Predicting Election Result Using Candidates' Reply on a Questionaire as Multidiemnsional Input Data

Md. Mohsin Ali Khan(Student No: 336790)
Yangjun Wang (Student No: 401450)
T-61.3050 Term Project, final report

November 17, 2013

Abstract

In this report the data from election commission have been used to predict the outcome of the election for a candidate and the associated political party name. The input data is multidimensional. Cross Validation has been used with the technique of Bayes Classification, Naive Bayes Classification,Nearest Mean and K-Nearest Neighbour. All these techniques have been applied with the presence and absence of both principal component analysis. Decision tree has also been used to do the prediction. It is found that some technique is better for election outcome prediction for the candidates and some is better for predicting the political party they come from. The detail methods and their results are discussed. By cross validation technique certain classification technique is finalized and the result of that technique on test data is presented.

Introduction

It is very common that people try to predict the result of the election. Candidates connect with people through media and try to present their views and commitment to make themselves acceptable to the voters. They answers to different questions on their ideology, commitments and future plan if they are elected to media or to the election commission. These questions and their answers are treated as the input of a prediction system. This study can help the political parties to understand the concerns of people and can help them to make their strategy for the future days.

Methods

There is two set of data. One is training set which is labelled. All the data in the training set has the information whether the candidate is elected and

which party they come from. The training set is split in two parts randomly. One is called training set and the other is validation set. All the techniques are applied on the training set first. Then the trained models are applied on the validation set and the F-score is calculated. Here is the formula of F-score. $F = \frac{2 \times Precision \times Recall}{(Precision + Recall)}$, when $TP > 0$ and $F = 0$, when $TP = 0$. Where $Precision = TP/(TP + FP)$, $Recall = TP/(TP + FN)$. $FP =$ "Number of false positives", $FN =$ "Number of false negatives" and $TP =$ "Number of true positives". For party labels, a separate F-score for each class (party) and an average over all of them is calculated.There is two task. One is to predict the election result and the other is to predict the party the candidate comes from. So there is two F-score. F-score for election result and F-score for party prediction. The model which has the maximum F-score on validation set is chosen as the Model. The chosen model is then applied on the test data set to predict the election result and the party. The F-score on test data set is presented. Input data is multidimensional. Each data x is represented as a vector of d dimension. x_i is the ith dimension. Estimate of mean is: $\mu_i \rightarrow m_i = \sum_{t=1}^{N} \frac{{x_i}^t}{N}$. Estimate of co-variance is $\sigma_{ij} \rightarrow s_{ij} = \sum_{t=1}^{N} \frac{({x_i}^t - m_i)({x_i}^t - m_j)}{N}$ where ${s_i}^2 = s_{ii}$. The following techniques are applied on the training set.

1 Naive Bayes: Common co-variance matrix and class specific mean. And use the following discriminating functions. The function that have the maximum value classify the result to the corresponding class of that function. $g_i(x) = -\frac{1}{2} \sum_{j=1}^{d} \frac{({x_j}^t - m_{ij})^2}{{s_j}^2} + log\hat{P}(C_i)$ where d is the number of dimensions that the data has.

2 PCA and Naive Bayes: Do Prinicpal Compontent analysis first. Take class specific mean using the principal components. Calculate the common covariance matrix using the principal components.Then use the discriminating functions

3 Nearest Mean: The means are class-specific, covariance matrix Σ is common and proportional to unit matrix. The discriminant function is: $g_i(x) = -\|x - m_i\|^2$

4 PCA and Nearest Mean: Apply the Nearest Mean technique using the principal components

5 K Nearest Neighbour: Find k nearest neighbour of x. Predict the class of x to be the majority class of k nearest neighbours. Parameter k is defined by cross validation.

