVM while using far fewer relevance vectors than the number of support vectors for SVM in several benchmark studies (Tipping, 2001a,b; Majumder et al., 2005; Bishop and Tipping, 2000). Compared with SVM, it is not necessary for RVM to tune any regularization parameter during the training phase, neither for kernel function to satisfy Mercer's condition. Furthermore, the predictions are probabilistic. For regression problems, the RVM makes predictions based on the function:

$$y(x,\omega)=\sum_{i=1}^{N}\omega_i K(x,x_i)+\omega_0 \tag{1}$$

where $K(x,x_i)$ is a kernel function, which effectively defining one basis function for each example in the training set, and $\omega=(\omega_0,\omega_1,\ldots,\omega_N)^{\mathrm{T}}$ are adjustable parameters (or weights). Inferring weights procedures is under a fully probabilistic framework. Specifically, a Gaussian prior distribution of zero mean and variance $\sigma_{\omega_j}^2\equiv\alpha_j^{-1}$ is defined over each weight:

$$p(\omega|\alpha)=\prod_{i=0}^{N}N(\omega_i|0,\alpha_i^{-1}) \tag{2}$$

where the key to obtain sparsity is the use of $N+1$ independent hyperparameters $\alpha=(\alpha_0,\alpha_1,\ldots,\alpha_N)^{\mathrm{T}}$, one per weight (or basis function), which moderate the strength of the prior information.

Given a data set of input-target pairs $G=\{(x_i,t_i)\}_{i=1}^{N}$ (where x_i is the input vector, t_i is the desired real-valued labeling, and N is the number of the input records). Suppose the targets are independent and noise is assumed to be mean-zeros Gaussian with variance σ^2. Thus, the likelihood of the complete data set can be written as

$$p(t|\omega,\sigma^2)=(2\pi\sigma^2)^{-N/2}\exp\left\{-\frac{1}{2\sigma^2}\|t-\Phi\omega\|^2\right\} \tag{3}$$

where $t=(t_1,t_2,\ldots,t_N)^{\mathrm{T}}$, $\Phi=[\phi(x_1),\phi(x_2),\ldots,\phi(x_N)]^{\mathrm{T}}$ and $\phi(x_n)=[1,K(x_n,x_1),K(x_n,x_2),\ldots,K(x_n,x_N)]^{\mathrm{T}}$.

Having defined the prior distribution and likelihood function, from Bayes' rule, the posterior over weights is thus given by

$$p(\omega|t,\alpha,\sigma^2)=\frac{p(t|\omega,\sigma^2)p(\omega|\alpha)}{p(t|\alpha,\sigma^2)}\sim N(\omega|\mu,\Sigma) \tag{4}$$

where the posterior covariance and mean are, respectively,

$$\Sigma=(\sigma^{-2}\Phi^{\mathrm{T}}\Phi+A)^{-1} \tag{5}$$

$$\mu=\sigma^{-2}\Sigma\Phi^{\mathrm{T}}t \tag{6}$$

with $A=\mathrm{diag}(\alpha)=\mathrm{diag}(\alpha_0,\alpha_1,\ldots,\alpha_N)$.

The likelihood distribution over the training targets can be "marginalized" by integrating out the weights to obtain the marginal likelihood for the hyperparameters:

$$p(t|\alpha,\sigma^2)=\int p(t|\omega,\sigma^2)p(\omega|\alpha)\mathrm{d}\omega\sim N(0,C) \tag{7}$$

where the covariance is given by $C=\sigma^2 I+\Phi A^{-1}\Phi^{\mathrm{T}}$.

The estimated value of the model weights is given by the mean of the posterior distribution, which is also the maximum a posteriori (MAP) estimate of the weights, and depends on the value of the hyperparameters α and of the noise σ^2 whose estimated value is obtained by maximizing (7).

