Pattern Recognition Project-1

9957513
多媒體工程所
宋秉一

Content

Maximum Likelihood Estimation

To estimate the normal PDF of class ω from training data, we use the Maximum Likelihood method.
Given N feature vectors in class ω, and X_m $(m = 1 \ldots N)$ is $k \times 1$ feature vector, where k is feature dimension.

1-Dimensional Normal distribution

The estimated parameters of 1-Dimensional normal PDF in dimension j $(j = 1,2 \ldots k)$

$$\hat{u}_j = E[X] = \frac{1}{N}\sum_{m=1}^{N} X_{mj}$$

$$\hat{\Sigma}_j = E[(X_j - \hat{u}_j)(X_j - \hat{u}_j)^T] = \frac{1}{N}\sum_{m=1}^{N} (X_{mj} - \hat{u}_j)(X_{mj} - \hat{u}_j)^T$$

The normal PDF for class ω with dimension j could be estimated by the parameters $\hat{u}_j, \hat{\Sigma}_j$.

$$P(x_j|\omega) = \mathcal{N}(\hat{u}_j, \hat{\Sigma}_j)$$

Multivariate normal distribution

The estimated parameters of multivariate normal PDF

$$\hat{u}_{k\times 1} = E[X] = \frac{1}{N}\sum_{m=1}^{N} X_m$$

$$\hat{\Sigma}_{k\times k} = E[(X - \hat{u})(X - \hat{u})^T] = \frac{1}{N}\sum_{m=1}^{N} (X_m - \hat{u})(X_m - \hat{u})^T$$

The normal PDF for class ω could be estimated by the parameters in the previous section.

$$P(x|\omega) = \mathcal{N}(\hat{u}, \hat{\Sigma})$$

Bayesian Classifier

Introduction

The Bayesian Classifier presents the PDF function with all dimensions by applying the covariance matrix. The PDF is used for classification which we choice the classification result with maximum probability in PDF.

Decision Rule

Classify x to class ω_i

$$P(\omega_i|x)P(x) > P\left(\omega_j\middle|x\right) P(x) \;\; \forall\, j \neq i$$

$$P(x|\omega_i)P(\omega_i) > P\left(x\middle|\omega_j\right)P\left(\omega_j\right) \forall\, j \neq i$$

Where $P(x|\omega)$ and $P(\omega)$ are known from the training data.

Naïve-Bayes Classifier

Introduction

The Naïve-Bayes Classifier is an independent feature mode, applying the Bayesian theorem with independence assumptions of each feature dimension.

Decision Rule

Instead of considering the all dimension variance by covariance matrix, the class PDF will be the product of the PDF in each dimension.

$$P(x|\omega_i) = \prod_{j=1}^{k} P\left(x_j\middle|\omega_i\right)$$

Classify x to class ω_i

$$P(\omega_i|x)P(x) > P\left(\omega_j\middle|x\right) P(x) \;\; \forall\, j \neq i$$

$$P(x|\omega_i)P(\omega_i) > P\left(x\middle|\omega_j\right)P\left(\omega_j\right) \forall\, j \neq i$$

Where $P(x|\omega)$ and $P(\omega)$ are known from the training data.

Program Workflow

Figure 1.Program Workflow diagram

The main program includes three subprograms. The first part is the file loader, the file loader loads classes file as a struct array of classes, and each class object has some class attribute.

The second part is the training module. In this module, the classes will be partitioned into training set and testing set. We get the PDF parameters from the training set in each class. The testing set will be used in the next module.

The third part is the validation module. We use k-fold cross validation on the testing set which we obtained from the training module. Each testing sample will get the PDF probabilities correspond to every class, and we determine the maximum corresponded class as the classification result. The confusion matrix we used for evaluate will be mentioned in the next section.

Experiment Result

K-fold cross validation

We use the k-fold cross validation to measure the performance of classifier. The original datasets are partitioned into k subsets randomly. Each of the subset use once as the testing data, and the other subsets are used for training. Compute the classification performance using the average result of k testing datasets. The confusion matrix M shows below, M_{ij} presents the actual class i with predict class j.

	A	B	C
A	10	0.0	0.0
B	0.0	9.6	0.4
C	0.0	0.2	9.8
Error rate		0.0200	

Table 1.Average Bayesian confusion matrix with k=5.

	A	B	C
A	9.6	0.0	0.4
B	0.0	8.8	1.2
C	0.0	0.6	9.4
Error rate		0.0733	

Table 2.Average Naive-Bayes confusion matrix with k=5.

	A	B	C
A	5.0	0.0	0.0
B	0.0	4.7	0.3
C	0.0	0.1	4.9
Error rate		0.0267	

Table 3.Average Bayesian confusion matrix with k=10.

	A	B	C
A	4.8	0.1	0.1
B	0.0	4.4	0.6
C	0.0	0.3	4.7
Error rate		0.0733	

Table 4.Average Naive-Bayes confusion matrix with k=10.

Feature Dimension selection

We project data into each one dimension and see the distribution to analysis the feature dimension we should select. The class 1, 2, 3 are plot as red, green and blue points show as below. Try to find which dimension is more separable. The dimension 3 and 4 is obviously separate the class 3 and the others.

