CS5011: Programming Assignment #1

Girish Raguvir J, CS14B059, IIT Madras

September 2016

1 Synthetic Data-set Creation

We need two parameters to define a Gaussian distribution. It is the mean and the covariance. In the given problem, we need to define two multivariate Gaussian with different means and same covariance matrix. The mean of the two Gaussian was defined by two randomly generated 20x1 matrices. The common covariance matrix was defined as A^tA, where A is a randomly generated 20x20 matrix, to ensure it is symmetric and positive-semidefinite.

Overlap between the two Gaussians was ensured by generating closely placed means. The second mean was obtained by adding a small random noise to the first mean.

These parameters for creating DS1 is documented in **p1_parameters.txt** under Results folder.

2 Linear Classification

A linear regression model was learnt by regressing on indicator variables.

2.1 Coefficients & Intercept

Documented in **p2_coefficients.txt** under Results folder.

2.2 Best Fit Performance Metrics

Best fit accuracy: 70.75%
Precision: 0.715
Recall: 0.69
F-measure: 0.702

3 k-NN classifier

In the given dataset, since we are handling balanced binary classes,the accuracy of classification is a very good performance measure. Thus plotting Accuracy vs K we get the Figure 1. As we can see, the graph obtained is not very smooth and does not seem to follow any particular trend. However, the best accuracy is obtained for K=6.

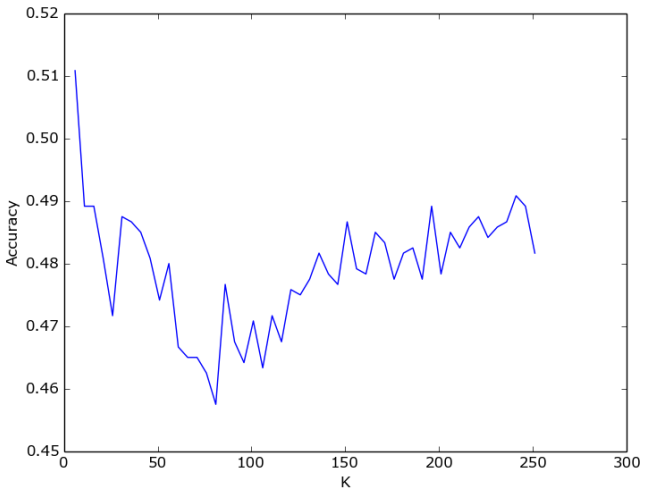


Figure 1: k-NN: Accuracy vs K

3.1 Best Fit Performance Metrics

Best fit accuracy: 51.08%
Precision: 0.5161
Recall: 0.347

F-measure: 0.4148

The performance of k-NN is quite bad when compared to linear regression with achieved an accuracy of 70%. The performance in this scenario is nearly as bad as random guessing, which reflects quite poorly on the model.

4 Data Imputation

For this problem we use the Communities and Crime Data Set from UCI repository. However, a lot of columns in the dataset provided have missing values. However when we go through the description of the dataset, we find that the first 5 features are described as non-predictive and hence we can safely remove them from the dataset instead of trying to fill in those nominal missing values. Other missing values are filled with the mean value of that column added with some uniform random noise.

We should refrain from filling all the missing values with just the mean so that a large bias is not introduced into the model.Filling in with means also attenuates any correlations involving the features that are imputed and reduces the variability of the data.This often causes biased estimates, irrespective of the underlying missing data mechanism

The addition of a small uniform random noise will reduce the above effects to some extent.

The dataset cleaned using the above logic has is saved as communities_cleaned.csv under Datasets folder.

5 Linear Regression

5.1 Coefficients & Intercept

Documented in **p5_coefficients.txt** under Results folder.

5.2 Best Fit Performance Metrics

Mean RSE: 0.01884

Here Mean RSE has been computed by averaging residual error over 5 different 80-20 splits.

6 Regularized Linear Regression

The residual error,along with the coefficients learned, for various values of lambda has been documented in **p6_lambda_values.txt** under Results folder.

