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.