Given a new input x_*, the probability distribution of the corresponding output y_* is given by the (Gaussian) predictive distribution:

$$p(t_*|x_*,\alpha_{\mathrm{MP}},\sigma_{\mathrm{MP}}^2)=\int p(t_*|x_*,\omega,\sigma_{\mathrm{MP}}^2)p(\omega|t,\alpha_{\mathrm{MP}},\sigma_{\mathrm{MP}}^2)\mathrm{d}\omega\sim N(y_*,\sigma_*^2) \tag{8}$$

where the mean and the variance (uncertainty) of the prediction are, respectively,

$$y_*=\mu^{\mathrm{T}}\phi(x_*), \tag{9}$$

$$\sigma_*^2=\sigma_{\mathrm{MP}}^2+\phi(x_*)^{\mathrm{T}}\Sigma\phi(x_*). \tag{10}$$

The RVM is built on the few training samples whose associated hyperparameters do not go to infinity during the training process, leading to a sparse solution. These remaining samples are called the relevance vectors (RVs), resembling the SVs in the SVM framework. We give the pseudo-code of the RVM algorithm in Algorithm 1.

Relevance vector classification follows an essentially identical framework as for regression, for simplicity we omit details here:

Input: $S=\{(x_i,t_i)\}_{i=1}^{N}$: training data set; N: the number of the independent samples; ε_n: additive noise assumed to be mean-zero Gaussian with variance σ^2.

Output: $S'\subseteq S$: relevance vectors; $y(x,\omega)$: predicted function.

Termination conditions: training samples $S=\{(x_i,t_i)\}_{i=1}^{N}$ are all trained.

Algorithm 1 (*RVM*).

Begin
hyperparameters α and σ^2 are obtained according to maximizing Eq. (7) the marginal likelihood for hyperparameters α and σ^2;
model weights are given by mean of the posterior distribution Eq. (4); **for** $i = 1$ to N
{ **if** $\omega_i \neq 0$
{The corresponding point (x_i, t_i) is a relevance vector;}
$i = i + 1$;
}
predicted function $y(x, \omega)$ is calculated according to Eq. (1);
End

3. Fuzzy relevance vector machine

It has been emphasized above that the performance of RVM relies on only a few training points represent "prototypical" samples of classes, except them, most training points could be considered of no use or even harmful. Thus, to further improve the performance of RVM model, different training points should be treated with various attitudes, i.e. a fuzzy membership s_i associated with each training point x_i can be used as the attitude of meaningful. We extend the concept of RVM with fuzzy membership and named it as FRVM.

3.1. For regression problems

The final model obtained by RVM can be regarded as minimize:

$$L(\omega) = \frac{1}{2\sigma^2}\sum_{n=1}^{N}[t_n - \omega^{\mathrm{T}}\phi(x_n)]^2 + \sum_{i=0}^{N}\log|\omega_i| \tag{11}$$

where the first term of Eq. (11) is a measure of sum error in the RVM, and the second term of Eq. (11) is the regularization term. When introducing the fuzzy membership s_i of the corresponding point x_i, Eq. (11) can be changed as follows:

$$L(\omega) = \frac{1}{2\sigma^2}\sum_{n=1}^{N}[s_n(t_n - \omega^{\mathrm{T}}\phi(x_n))]^2 + \sum_{i=0}^{N}\log|\omega_i| \tag{12}$$

where the left term of Eq. (12) can be regarded as weighted sum error in the RVM.

Considering the difference between Eqs. (11) and (12) in the Bayesian inference, it is necessary to reformulate Eqs. (3), (5) and (6) in classical RVM. Eq. (3) is redefined as follows:

$$p(t|\omega, \sigma^2) = (2\pi\sigma^2)^{-N/2}\exp\left\{-\frac{1}{2\sigma^2}\sum_{n=1}^{N}[s_n(t_n - \omega^{\mathrm{T}}\phi(x_n))]^2\right\} \tag{13}$$

Eqs. (5) and (6) are altered based on Eq. (13), gives:

$$\Sigma = (\sigma^{-2}\Phi^{\mathrm{T}}G^2\Phi + A)^{-1} \tag{14}$$

with $G = \mathrm{diag}(s_1, s_2, \ldots, s_N)$:

$$\mu = \sigma^{-2}\Sigma\Phi^{\mathrm{T}}Gt. \tag{15}$$

Eq. (7) for marginal likelihood is replaced by

$$p(t|\alpha, \sigma^2) = \int p(t|\omega, \sigma^2)p(\omega|\alpha)\mathrm{d}\omega \sim N(0, C), \tag{16}$$

where the covariance is given by $C = \sigma^2 I + G\Phi A^{-1}\Phi^{\mathrm{T}}G^{\mathrm{T}}$.