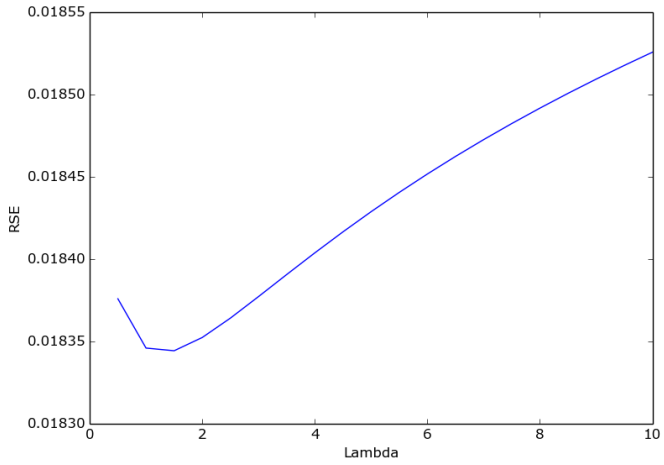


Figure 2: Ridge: RSE vs Lambda

The value of lambda which gives the best best fit is lambda = 1.5. As we can see from Figure-2, the RSE vs Lambda graph hits a minima at lambda = 1.5.

Yes, we can use the magnitudes of the coefficients learnt to determine the relative importance of the different features. Features with larger absolute value of coefficient can be concluded to be of more importance. We can then use this information to just pick out features which have absolute value of

coefficient to be greater than a given threshold and hence do feature selection. Lasso regression is more effective when it comes to using regression for feature selection but ridge can be used too.

Testing this idea out on the first split with a threshold of 0.05, we get

Intial number of features: 123
Intial RSE: 0.019914
Number of features after feature selection: 41
RSE after feature selection: 0.020975

Refer **p6_feature_selection.txt**, under Results folder, for coefficients learnt.

7 Feature Extraction-Problem 7

For this problem, we use PCA to extract 1 feature and we then use the data in this projected space to train a linear regression model using indicator random variables.

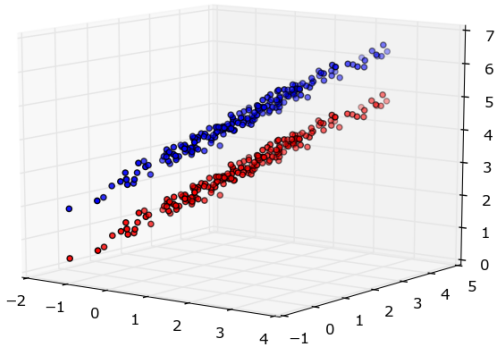


Figure 3: 3D plot of dataset

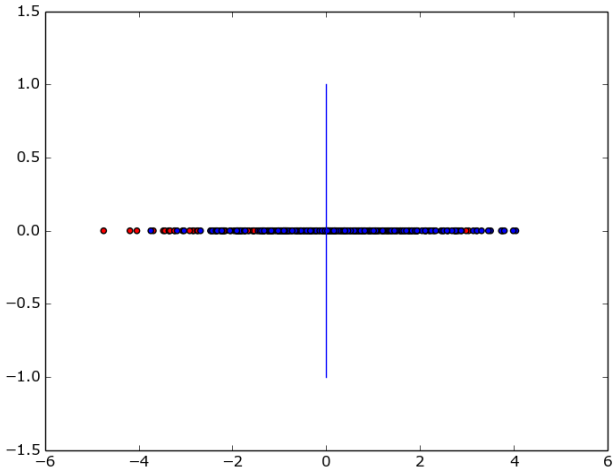


Figure 4: Plot of dataset in the projected space along with the classifier boundary

Refer Figure-3 for 3-D plot of the data.

Refer Figure-4 for plot of dataset in the projected space along with the classifier boundary.

7.1 Performance Metrics

Confusion Matrix:

$$\begin{vmatrix} 78 & 122 \\ 125 & 75 \end{vmatrix}$$

Accuracy: 38.25%

Class 0
Precision: 0.3842
Recall: 0.39
F-measure: 0.3870

Class 1
Precision: 0.3807

Recall: 0.375
F-measure: 0.3778

8 Feature Extraction-Problem 8

For this problem, we perform LDA and we then use the data in this projected space to train a linear regression model using indicator random variables.

