CS5691 - PRML Assignment 1

Meenakshi R (AE15B051), A V Lakshmy (CS16B101)

March 16, 2019

1 Naive Bayes and Bayes Classifier

1.1 Algorithm

We trained 3 models of Naive Bayes and 2 models of Bayes classifiers, on datasets 1 and 2. For this, we first found the class-wise prior probabilities, and the Gaussian class conditional densities. We used Maximum Likelihood Estimation for computing the class-wise means as well as covariances (varying as per the question). Then, we used the the Bayes Theorem to compute the accuracy of our classifier.

1.2 Observations and Plots

1.2.1 Training and Validation Accuracies for Datasets (1) and (2)

Model	Train Accuracy (1)	Validation Accuracy (1)	Train Accuracy (2)	Validation Accuracy (2)
1	58.70%	58.96%	58.38%	59.70%
2	96.57%	**98.07%**	91.33%	88.74%
3	96.57%	98.07%	93.52%	92.30%
4	96.35%	97.93%	91.68%	90.07%
5	96.57%	98.07%	96.00%	**94.37%**

The best model for **dataset 1** had a validation accuracy 98.07% (**Model 2: Naive Bayes, same covariance**), while that for **dataset 2** had a validation accuracy of 94.37% (**Model 5: Bayes, different covariance**).