The updating procedure for α is the same as classical RVM, i.e. does not need to change:

$$\alpha_i^{\mathrm{new}} = \frac{\gamma_i}{\mu_i^2}, \tag{17}$$

where μ_i is the ith posterior mean weight from (15) and γ_i is defined by

$$\gamma_i = 1 - \alpha_i\Sigma_{ii} \tag{18}$$

With Σ_{ii} the ith diagonal element of the posterior weight covariance from (16) computed with the current α and σ^2 values. For the noise variance σ^2, the update is redefined as follows:

$$(\sigma^2)^{\mathrm{new}} = \|Gt - G\Phi\mu\|^2/(N - \Sigma_i\gamma_i) \tag{19}$$

3.2. For classification problems

For two-class classification, applying the generalization linear model and logistic sigmoid link function, the likelihood can be written as follows:

$$p(t|\omega) = \prod_{n=1}^{N}\sigma\{y(x_n;\omega)\}^{t_n}[1 - \sigma\{y(x_n;\omega)\}]^{1-t_n} \tag{20}$$

where the targets $t_n \in \{0,1\}$.

The most probable weights ω_{MP} can be obtained by finding the minimum over of

$$-\log\{p(t|\omega)p(\omega|\alpha)\} = -\sum_{n=1}^{N}[t_n\log y_n + (1 - t_n) \times \log(1 - y_n)] + \frac{1}{2}\omega^{\mathrm{T}}A\omega \tag{21}$$

with $y_n = \sigma\{y(x_n;\omega)\}$. Where the first term of Eq. (21) is the sum error of data, and the second term of Eq. (21) is the regularization term. When introducing the fuzzy membership of the corresponding point, Eq. (21) can be changed as follows:

$$-\log\{p(t|\omega)p(\omega|\alpha)\} = -\sum_{n=1}^{N}s_n[t_n\log y_n + (1 - t_n) \times \log(1 - y_n)] + \frac{1}{2}\omega^{\mathrm{T}}A\omega \tag{22}$$

The following procedure adopt the efficient "iteratively-reweighed least-squares" algorithm to find ω_{MP}.

The gradient and Hessian matrix of (22) are given by

$$g = \Phi^{\mathrm{T}} \operatorname{diag}(s)(y - t) + A\omega \tag{23}$$

$$\text{Hessian} = \Phi^{\mathrm{T}} \operatorname{diag}(s) B \Phi + A \tag{24}$$

where $B = \operatorname{diag}(\beta_1, \beta_2, \ldots, \beta_N)$ is a diagonal matrix with $\beta_n = \sigma\{y(x_n)\}[1 - \sigma\{y(x_n)\}]$. The posterior covariance is thus given by

$$\Sigma = (\Phi^{\mathrm{T}} \operatorname{diag}(s) B \Phi + A)^{-1} \tag{25}$$

Using the statistics Σ and ω_{MP}, the hyperparameters α are updated with (17) in identical fashion to the regression case.

From the above, we can easily conclude that it is standard RVM if we set all $s_i = 1$. With different value of s_i, we can control the trade-off of the respective training point (x_i, t_i) in the system. A smaller value of s_i makes the corresponding point (x_i, t_i) less important in the training. So RVM is the special case of FRVM if we set all $s_i = 1$ and FRVM is more suitable for classify and regression.

4. Generating the fuzzy memberships

There are two typical examples for unbalanced data: data with time property or data with classes that have very unequal frequency. In many real applications, some samples may be outliers or be corrupted by noise, Since the RVM depends on only a small part of the data points (RVs), it may become sensitive to noises or outliers in the training set. In this paper, fuzzy memberships are utilized to make RVM more robust under above circumstance. Specifically, different fuzzy membership functions are defined.