Refer Figure-3 for 3-D plot of the data.

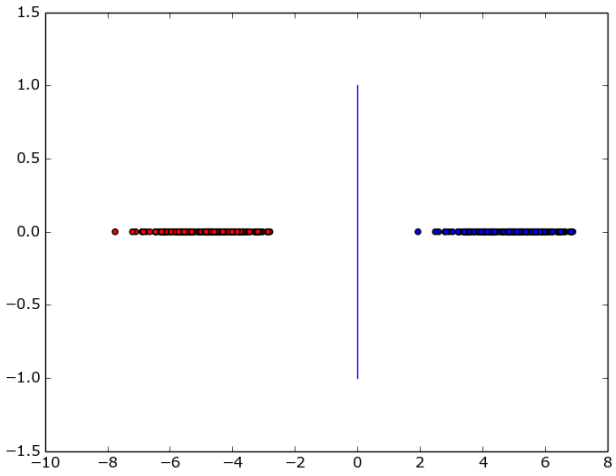


Figure 5: Plot of dataset in the projected space along with the classifier boundary

Refer Figure-5 for plot of dataset in the projected space along with the classifier boundary.

8.1 Performance Metrics

Confusion Matrix:

$$\begin{vmatrix} 200 & 0 \\ 0 & 200 \end{vmatrix}$$

Accuracy: 100%

Class 0
Precision: 1.0
Recall: 1.0
F-measure: 1.0

Class 1
Precision: 1.0
Recall: 1.0
F-measure: 1.0

8.2 Conclusion

Considering the performance of PCA and LDA in the given scenario, LDA is the obvious winner. From the 3-D plot of the dataset we can see the two classes are distinctly seperable and clearly the dimension chosen by LDA gave us a better seperation between the two classes, in the projected space, than PCA.

PCA is generally not optimal for classification.PCA doesnot not consider the class labels at all. While keeping the dimensions of largest variance is a good idea, it's not always enough as we see in the above scenario. The discriminant dimensions could be thrown out at times. LDA on the other hand considers both the features and the class labels and finds direction which maximize inter-class variance and minimizes intra-class variance. LDA thus performs better for the given scenario.

9 Logistic Regression

9.1 2-class Logistic Regression

In this problem, we perform Logistic Regression on the forest-mountain dataset.

9.1.1 Performance Metrics

Confusion Matrix:

$$\begin{vmatrix} 14 & 6 \\ 1 & 19 \end{vmatrix}$$

Accuracy: 82.5%

Class Mountain
Precision: 0.933
Recall: 0.7
F-measure: 0.8

Class 1
Precision: 0.76
Recall: 0.95
F-measure: 0.844

9.2 L_1-regularized Logistic Regression

In this problem, we perform L_1-regularized Logistic Regression on the forest-mountain dataset by using the code provided by Boyds Group.For identifying the optimal lambda for regularization, we plot the Accuracy vs Lambda curve and we chose the best lambda.

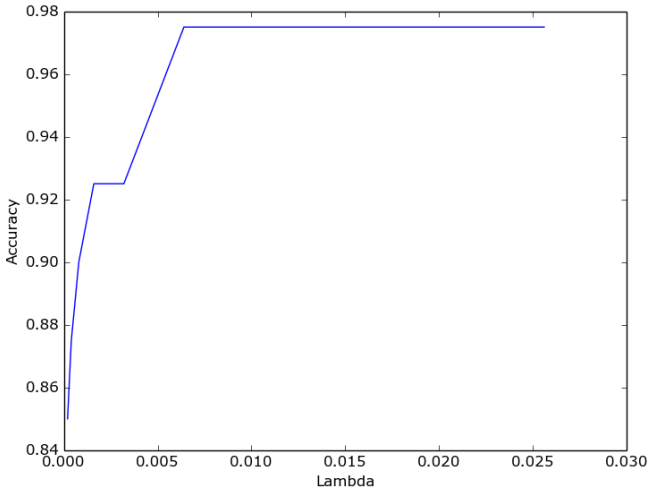


Figure 6: L_1: Accuracy vs Lambda

As we can see the graph plateaus after lambda = 0.064.So we just take lambda = 0.064 as the ideal lambda and the performance of that model is as given below.