6 PCA and K Nearest Neighbour: K nearest neighbour is applied on the principal components

7 Decision Tree: Uses classification tree.

Experiment

Naive Bayes

The data is multivariate data with 199 dimensions. It means that input data is a data set of 1×199 vectors, where the 199 dimensions correspond to 199 features. To deal with these multidimensional input data class specific mean for each dimension and a common co-variance matrix is calculated.The discriminant function mentioned in the method section is used and the class is defined with the following equation:

$$Class_{predicted} = arg\,max_i g_i(x) \textbf{ where } g_i(x) = -\frac{1}{2}\sum_{j=1}^{d}\frac{(x_j - m_{ij})^2}{{s_j}^2} + log\hat{P}(C_i)$$

Where m_{ij} is the mean of jth dimensionality of class i. ${S_j}^2$ is variance of jth dimension. $P(C_i)$ is the probability of class i. The training set had 1137 input vectors. 800 input vectors were used for training the model. Which means, the class specific mean and the common co-variance matrix is calculated using these 800 input vectors. Then the above discriminating function is applied on the rest of the 237 vectors. And then the F-score of the prediction for these 237 vectors are considered to be the F-score of validation data set.

PCA and Naive Bayes

The principal components of the 800 input vectors are calculated first. Then the class specific means of each dimension m_i and the co-variance matrix Σ is calculated using the principal components. Then using these means and co-variance the discriminant function is applied on the validation set. Before applying the discriminant function on an input x, the PCA is done on x.

Nearest Mean

The means (m_i) of every dimensions which has the same result classification i (for i from 1 to K). K is 2 for election classification, 17 for party classification. In our experiment, among 1037 rows data, 800 rows are used for training data.

PCA and Nearest Mean

The means are calculated on the principal components

K Nearest Neighbour

K has been chosen by validation. Choosing the value of K to be 3 has given the optimum result for predicting the election and choosing the value of K to be 8 has given the optimum result for predicting the party. So, here the K Nearest Neighbour has become 3-Nearest Neighbour and 8-Nearest Neighbour.

Decision Tree

A decision tree is a hierarchical data structure implementing the divide-and-conquer strategy. We used classification tree in our project, which is a decision tree for classification. It is a hierarchical model for supervised learning whereby the local region is identified in a sequence of recursive splits in a small number of steps. Given an input data at a node of classification tree, the outcome are branches of this node, it would be a leaf node given one class of this classification, or it would be internal decision nodes. According to the properties of the training data, there are 199 dimensions in the data set and each dimension would be binary or not, continue or distribute. We implement classification tree as following steps. 1. Load Data 2. Make a classification tree electTree using classregtree function. Classregtree is a function of Matlab, creating a decision tree. In this step, we got two trees , one for election prediction and another for parities prediction.

Result

Method	$F_{v,e}$	$F_{v,p}$	$F_{t,e}$	$F_{t,p}$
Naive Bayes	0.23944	0.44112	0.12030	0.07629
PCA and Naive Bayes	0.22785	0.36533	0.10884	0.05592
Nearest Mean	0.23776	0.02211	0.13284	0.01628
PCA and Nearest Mean	0.25758	0.01667	0.10601	0.03315
K Nearest Neighbour	0.35616	0.24979	0.56911	0.36604
Decision Tree	0.43478	0.28027	0.21176	0.25401

In the above table $F_{v,e}$, $F_{v,p}$, $F_{t,e}$, $F_{t,p}$ stands for F-score of Validation set for election and party and of test set for election and party respectively.

Discussion

From the validation F-score we find that the decision tree has the best F-score for both election and party prediction. So, we choose the Decision tree to be our model. Unfortunately we find that the K-Nearest Neighbour gives the best result of test data. The result we have achieved so far is not yet that much good. But we observe that some techniques are better than some other techniques predicting the classes. We believe that there are some techniques or variation of the techniques that we have used which can generate better F-score. We ran out of time to try out further techniques.

Reference

The techniques and mathematical formulas are taken from the Lectures of the course.

Appendices