4.1. Data with time property

Given a sequence of training points

$$\{(x_1, y_1, t_1), \ldots, (x_N, y_N, t_N)\} \tag{26}$$

where $t_1 \leqslant \cdots \leqslant t_N$ is the time when the point arrived in the system. Let fuzzy membership s_i be a function of time t_i

$$s_i = f(t_i) \tag{27}$$

such that $s_1 = \mu \leqslant \cdots \leqslant s_N = 1$.

A quadric function of time is chosed to approximate fuzzy membership function

$$s_i = f(t_i) = a(t_i - b)^2 + c \tag{28}$$

by applying the boundary conditions, the following shows

$$s_i = f(t_i) = (1 - \mu)\{(t_i - t_1)/(t_N - t_1)\}^2 + \mu \tag{29}$$

4.2. Classes with very unequal frequency

In some classification problem, cost might vary, i.e. cost matrices are constructed for different classes. The prototypical example of the problem of cost-sensitive classification is medical diagnosis which reflect 90% healthy and 10% disease. Classification error rate or accuracy is not the best measure here. In this situation, let fuzzy membership s_i be a function of class y_i

$$s_i = \begin{cases} s_+, y_i = 1 \\ s_-, y_i = -1 \end{cases} \tag{30}$$

4.3. Outliers and noises

Lin and Wang have proposed two strategies for automatic setting of fuzzy memberships in train support vector machine with noise data (Lin and Wang, 2004). Suppose a heuristic function $h(x)$ is highly relevant to the fuzzy membership function $\mu(x)$. The relationship between functions $h(x)$ and $\mu(x)$ is defined as

$$\mu(x) = \begin{cases} 1, \\ \quad \text{if } h(x) > h_c \\ \sigma + (1 - \sigma)\{(h(x) - h_T)/(h_C - h_T)\}^d, \\ \quad \text{if } h(x) < h_T \\ \sigma, \\ \quad \text{otherwise} \end{cases} \tag{31}$$

In the context of discriminating between noises and data, there are two strategies to define the heuristic function $h(x)$. One is based on kernel-target alignment (Cristianini et al., 2002) and the other is k-NN.

The strategy of kernel-target alignment (KT) uses the function $f_K(x_i, y_i)$ as the heuristic function $h(x_i)$, i.e.

$$h(x_i) = f_K(x_i, y_i) = \sum_{j=1}^{N} y_i y_j K(x_i, x_j) \tag{32}$$

where $K(x_i, x_j) = \exp(-r^{-2}\|x_i - x_j\|^2)$. The strategy of k-NN (k-NN) defines the heuristic function in the following procedure:

- Find a set S_i^K that consists of k nearest neighbors of x_i.
- Count the number of data points in the set S_i^K that the class label is the same as that of the data point x_i, and represents it as n_i.
- Define $h(x_i) = n_i$.

Results presented in (Lin and Wang, 2004) indicate that both two strategies slightly improve the performance of FSVM, meanwhile, the time complexity of FSVM increases due to the need to estimate many extra parameters. To make a trade-off between predicting performance and complexity, a simple way to set the fuzzy membership could be used as an alternative.

Denote the mean of class +1 as x_+ and the mean of class −1 as x_-. The radius of class +1 is

$$r_+ = \max_{\{x_i | y_i = 1\}} \|x_+ - x_i\| \tag{33}$$

and the radius of class −1 is

$$r_- = \max_{\{x_i | y_i = -1\}} \|x_- - x_i\| \tag{34}$$

Then fuzzy membership s_i of x_i in class +1 is defined as follows

$$s_i = \begin{cases} 1, & \\ \quad \|x_i - x_+\|/r_+ \leqslant \|x_i - x_-\|/r_- & \\ \{(\|x_i - x_-\|/r_-)/(\|x_i - x_+\|/r_+)\}^q, & \\ \quad \text{else} & \end{cases} \tag{35}$$

and the fuzzy membership s_i of x_i in class −1 is similar to class +1

$$s_i = \begin{cases} 1, & \\ \quad \|x_i - x_-\|/r_- \leqslant \|x_i - x_+\|/r_+ & \\ \{(\|x_i - x_+\|/r_+)/(\|x_i - x_-\|/r_-)\}^q, & \\ \quad \text{else} & \end{cases} \tag{36}$$

5. Experiments

We have described the basis idea, method and process of FRVM framework in detail. By weighting the punishment terms of error in Bayesian inference process of RVM, problems such as training with unbalanced data or noisy data can be solved with FRVM. Now, to validate the effectiveness of FRVM framework, a synthetic dataset and eight real datasets are selected to evaluate the performance of FRVM as follows.