9.2.1 Performance Metrics

Confusion Matrix:

$$\begin{vmatrix} 19 & 1 \\ 0 & 20 \end{vmatrix}$$

Accuracy: 97.5%

Class Mountain
Precision: 1.0
Recall: 0.95
F-measure: 0.9744

Class Forest
Precision: 0.952
Recall: 1.0

F-measure: 0.976

10 Naive Bayesian classifier

10.1 MLE with Multinomial Likelihood

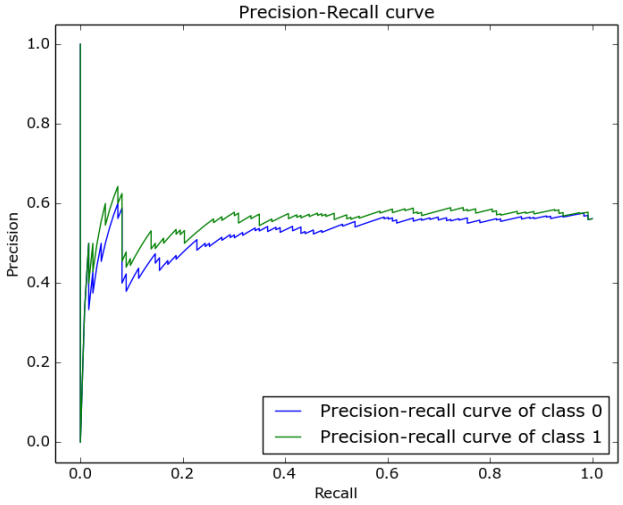


Figure 7: PR-Curve

10.1.1 Best fit performance metrics

Confusion Matrix:

$$\begin{vmatrix} 95 & 1 \\ 4 & 119 \end{vmatrix}$$

Accuracy: 97.7%

Class Spam
Precision: 0.9595
Recall: 0.9896
F-measure: 0.9744

Class Ham
Precision: 0.9917
Recall: 0.9675
F-measure: 0.979

10.2 MLE with Bernoulli Likelihood

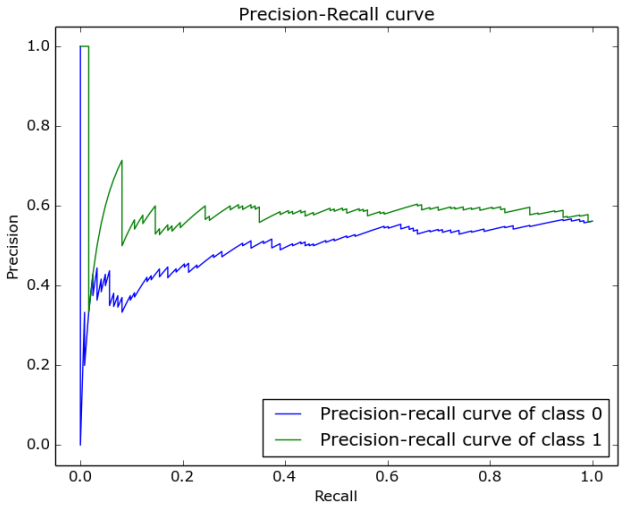


Figure 8: PR-Curve

10.2.1 Best fit performance metrics

Confusion Matrix:

$$\begin{vmatrix} 84 & 12 \\ 1 & 122 \end{vmatrix}$$

Accuracy: 94.06%

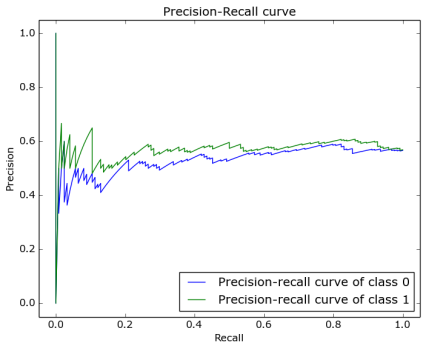
Class Spam
Precision: 0.988
Recall: 0.875

