Pattern Recognition 48 (2015) 2072–2083

Contents lists available at ScienceDirect

Pattern Recognition

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

Improved support vector machine algorithm for heterogeneous data

Shili Peng [a,b], Qinghua Hu [a,c,*], Yinli Chen [b], Jianwu Dang [a,c]

[a] School of Computer Science and Technology, Tianjin University, Tianjin, China
[b] Department of Computer Science and Technology, Guangdong University of Finance, Guangzhou, China
[c] Tianjin key Laboratory of Cognitive Computing, Tianjin, China

ARTICLE INFO

Article history:
Received 17 October 2013
Received in revised form
2 November 2014
Accepted 17 December 2014
Available online 24 December 2014

Keywords:
Support vector machine
Heterogeneous data
Nominal attribute
Numerical attribute
Classification learning

ABSTRACT

A support vector machine (SVM) is a popular algorithm for classification learning. The classical SVM effectively manages classification tasks defined by means of numerical attributes. However, both numerical and nominal attributes are used in practical tasks and the classical SVM does not fully consider the difference between them. Nominal attributes are usually regarded as numerical after coding. This may deteriorate the performance of learning algorithms. In this study, we propose a novel SVM algorithm for learning with heterogeneous data, known as a heterogeneous SVM (HSVM). The proposed algorithm learns an mapping to embed nominal attributes into a real space by minimizing an estimated generalization error, instead of by direct coding. Extensive experiments are conducted, and some interesting results are obtained. The experiments show that HSVM improves classification performance for both nominal and heterogeneous data.

© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

In the last decade, the support vector machine (SVM) classifier [1] has proven to be an effective method in the field of machine learning. SVM possesses advantages with respect to the management of high dimensional data and reveals effective generalization capability. It has been widely used in various applications, including handwritten digits recognition [2,3], time series classification [4,5], gene selection [6,7], and image retrieval [8–10].

However, SVM assumes that samples are represented with vectors of real numbers [11]. If nominal attributes exist, they are usually converted into numerical attributes before learning occurs. Integer and one-of-n coding are popular methods used in managing nominal attributes. If the number of values in a nominal attribute is not large, one-of-n coding might be more stable than integer coding [11]. In fact, both methods possess disadvantages. Regarding integer coding, performance is easily affected by the coding mechanism because different coding methods lead to different distances between samples. With respect to one-of-n coding, a nominal attribute is mapped into multiple binary attributes. After one-of-n coding is completed, the number of attributes is equal to the number of values of the original nominal attribute. This method can effectively prevent instability problems in integer coding. However, it may dramatically increase the dimensions of samples if a lot of different values exist in the nominal attributes. Furthermore, both integer coding and one-of-n coding do not take full advantage of the implicit classification information of samples.

Three different methods exist for managing heterogeneous data. The first is to convert nominal attributes to integers through coding, and then consider them as numerical attributes. Its major problem is instability as the performance is easily affected by the use of a coding mechanism. The second method is to discretize numerical attributes, and then treat them as nominal attributes, as done in C4.5 [12], classification and regression tree (CART) [13] and other methods. In general, discretization causes information loss. The third method is to learn a distance, such as the value difference metric (VDM), heterogeneous value difference metric (HVDM) and other methods [14–16]. This type of method can be combined with classifiers based on distance (e.g. K-nearest neighbor) [17,18]. In distance learning algorithms, we usually adopt an overlap method or a Bayesian approach to deal with nominal attributes. The overlap is a simple and effective method. However, it only determines whether nominal attributes are equal to one another, and does not fully exploit classification information. The Bayesian approach is very effective for handling nominal attributes. However, the use of this approach implies that all attributes are independent. Therefore, its performance will degenerate if relation among attributes is very high. Such as XOR data, the probability of each attribute is the same, VDM then results in zero distance between attributes [19]. Moreover, the performance of these algorithms may deteriorate when decisions depend on multiple attributes [19].

Essential differences exist between nominal and numerical attributes. In general, a numerical attribute describes a particular feature of a sample. If the value of a numerical attribute is changed, the entire sample is changed such that the new sample is no longer the

[*] Corresponding author.

http://dx.doi.org/10.1016/j.patcog.2014.12.015
0031-3203/© 2014 Elsevier Ltd. All rights reserved.

previous. However, the value of a nominal attribute simply indicates a certain nominal value and does not describe the specific character of the sample. Regardless of the value of nominal attribute, as long as the same nominal attribute is assigned the same value, no problems will occur. Thus, the nominal attribute is not limited to a fixed value which makes it possible for a nominal attribute to be mapped into a real number according to the classification information. Based on this observation, we develop a new approach to manage nominal attributes. In order to deal effectively with heterogeneous data, we use classification information by mapping nominal attributes into a real space based on generalization error estimation. The values of nominal attributes are obtained from an optimization task rather than from integer or one-of-n coding. After mapping is completed, nominal attributes are treated numerically in the subsequent learning procedure.

SVM has been successfully applied to various classification tasks that use numerical data. However, the topic of training SVM with heterogeneous data has not been fully examined. In this study, we design a novel heterogeneous support vector machine (HSVM) algorithm to classify heterogeneous data. Our HSVM maps nominal attributes into a real space by minimizing generalization error. The main advantages of HSVM are listed as follows: (1) HSVM can effectively improve the performance of SVM in dealing with nominal data or heterogeneous data, (2) HSVM can improve the interpretability of decisions, and (3) HSVM is effective in learning with imbalanced data.