5.1. Experiment with unbalanced data

Ripley's synthetic data[1] is used to evaluate the FRVM with unbalanced data. This data consists of two features and one targeted variable.

Table 1 gives the results of simulations with unbalanced data. Both two typical examples for unbalanced data are given. As shown in Table 1, we can see the following results:

(1) In the training phase, the training error of FRVM is worse than that of RVM. That is because the principle disadvantage of RVM is the complexity of the training phase, as it is necessary to repeatedly compute and invert the Hessian matrix, requiring $O(N^2)$ storage and $O(N^3)$ computation. And this disadvantage is strengthened in the case of FRVM. A little rounding computation error because of the compute may lead to a big error in the result, at the same time the data is unbalanced, so in the training phase the training error of FRVM is worse than that of RVM.
(2) Test errors and unbalanced errors (errors of later data for data with time property and errors of class +1 for data with classed that have very unequal frequency) of FRVM are consistently smaller than that of RVM. It is natural because in FRVM theory different train data has different contribution to the final FRVM model and RVM is the special case of FRVM if we set all $s_i = 1$, so FRVM is more suitable for classify and regression.
(3) FRVM uses more relevance vectors than RVM. It is noted that a smaller s_i reduces the effect the parameter ω_i in problem (13) such that the corresponding point (x_i, t_i) is treated as less important. An important

Table 1
Comparison between the RVM method and the FRVM method in models constructed with unbalanced data about training errors, test errors, unbalanced errors and relevance vectors

	Data with time property		Classes with very unequal frequency	
	RVM	FRVM	RVM	FRVM
Training errors (%)	6.40	6.40	6.40	7.60
Relevance vectors	4	5	4	6
Test errors (%)	9.60	9.30	9.60	8.50
Unbalanced errors (%)	4.40	4.20	5.30	3.40

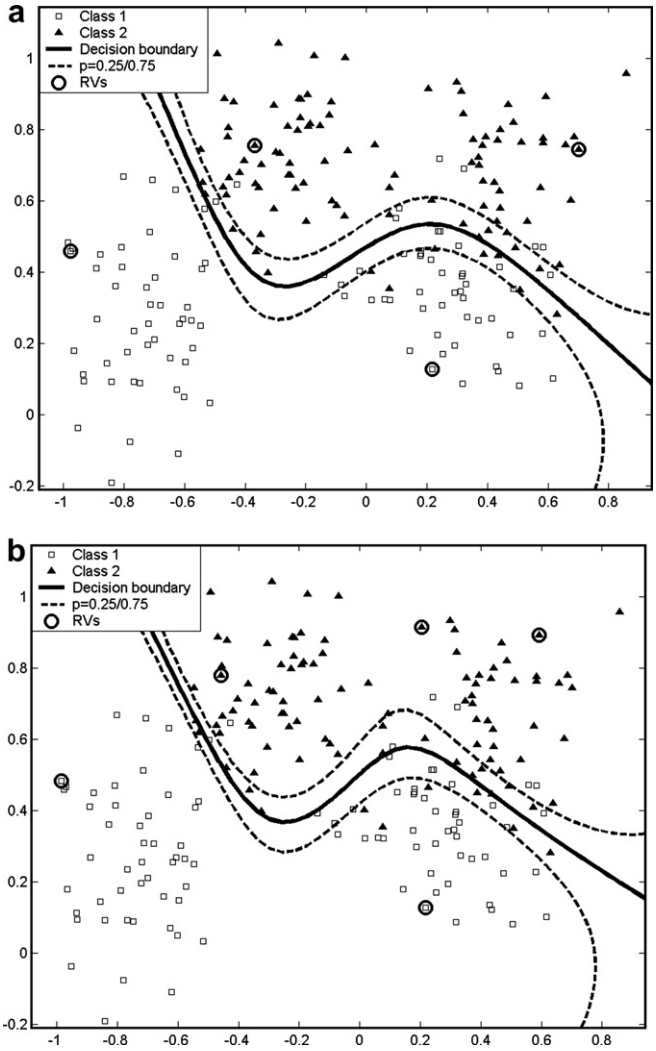

Fig. 1. Comparison of the visual results for data with time property between RVM method and SVM method.