F-measure: 0.928

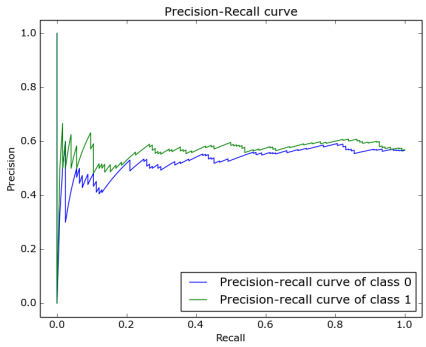
Class Ham
Precision: 0.91
Recall: 0.9918
F-measure: 0.949

10.3 Bayesian Parameter Estimation with Dirichlet prior

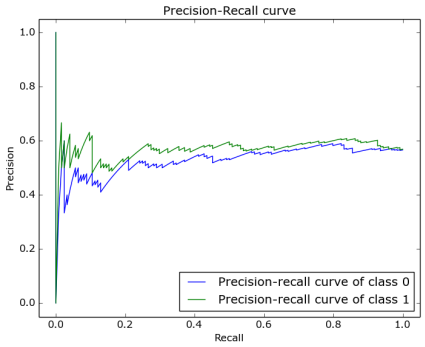
For this problem, I have used a Multinomial likelihood to make use of the conjugate pair property.



