Volume-5, Issue-3, March - 2016 • ISSN No 2277 - 8160 IF : 3.62 | IC Value 70.36

Research Paper Mathematics

A Feature Extraction Method By Employing Optimization Theory And Space Rotation

Jialin Tian

School of Computer Science and Software Engineering, Tianjin Polytechnic University, Tianjin 300387

ABSTRACT

In last decade, we have witnessed a burst of data in all fields. Mining the patterns and reducing the dimensionality of the data space is of particular value. In previous studies, the Principal Component Analysis method is frequently employed in dimension reduction and feature extraction. In this study, we propose a novel feature extraction method. This method integrates the concept of space rotation and optimization theory. By a number of iterations of space rotation, the information that is useful for classification is accumulated to the first several dimensions. A comprehensive experiment on 14 datasets and 3 classification algorithms demonstrate that the proposed algorithm is superior to the Principal Component Analysis method.

KEYWORDS : dimension reduction, feature extraction, space rotation

INTRODUCTION

In the last decade, we have witnessed a burst of data accumulated by Internet enterprises, government and research institutions. Mining the hidden patterns and knowledge from massive disordered data has received more and more attention. Among the field of data mining and machine learning, feature extraction, which transforms original data from a high dimensional feature space to a low dimensional feature space while retaining the useful information, is of particular importance. In previous studies, Principal Component Analysis (PCA) is widely employed for feature extraction. The PCA method rotates the original data space such that the axes of the new coordinate system point into the directions of highest variance of the data. However, the variance of data has clear limitations and is not suitable for classification problems. To this end, we have proposed a new method for feature extraction that is based on optimization theory.

METHODS

Rotation matrix

In a two dimensional space, a rotation matrix is written as

$$R(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

While in the case of *n*-dimensional feature space, the rotation matrix is more complex. It is written as , where

$$R = M_{1,2}^{2,3} \cdot M_{1,3}^{2,3} \cdots M_{1,n}^{2,3} \cdots M_{1,n}^{n-1,n}, \text{where}$$

$$M_{i,j} = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & \cos\theta & \cdots & -\sin\theta & \\ & & \vdots & \ddots & \vdots & \\ & & \sin\theta & \cdots & \cos\theta & \\ & & & & & \ddots \\ & & & & & & 1 \end{pmatrix}$$

where the first locates at the intersection of the *i*-th row and *j*-th column.

Optimization method

Denote is the input feature space and , . For the calculation of a hyper-plan that distinguishes samples of different classes, we need to solve the following problem:

$$\begin{cases} \min \Phi(w) = \frac{1}{2}(w \cdot w) \\ s.t.\ y_i[(x_i \cdot w) - b] \geq 1,\ i = 1,2,\cdots,l \end{cases} \quad (1)$$

If the samples of the two classes cannot be cleanly classified, we introduce the soft variables,

$$\xi_i \geq 0,\ \ i = 1,2,\cdots,l \quad (2)$$

Therefore, the optimization problem is transformed to:

$$\begin{cases} \min \Phi(w,\xi) = \frac{1}{2}(w \cdot w) + C \cdot (\sum_{i=1}^{l} \xi_i) \\ s.t.\ y_i[(x_i \cdot w) - b] \geq 1 - \xi_i \\ \xi_i \geq 0,\ i = 1,2,\cdots,l \end{cases} \quad (3)$$

By employing the Lagrange Multiplier, we can calculate a hyper-plan for problem (3).

Feature extraction algorithm

The proposed algorithm consists of the following 6 steps:

1. Solve problem (3) and calculate the hyper-plan P_n by using the complete feature space. The normal vector of P_n is denoted as W_n.
2. Calculate the rotation matrix based on W_n.
3. Perform the second step for *n*-1 times and calculate the rotation matrix . Multiply the rotation matrixes calculated in the second and third steps to the feature space.
4. Through the rotation process, the 2nd to the last dimensions of W_n are transferred to 0. In other words, W_n parallels to the *x*-axis.
5. In the rotated feature space, remove the first dimension and the remaining *n*-1 features constitute a new feature space W_{n-1}.
6. If the dimension of the new feature space is greater than 1, go to step 1; otherwise, the algorithm is finished.

EXPERIMENT SETUP

The datasets used in this study come from the standard UCI database. We strictly follow the 10 folds cross validation protocol. We have also performed data preprocessing as follows.

Samples with missing values

Some samples in the UCI datasets are with missing values. These samples are removed from the datasets in our experiments. A dataset is retained if it contains at least 200 valid samples.

Samples with discrete values

The optimization method needs numerical inputs and cannot process discrete values. Hence, the discrete values in the original datasets need to be processed. If a feature contains *n* different discrete values, we transform the discrete value into a *n*-dimensional vector. For instance, the *i*-th discrete value is represented as (0, 0, …, 1, 0,…, 0), where the *i*-th value of the vector is 1 and the remaining values are 0.