1.2.2 Confusion Matrices on Test Sets for Best Models

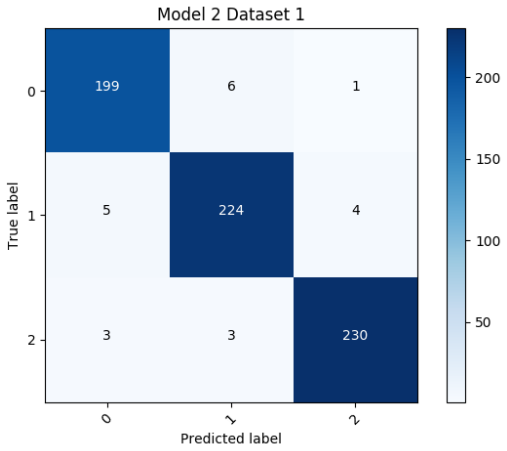


Figure 1: Dataset 1

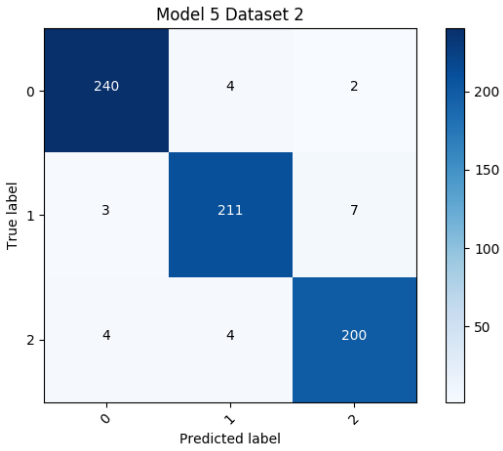


Figure 2: Dataset 2

1.2.3 Decision Boundaries for Best Models

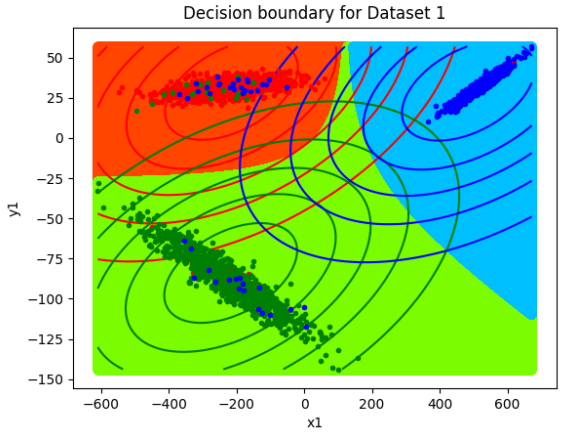


Figure 3: Dataset 1: Best Model Decision Boundaries

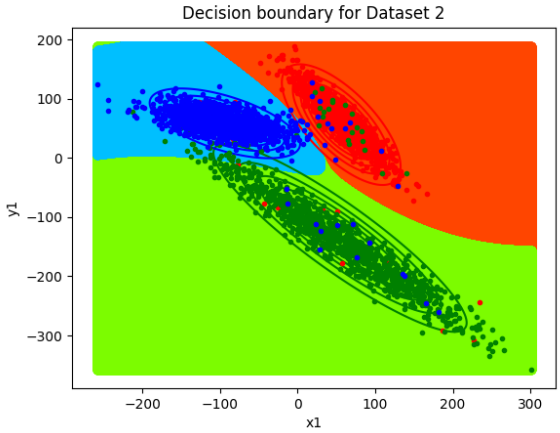


Figure 4: Dataset 2: Best Model Decision Boundaries

1.2.4 Decision Surfaces for Best Models

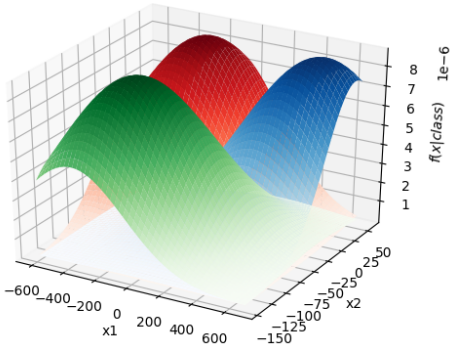


Figure 5: Dataset 1: Best Model Class Conditional Densities

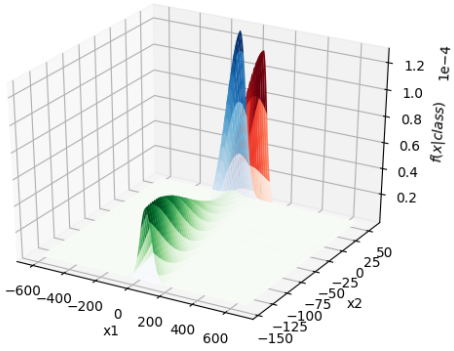


Figure 6: Dataset 2: Best Model Class Conditional Densities

Class 0	Class 1	Class 2
Red	Green	Blue

1.2.5 Contour Curves and Eigenvectors for Best Models

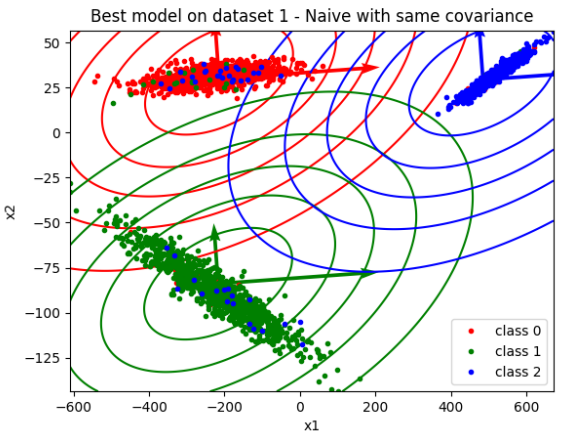


Figure 7: Dataset 1: Best Model Contours and Covariance Eigenvectors

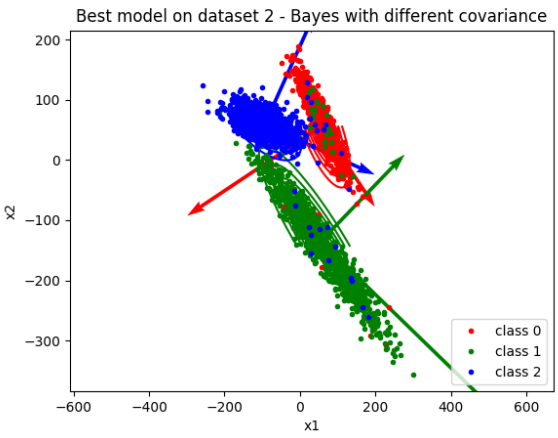


Figure 8: Dataset 2: Best Model Contours and Covariance Eigenvectors

1.3 Inferences

The best model for dataset 1 is Naive Bayes classifier because, from the contours (1.2.5), it seems that the features are independent (even if we know one of the features, it is possible to narrow it down to one of the classes, depending on the range of values taken). But for the dataset 2, we need a Bayes classifier, because we necessarily need both features to pinpoint the class (the features are not independent).