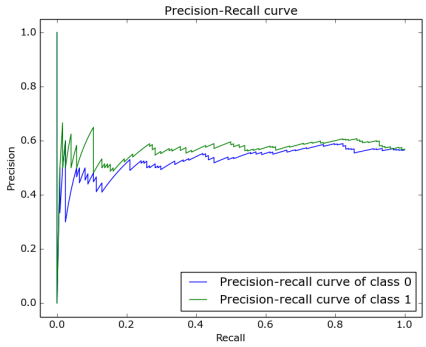
(a) alpha_1



(b) alpha_2



(c) alpha_3



(d) alpha_4

Figure 9: PR-Curves

10.3.1 Best fit alpha

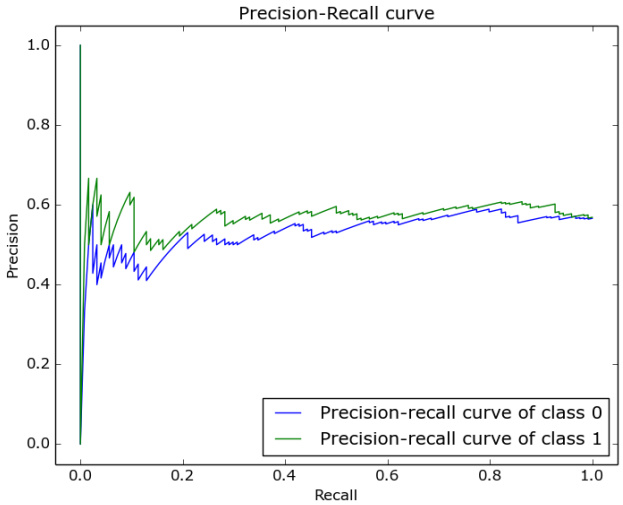


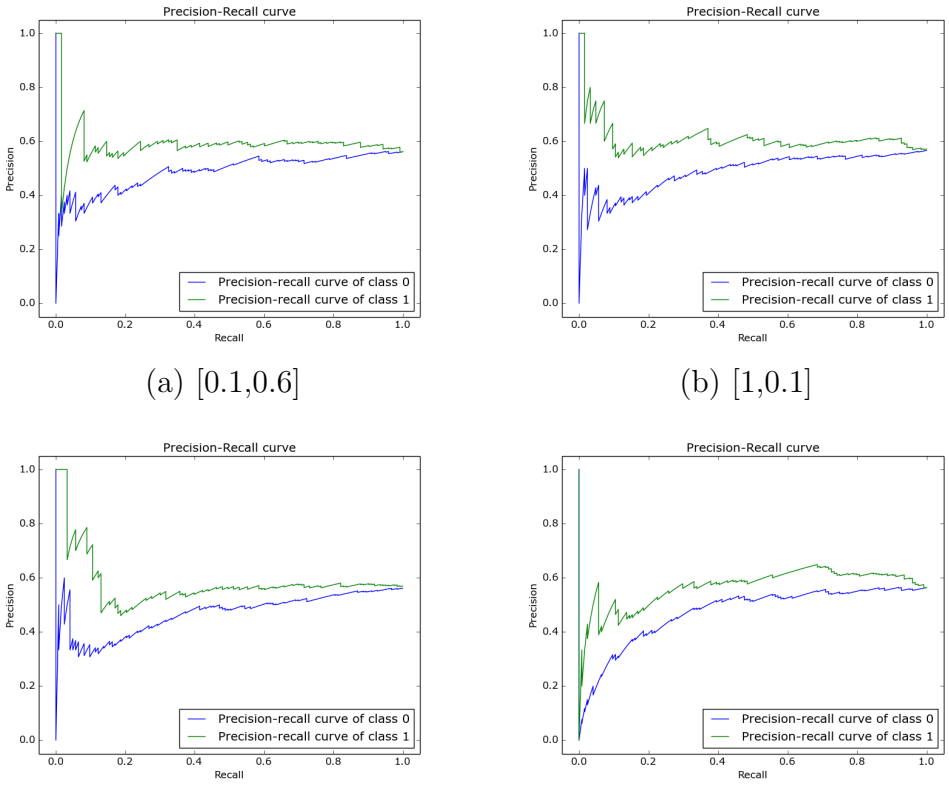
Figure 10: PR-Curve

10.3.2 Best fit performance metrics

Documented in best_fit_alpha_dir_p10.txt under Results folder.

10.4 Bayesian Parameter Estimation with Beta prior

For this problem, I have used a Bernoulli likelihood to make use of the conjugate pair property.



(a) [0.1,0.6] (b) [1,0.1]

(c) [4,4] (d) [10,5]

Figure 11: PR-Curves

Five-fold averaged metrics for the above values has been documented in beta_alphas.txt under Results folder.

10.4.1 Best fit alpha

The highest accuracy, out of the values tried, is obtained for the pair [0.1,0.6].

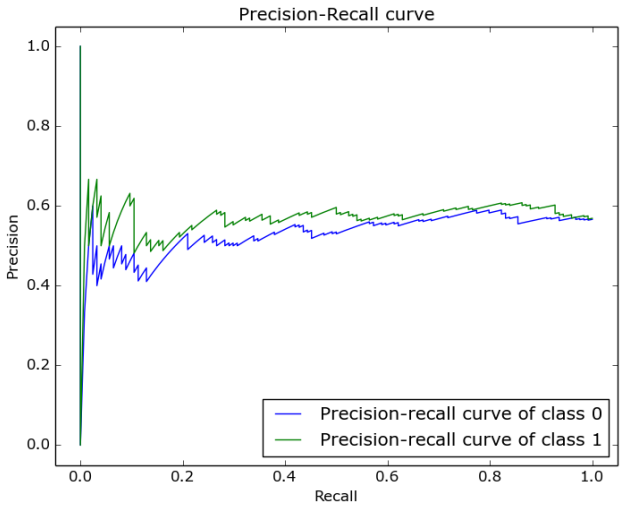


Figure 12: PR-Curve

10.4.2 Best fit performance metrics

Five-fold average.

Accuracy: 97.26%

Class Spam
Precision: 0.96
Recall: 0.98
F-measure: 0.97

Class Ham
Precision: 0.98
Recall: 0.97
F-measure: 0.975

10.5 Conclusion

As we can see the choice of the parameters clearly affect the performance of the model.Often the optimum set of parameters for these models can be

obtained by using proper tuning and update methods. Knowing something about the class distribution can also be incorporated through the prior to get a better model.