[1] http://www.stats.ox.ac.uk/pub/PRNN/.

difference between RVM and FRVM is that the points with the same value of ω_i may indicate a different type of relevance vectors in FRVM due to the factor s_i, that is to say there are two irrelevance vectors (x_i, t_i) and (x_k, t_k) with the same parameter ω_i in RVM, whose fuzzy degree, respectively, is s_i and s_k in FRVM. If the value of s_i is much bigger than the value of s_k, the corresponding point (x_i, t_i) becomes much important in the training, and (x_i, t_i) becomes relevance vector. So FRVM uses more relevance vectors than RVM.

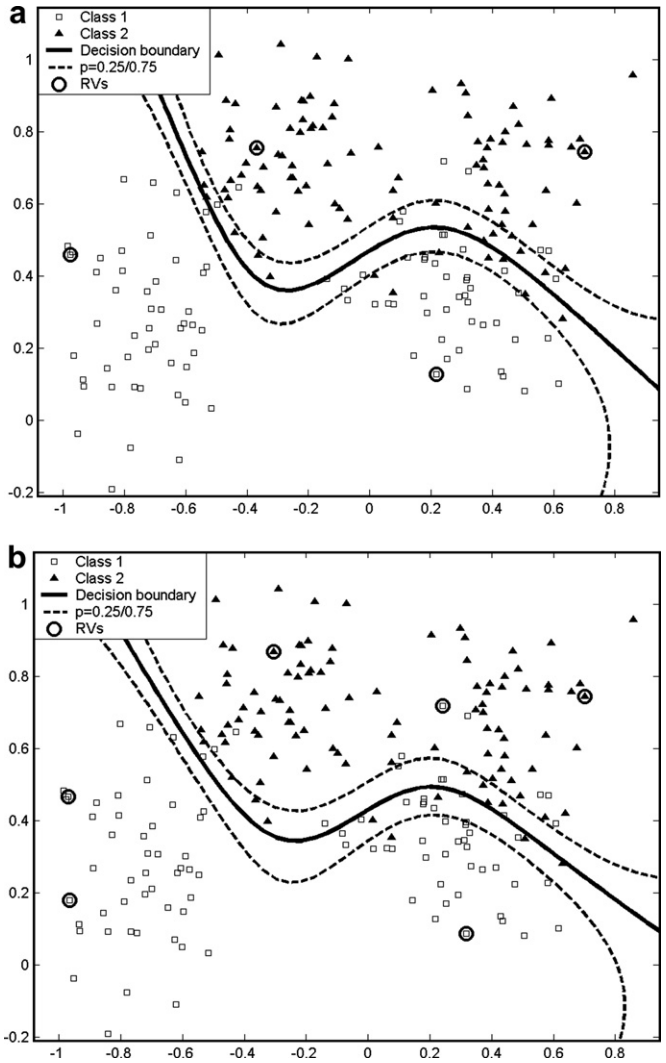

Fig. 2. Comparison of the visual results for classes with very unequal frequency between RVM method and FRVM method.

Table 2
Comparison between the RVM method and the FRVM method in models constructed with noise data about test errors

Data sets	Test errors (%)	
	RVM	FRVM
Titanic	22.79 ± 0.34	22.54 ± 0.42
Breast-cancer	28.18 ± 4.89	27.14 ± 5.52
Banana	10.94 ± 0.48	10.67 ± 0.40
Thyroid	3.67 ± 1.86	3.17 ± 1.88
Diabetis	24.80 ± 2.49	24.17 ± 2.47
Heart	19.8 ± 4.38	18.1 ± 3.48
Waveform	10.91 ± 0.36	10.61 ± 0.56
Twonorm	3.51 ± 0.40	3.36 ± 0.42

Fig. 1a and b are the visual results for data with time property by using RVM and FRVM, respectively. Fig. 2a and b are the visual results for data with classes that have very unequal frequency by using RVM and FRVM, respectively.