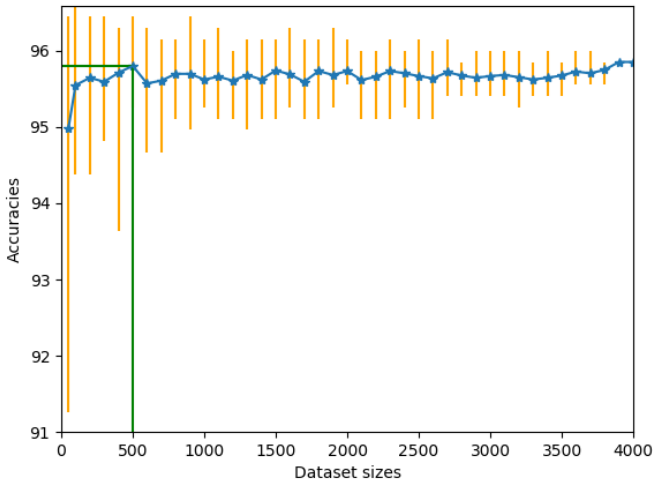
2 Test Accuracy of Bayes Classifier with Varying Training Dataset Size

2.1 Algorithm

The best model for dataset 2 was found to be **Model 5 (Bayes, different covariance)**. For different training set sizes, we averaged the test accuracy over 20 replications, and obtained the plot with standard error bars.

2.2 Observations and Plots

Figure 9



It requires a **dataset size of 500** to obtain **95.8% test accuracy**.

2.3 Inferences

With increase in dataset size, the error bars become shorter, indicating that the variance of test error decreases. Also, the test accuracy generally increases with increase in train size.

3 Accuracy of Bayes Classifier with Varying Number of Features

3.1 Algorithm

For datasets 3 and 4, we built a Bayes classifier and found the training accuracy considering 1, 2 and 3 features at a time.

3.2 Observations and Plots

3.2.1 Dataset 3

Train Size	Train error (1 feature)	Train error (2 features)	Train error (3 features)
2	0.50	0.50	0.50
3	0.67	0.67	0.67
4	**0.25**	**0.50**	**0.25**
5	**0.20**	**0.60**	**0.60**
6	0.50	0.17	0.17
7	0.43	0.14	0.14
8	0.38	0.12	0.12
9	0.33	0.22	0.22
10	0.30	0.30	0.30
50	0.24	0.14	0.14
100	0.23	0.19	0.19
500	0.21	0.16	0.16
1000	0.18	0.15	0.15
3000	0.17	0.15	0.15

3.2.2 Dataset 4

Train Size	Train error (1 feature)	Train error (2 features)	Train error (3 features)
2	0.50	0.50	0.50
3	0.67	0.67	0.67
4	0.00	0.00	0.00
5	0.00	0.00	0.00
6	0.17	0.00	0.00
7	0.14	0.14	0.57
8	**0.12**	**0.12**	**0.62**
9	**0.11**	**0.11**	**0.67**
10	0.10	0.10	0.00
50	0.14	0.12	0.06
100	0.14	0.12	0.06
500	0.12	0.06	0.06
1000	0.11	0.06	0.06
3000	0.11	0.06	0.05

The train error increased with increase in number of features for very small train sizes of **n = 4** and **n = 5** for **dataset 3**, and **n = 8** and **n = 9** for **dataset 4**.

3.3 Inferences

It is possible for a dataset that is small when compared to the number of features, the training error increases. This may happen because, with such small datasets, the model suffers heavy bias and poor generalization. Choosing a slightly different dataset may lead to a very different result. Using a small set of features by feature selection (or increasing dataset size) helps fix the problem.