Data normalization

As data normalization can speed up the coverage of gradient descent algorithm and improve the performance of classification algorithms, we have performed data normalization on our datasets. All feature values are transferred to 0.1~1.0 by the following method:

- Denote the maximal value of a feature as , the minimal value of a feature as , the value before normalization as , and the value after normalization as .
- The transformation between and is:

$$V' = \frac{V - V_{min}}{V_{max} - V_{min}} \cdot \frac{9}{10} + 0.1$$

Setup for classifiers
We choose three distinct classifiers for the evaluation of our method, including NaïveBayes, Kstar and DecisionStump. The basic concept of the three classifiers is given as follows:

NaïveBayes is a classification algorithm based on the Bayes' theorem. It assumes that all features are independent.

Kstar is a classification algorithm based on Euclidian distance. A sample with unknown class label is determined by the *k* closest samples that have been assigned with class labels.

DecisionStump is a classification method that is similar to Decision tree. The difference is that Decision tree contains multiple layers while DecisionStump contains only one layer.

The three classifiers were run with default parameters and on three different feature space:

The original feature space: Features are the same as the UCI database. No feature selection or extraction is performed.

Feature space extracted by PCA: We employ the PCA algorithm on the original feature space and the algorithm generates a new feature space.

Table 1. Prediction accuracy of DecisionStump on three distinct feature spaces.

Dataset	DecisionStump		
	Original feature space	PCA feature space	Rotation matrix feature space
credit-g	70.00%	70.00%	72.20%
clean1	58.12%	60.70%	96.18%
breast-w	66.37%	57.65%	73.55%
house-votes-84	69.39%	73.30%	72.96%
kr-vs-kp	90.93%	96.36%	97.37%
breast-cancer	86.35%	84.68%	84.81%
diabetes	71.62%	69.67%	72.78%
credit-a	71.85%	67.78%	84.07%
heart-statlog	96.93%	87.87%	93.49%
vote	82.38%	83.49%	87.56%
mushroom	89.90%	85.29%	97.51%
sonar	71.31%	66.71%	63.68%
ionosphere	69.95%	62.30%	98.32%
tic-tac-toe	96.93%	87.87%	93.49%

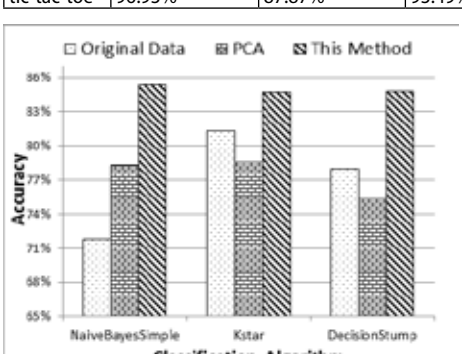

Figure1. Prediction accuracies of NaiveBayes, Kstar and DecisionStump algorithms on three distinct feature spaces: the original data space, the PCA feature space and the feature space generated by this method.

eature space extracted by the rotational matrix algorithm: We employ the rotational matrix algorithm (proposed in this paper) on the original feature space and the algorithm generates a new feature space.

RESULTS
The NaiveBayes, Kstar and DecisionStump algorithms are run in three distinct feature spaces. Prediction accuracy is used to evaluate the performance.

The results of DecisionStump are given in Table 1. For 12 out of 14 datasets, the DecisionStump algorithm achieves higher prediction accuracy on the rotation matrix feature space than on the PCA feature space. For 10 out of 14 datasets, the DecisionStump algorithm achieves higher prediction accuracy on the rotation matrix feature space than on the original feature space. On average, when the DecisionStump algorithm is employed, the prediction accuracy on the rotation matrix feature space is 85%. To compare, the prediction accuracies are 78% and 75% on the original feature space and PCA feature space respectively, see Figure 1.

Similar trends are observed for Kstar and NaiveBayes algorithms. For 11 out of 14 datasets, the Kstar algorithm achieves higher prediction accuracy on the rotation matrix feature space than on the PCA feature space. For 12 out of 14 datasets, the Kstar algorithm achieves higher prediction accuracy on the rotation matrix feature space than on the original feature space. On average, when the Kstar algorithm is employed, the prediction accuracy on the rotation matrix feature space is 85%. To compare, the prediction accuracies are 81% and 79% on the original feature space and PCA feature space respectively, see Figure 1. Similarly, for all 14 datasets, the NaiveBayes algorithm achieves higher prediction accuracy on the rotation matrix feature space than on the PCA feature space. For 12 out of 14 datasets, the NaiveBayes algorithm achieves higher prediction accuracy on the rotation matrix feature space than on the original feature space. On average, when the NaiveBayes algorithm is employed, the prediction accuracy on the rotation matrix feature space is 85%. To compare, the prediction accuracies are 71% and 78% on the original feature space and PCA feature space respectively, see Figure 1.

CONCLUSIONS
This study proposes a feature extraction method that integrates the rotation matrix technique and the optimization theory. Comprehensive experiments on 14 datasets and 3 classification algorithms demonstrate that the proposed rotation matrix feature space is superior to the original feature space and the PCA feature space.

REFERENCES:
[1] Burges C.J .C.(1998). "A tutorial on support vector machines f or pattern recognition". Data Mining and Knowledge Discovery, 2 (2) : 1~47
[2] Vapnik V.N.(1999), "An overview of statistical learning theory." IEEE Transactions on Neural Network , 10 (5) : 988~999