5.2. Experiment with noisy data

Eight datasets[2] are used to investigate the performance of FRVM with noise, a total of 100 training/test splits are provided by the authors of these datasets, our results show average over the 10th, 20th, 100th of those. All parameters are estimated through 10-fold cross-validation.

Table 2 presents the results of simulations with noise which show that FRVM can improve the performance of RVM when the data contains noisy data.

From above results, we can easily conclude that FRVM can cope with unbalanced data and noisy data better than RVM.

6. Conclusion

Pattern recognition is a well-studied problem in machine learning. Various techniques such as decision trees, neural networks, and rule induction, have been developed and successfully applied to many domains. Many of these standard pattern recognition algorithms usually assume that training samples are evenly distributed among different classes and without corrupted by noise. However, unbalanced data sets or noisy data sets appear frequently in real world machine learning problems, and increase difficulties in the training phase.

This paper presents a class of fuzzy relevance vector machine that combines the fuzzy mathematics with RVM. Fuzzy relevance vector machine imposes a fuzzy membership to each input point such that different input points can make different contributions to the Bayesian learning process. By setting different types of fuzzy memberships, problems such as training with unbalanced data or noisy data can be solved with a extension towards RVM. Experiments on both synthesis data and real world data have demonstrated that the proposed fuzzy relevance vector machine is reasonable and its performance is more robust than that of regular RVM.

There still remains some directions for future work. One is to "dig deep" the machine learning problems, then more

[2] These datasets are available at http://ida.first.fraunhofer.de/projects/bench/.

suitable model of fuzzy membership function can be built.

Acknowledgements

The authors would like to thank Chun-Fu Lin of National Taiwan University for providing us with FSVM code.

References

Bishop, C.M., Tipping, Michael E., 2000. Variational relevance vector machine. In: Proc. 16th Conf. on Uncertainty in Artificial Intelligence. Morgan Kaufman Publishers.

Cristianini, N., Shawe-Taylor, J., Elisseeff, A., Kandola, J., 2002. On kernel-target alignment. In: Advances in Neural Information Processing Systems, vol. 14. MIT Press, pp. 367–373.

Fu Lin, J., Wang, Sheng De, 2005. Fuzzy support vector machines with automatic membership setting. Support Vector Machines: Theory Appl. Studies Fuzziness Soft Comput., 233–254.

Guo, Hong, Murphey, Yi L., 2001. Neural learning from unbalanced data using noise modeling. Lecture Notes Comput. Sci. 2070, 259.

Lin, Chun-Fu, Wang, Sheng-De, 2002. Fuzzy support vector machines. IEEE Trans. Neural Networks 13 (2), 464–471.

Lin, Chun-Fu, Wang, Sheng-De, 2004. Training algorithms for fuzzy support vector machines with noisy data. Pattern Recognition Lett. 25 (14), 1647–1656.

Majumder, S.K., Ghosh, N., Gupta, P.K., 2005. Relevance vector machine for optical diagnosis of cancer. Lasers Surg. Med. 36 (4), 323–333.

Murphey, Yi L., Guo, Hong, Feldkamp, Lee A., 2004. Neural learning from unbalanced data. Appl. Intell. 21 (2), 117–128.

Tao, Qing, Wu, Gao-Wei, Wang, Fei-Yue, Wang, Jue, 2005. Posterior probability support vector machines for unbalanced data. IEEE Trans. Neural Networks 16 (6), 1561–1573.

Tipping, M.E., 2001a. Sparse Bayesian learning and the relevance vector machine. J. Mach. Learn. Res. 1, 211–244.

Tipping, M.E., 2001b. The relevance vector machine. In: Advances in Neural Information Processing Systems, vol. 12. The MIT Press, Cambridge, MA.