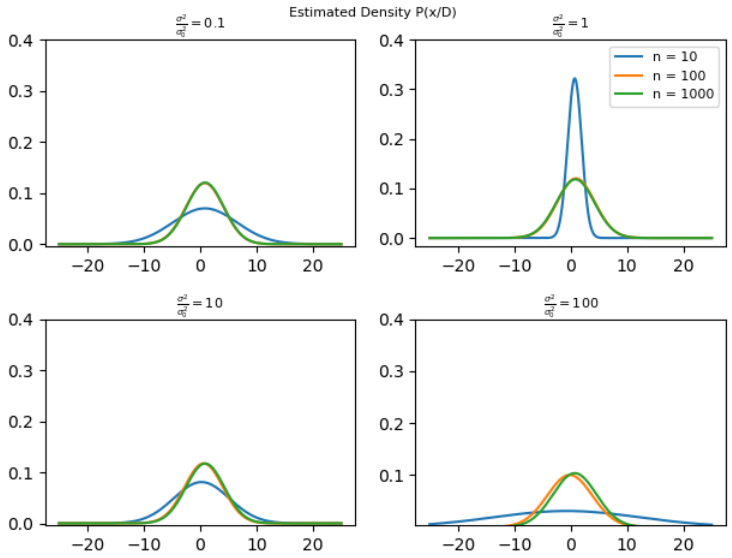
4 Bayesian Estimation of One-dimensional Gaussian Dataset

4.1 Algorithm

We used the standard formula for posterior probability for the Bayesian estimation of μ (in terms of the factor σ^2/σ_0^2), and ML estimation for estimating σ (in terms of μ).

4.2 Observations and Plots

Figure 10



4.3 Inferences

For larger values of n (100, 1000), the densities look almost the same for all the 4 cases. This is because, for large values of n, the Bayesian estimate of mean is almost same as sample mean. However, for n = 10, when σ^2 and σ_0^2 are equal, the posterior density has a very low variance. But, when σ^2 is much larger than σ_0^2, the density has a very high variance.

5 Linear Regression, Overfitting and Ridge Regression

5.1 Algorithm

We sampled 100 points, obtained a target value according to the given function, and added a noise factor to it. We performed polynomial regression by taking 10 points at a time.

5.2 Observations and Plots

5.2.1 Coefficients of polynomial for different degrees without ridge regression

We performed linear regression on training data of size of 10, varying the degrees, without regularization. As expected, the magnitude of coefficients increased with increase in the degree of polynomial due to overfitting

Degree	1	3	6	9
Coefficients	1.892	1.224	0.868	-1.151
	-0.464	8.883	22.220	159.477
		-22.295	-173.597	-3059.273
		14.043	749.704	27120.129
			-1667.473	-127424.214
			1724.313	343580.746
			-656.075	-548682.648
				511772.996
				-257144.113
				53676.031

5.2.2 Target output v/s x

Figure 11

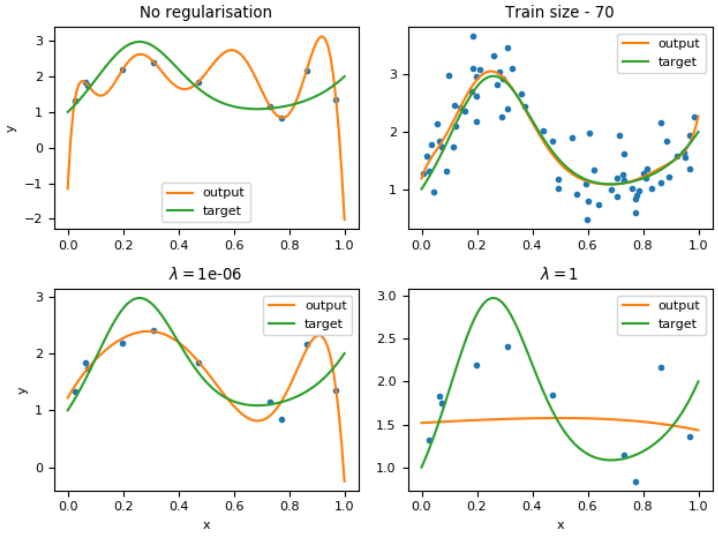
The first subplot clearly illustrates underfitting and the last one shows over-fitting.