The remainder of this paper is organized as follows. Section 2 reviews related studies. Section 3 presents a novel mapping algorithm for nominal attributes and HSVM. Sections 4 and 5 analyze the experiments using standard datasets. Section 6 concludes our study.

Notations used in the paper are described as follows. The variable n represents the number of training samples and x_i represents a sample with an index i. For nominal attributes, we use a^k to refer to the kth nominal attribute of the samples. Its values are expressed as $\{a_1^k, a_2^k, \dots, a_m^k\}$.

2. Related works

We map nominal attributes into a real space by minimizing the generalization error, and then use SVM to manage heterogeneous data. Thus, in this study, we employ the SVM algorithm, generalization error, and heterogeneous data. In this section, we review relevant terms and algorithms.

2.1. SVM and kernel functions

SVM is an effective method for binary classification tasks. It constructs an optimal separating hyperplane in a feature space. By a function Φ, we map an input vector x into a high dimensional feature space [20]. Given n samples $\{(x_i, y_i)\}_{i=1}^n$, SVM searches for a linear decision function with a maximum margin between different classes in the feature space, where x_i is an input vector with d dimensions, and y_i is a class label of x_i. The decision function $f(x) = \langle w, \Phi(x) \rangle + b$ defines a linear hyperplane in the feature space. The parameters w and b are obtained by solving the following convex quadratic problem:

$$\min_{w,b} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i,$$
$$\text{s.t.} \quad y_i(\langle w, \Phi(x)\rangle + b) \ge 1 - \xi_i, \quad \xi_i \ge 0, \ \forall i, \tag{1}$$

where C is a constant that penalizes for the training errors and ξ_i is a slack variable. $w \in R^d$ and $b \in R$ are the parameters of the hyperplane [1]. Instead of solving this optimization problem, we use the Lagrangian dual function to obtain a dual formula:

$$\max_{\alpha} \quad \sum_{i=1}^{n}\alpha - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \langle \Phi(x_i), \Phi(x_j)\rangle,$$
$$\text{s.t.} \quad \sum_{i=1}^{n}\alpha_i y_i = 0, \quad 0 \le \alpha_i \le C, \ \forall i, \tag{2}$$

where α_i is a dual variable and contains the upper bound C. The inner product $\langle \Phi(x_i), \Phi(x_j)\rangle$ in the feature space is computed by a kernel function: $\langle \Phi(x_i), \Phi(x_j)\rangle = K(x_i, x_j)$. By means of the kernel function, the inner product in a high dimensional feature space can be efficiently computed without an explicit nonlinear mapping. The dual formula shown in (2) is a convex quadratic optimization problem and possesses a global optimal solution [21]. The linear kernel function ($K_{LIN}(x_i, x_j)$), polynomial kernel function ($K_{POL}(x_i, x_j)$) and Gaussian kernel function ($K_{GAU}(x_i, x_j)$) are widely used in the following:

$$K_{LIN}(x_i, x_j) = \langle x_i, x_j \rangle;$$
$$K_{POL}(x_i, x_j) = (\langle x_i, x_j \rangle + 1)^q, \quad q \in N;$$
$$K_{GAU}(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right), \quad \sigma \in R_+. \tag{3}$$

2.2. Generalization error estimation

Some techniques used to estimate generalization errors, such as *leave one out* (LOO) as well as *span* and *radius margin* estimations, are common. LOO estimation consists of three steps: (1) remove one element from the training data, (2) construct a decision function over the remained data, and (3) test the model with the removed element [20]. LOO is nearly unbiased as an estimator of the expected generalization error [1], where the estimation is given as

$$E(p_{err}^{n-1}) = \frac{1}{n}E(L(x_1, y_1, \dots, x_n, y_n)), \tag{4}$$

where p_{err}^{n-1} is a probability of a classification error tested on the samples of size $n-1$, and $L(x_1, y_1, \dots, x_n, y_n)$ is the number of misclassified samples. LOO is an important statistical estimator of learning algorithms and it is frequently used in model selection. Unfortunately, it is time-consuming, as testing of each element in the training samples is required. Some generalization error estimations are derived from LOO, such as the *span* and *radius margin* estimations [1].

The concept of *span* for support vectors was first proposed by Chapelle and Vapnik [22]. The *span* is derived from an LOO error estimation [1,20] and the upper bound of the *span* is computed by means of

$$T = \frac{1}{n}\sum_{i=1}^{n}\Psi(\alpha_i^* s_p^2 - 1), \tag{5}$$

where Ψ is a step function (i.e. $\Psi(x) = 1$ if $x \ge 0$, and $\Psi(x) = 0$ otherwise) [20] and α_i^* is the optimal solution for dual formation, as shown in (2). The variable s_p^2 is the distance between the point $\Phi(x_p)$ and set Λ_p in the feature space where

$$\Lambda_p = \left\{ \sum_{i \ne p, \alpha_i^* \ge 0} \lambda_i \Phi(x_i), \sum_{i \ne p} \lambda_i = 1 \right\}. \tag{6}$$

The *span* is an upper bound of LOO and is not continuous. A small change in kernel functions causes a considerable change in the support vector set Λ_p. This change is discontinuous and results in discontinuous changes to s_p^2 and error bound T [20].

The *radius margin* estimate is another generalization error estimation and can be considered as a rough upper bound of the *span* estimation. Suppose that the maximal distance between different classes is γ, and R is the minimum radius of a sphere

Download English Version:

https://daneshyari.com/en/article/530166

Download Persian Version:

https://daneshyari.com/article/530166

Daneshyari.com