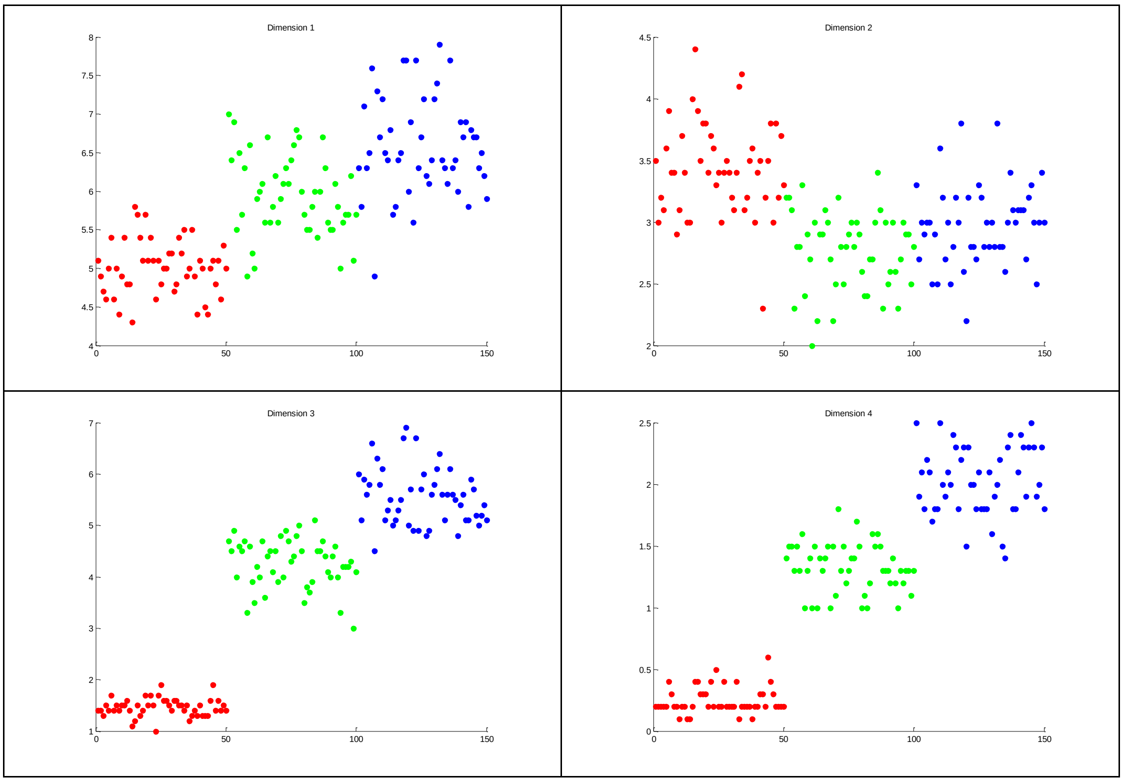

Figure 2.One dimension projections

We have tried every combinations of the feature dimension in the given data sample to find the best separable dimensions. The conclusion will be mention in the next section. The color blocks are the smallest error rate in each classifier.

Dimension Index	Bayesian Error rate	Naive-Bayes Error rate
1 2 3 4	0.0200	0.0733
1 2 3	0.0400	0.1467
2 3 4	0.0400	0.0667
1 3 4	0.0267	0.0600
1 2	0.2333	0.2400
1 3	0.0400	0.1000
1 4	0.0400	0.0467
2 3	0.0467	0.1067
2 4	0.0400	0.0533
3 4	0.0333	0.0467

Table 5.Error rate in different dimension combinations with k=5.

Discussions

The experiment result shows that the Bayesian classifier result less classification error than Naive-Bayes classifier. Due to the definition of Naïve-Bayes theorem that the features in different dimensions are independent, we consider that the features are not exactly independent from the others in the given data sample. But with only input dimension 3 and 4, the classification result of Naive-Bayes classifier is better than other dimension combinations.

Source Code

We use *MATLAB* to implement the two classifiers.

How to use

The main function includes four input arguments.
file name (string): the data file path.
k (int): k-fold cross validation.
rand (int): randomize input data index order with each class (0 = false, 1=true).
dimIndex (int array): select dimensions of data.(empty array [] for all dimension). The length should be smaller than input dimension.

```
>>main(filename, k, rand, dimIndex);
```

Example-1

The following example reads data from “data-iris.txt”, using 5-fold cross validation, no random data order with all dimension input to training and testing.

```
>>main('data-iris.txt', 5, 0, []);
average Bayesian confusion matrix:
  10.0000        0        0
        0   9.6000   0.4000
        0   0.2000   9.8000
Bayesian error rate
   0.0200
average Naive-Bayes confusion matrix:
   9.6000        0   0.4000
        0   8.8000   1.2000
        0   0.6000   9.4000
Naive-Bayes error rate
   0.0733
```

Example-2

This example reads data from “data-iris.txt”, using 10-fold cross validation, random data order with dimension 1 and 2 input to training and testing.

```
>>main('data-iris.txt', 10, 1, [1 2]);
```