5.2.3 Coefficients for 9th degree polynomial with different values of regularization coefficient

Coefficient	$\lambda = 10^{-6}$	$\lambda = 1$
w_0	1.219	1.518
w_1	8.481	0.180
w_2	-21.168	-0.068
w_3	35.425	-0.089
w_4	-49.196	-0.065
w_5	-30.128	-0.037
w_6	37.036	-0.016
w_7	73.314	0.003
w_8	35.173	0.005
w_9	-90.314	0.008

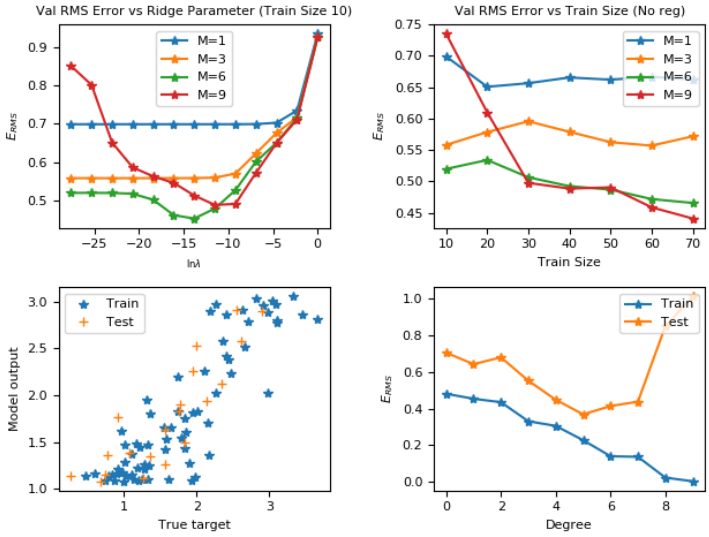
5.2.4 Analysis of overfitting in 9th degree polynomial

Figure 12



5.2.5 Root Mean Square Errors

Figure 13



5.3 Inferences

Comparing 5.2.3 with 5.2.1, coefficients for a 9th degree polynomial without regularization, we can see how much regularization helps in limiting the magnitude of the coefficients. The estimated function is also robust to noise when we control over-fitting as illustrated in 5.2.4. E_{RMS} on test data first decreases and then increases once the model is over-fitting.

6 Bias-Variance Trade-off

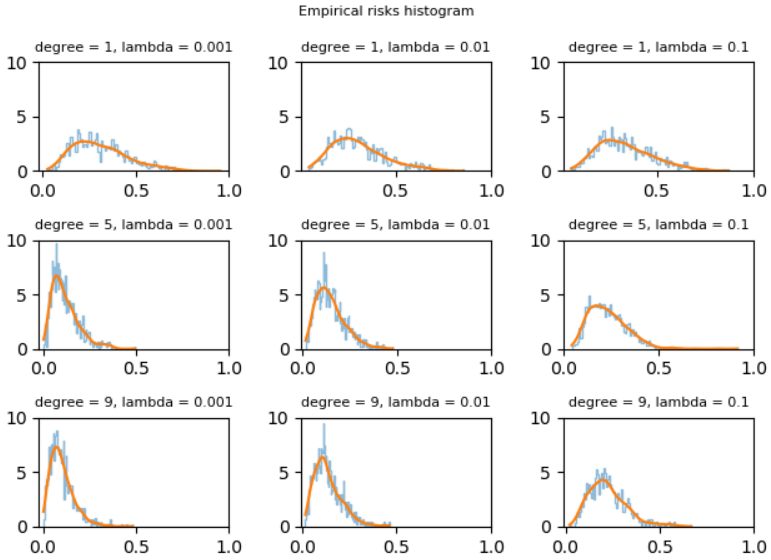
6.1 Algorithm

For each of the given degrees of polynomials and values of λ, we sampled 10 points from a uniform distribution corrupted with Gaussian noise. We performed ridge regression on the points, and stored the RMS error for 1000 iterations, and plotted a histogram of the same.

6.2 Observations and Plots

6.2.1 Empirical risks histograms

Figure 14



6.3 Inferences

As the degree of polynomial increases, the variance decreases, as we are going from an underfitting model towards an overfitting model. As the ridge parameter increases, the bias also increases, which is because we are going from an overfitting model towards an underfitting model. This illustrates the bias-variance trade-off of increasing model complexity v/s increasing ridge parameter.

7 Linear Regression with Gaussian Basis Functions

7.1 Algorithm

From the given bi-variate Gaussian data, we used k-means clustering to obtain the means of the Gaussian basis functions for different values of k. We experimented with different values for scalar σ of the basis functions. We then analyzed over-fitting for the case of $k = 60$ and controlled it using ridge regression by varying λ and by increasing train size.

7.2 Observations and Plots

7.2.1 Plot of the best models of approximated functions

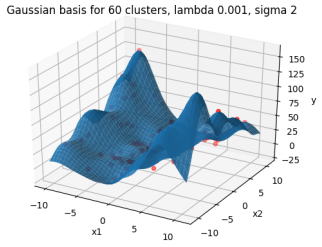


Figure 15: Dataset train100

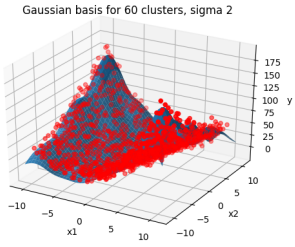


Figure 16: Dataset train1000

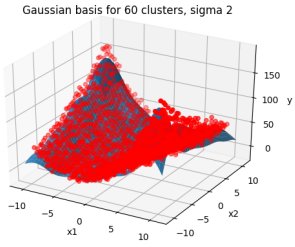
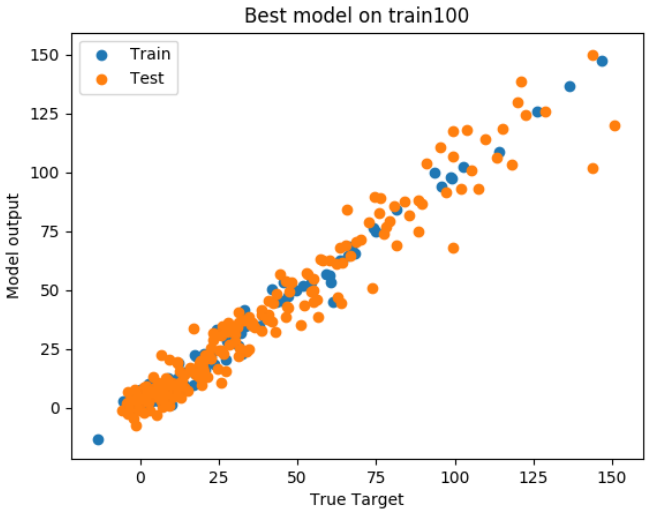


Figure 17: Dataset train2000

7.2.2 Scatter plot of target output v/s model output

Figure 18



7.2.3 E_{RMS} on training, validation and test data, $\sigma = 2$

Train Size	Clusters	λ	σ	Train E_{RMS}	Validation E_{RMS}	Test E_{RMS}
100	10	0.00	2.00	21.67	31.59	28.84
100	30	0.00	2.00	8.20	17.10	14.77
100	60	0.00	2.00	4.13	**12.69**	10.93
100	70	0.00	2.00	3.85	13.51	11.65
100	80	0.00	2.00	3.34	12.97	11.44
100	10	0.001	2.00	21.69	31.59	28.84
100	30	0.001	2.00	8.24	17.10	14.78
100	60	0.001	2.00	4.36	12.62	10.84
100	70	0.001	2.00	4.27	12.61	10.83
100	80	0.001	2.00	4.17	**12.43**	10.66
1000	10	0.00	2.00	32.75	32.99	28.82
1000	30	0.00	2.00	14.83	14.41	13.26
1000	60	0.00	2.00	7.74	8.70	7.89
1000	70	0.00	2.00	7.58	8.51	8.16
1000	80	0.00	2.00	7.29	**8.14**	7.87

7.3 Inferences

We can observe over-fitting in Figure 15 which can be controlled by increasing train size that gives smooth functions in the case of Figure 16 and 17. This is evident as Validation E_{RMS} increases with k when train size is small, without any regularization as given in 7.2.3