Report: Assignment 1
ELL409: Machine Intelligence and Learning

Siddhant, *2016MT10413*
Eshan Balachandar, *2016MT10616*
Shubham Jain, *2016CS10317*
Stuti Garg *2018VST9709*

***Abstract*— This report contains the comments, observations and results for experiments performed on the three given datasets.**

I. Fashion MNIST

A. Principal Component Analysis

Since the pixel data is rather sparse [784 dimensions], we first perform a principal component analysis. The first 84 dimensions correspond to a 10% distortion. By doing a reconstruction on the reduced data, we can see that the new image is sufficient for human classification. We will primarily be using the first 84 dimensions for further analysis.

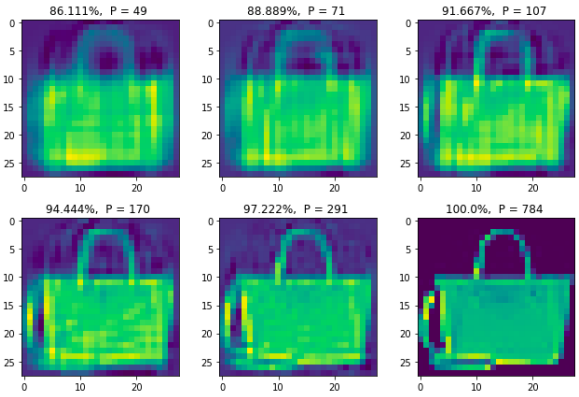

Fig. 1. Reconstruction from reduced dimension

B. Class Conditional Densities

Before performing a Bayesian Classifications on our data, we look at the class conditional densities of each of our 10 classes, with respect to the reduced features.

Across each feature, the class conditional densities are single-modal and for the later features, the Gaussian assumption seems valid. [Fig 2]

C. Gaussian Density Estimation

1) Argument: We can see that the class conditional densities are single modal across each feature and hence must be single modal across the joint density as well. Approximating them with a Gaussian [*Maximum Likelihood Estimation*] is a good idea.

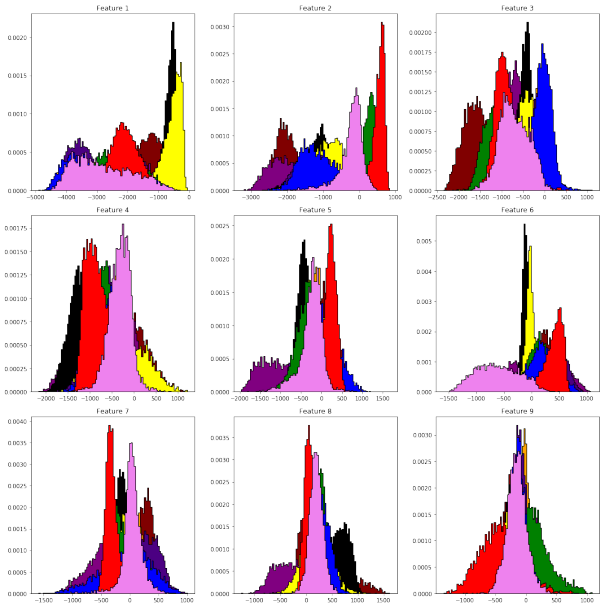

Fig. 2. Class Conditional Densities for first 9 features

2) Result: Both classifiers give peak performance for (Reduced Dimensions) P = 30
Bayes: 79.6%
Naive Bayes: 77.6%

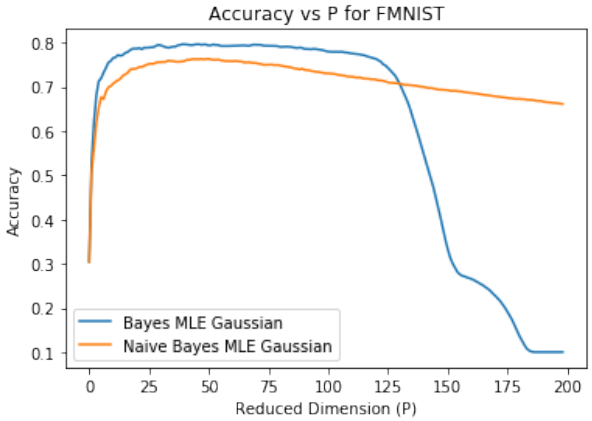

Fig. 3. Vary Dimensions

3) Bayes: As the number of features or dimensions grows, the amount of data we need to generalize accurately grows exponentially: **Curse of Dimensionality**. Therefore, our estimation using MLE is severely hurt for large dimension.
On the other hand, more dimensions also means more features and hence better estimation. In the plot, we see that the two opposing forces compete till P=125, after which the curse of dimensionality dominates.

4) Naive Bayes: The larger dimensional features have same distribution across classes, and assuming independence (as Naive Bayes does) doesn't help at all. On the other hand, more noise is accumulated with each dimension and hence the performance of Naive Bayes decreases with each extra dimension.
Also, for smaller p, Bayes performs much better than Naive Bayes, as expected, because the assumption of Independence is not valid.

D. Non-Parametric Density Estimation

Since the earlier dimensions have a skewed distribution (as seen from Fig 2), the Gaussian assumption is not very good. We can do better using non-parametric estimations, for the Bayesian Estimator.

1) Parzen Windows Estimation: For this estimation, h is a hyper-parameter which must be chosen for each class. One choice is to choose the same h for each class (which might be valid since all the classes are present in the same domain), and the chosen h will be the one that gives the best performance for the Bayes Classifier on a validation set.
Result: 85.2% for h = 128
This is the best classifier so far, probably because it can capture the skew of the class conditional densities. On the other hand, we can also choose a different h for each class, such that the choice of h leads to an estimator that maximizes the likelihood of a validation set.
Result: 67.8%
This subpar performance can be attributed to choice of hyperparameters than do not maximize Bayesian Performance, but rather the Likelihoods, which is apparently not the same thing.

2) K - Nearest Neighbor Estimation: We create a volume about each point containing K nearest Neighbors.
Result: 83.32% for K = 8
Like the parzen-window estimation, it can capture the skew of the class conditionals.

E. K - Nearest Neighbor Classifier

Result: 86.02% for K = 8
This performs on par with the non-parametric estimators above, because it works on similar principles.

F. Conclusion

The Bayesian Classifier with non-parametric estimates work the best [87.6%]. We can probably not do better without some feature engineering, and without using the pixel positions. We are only using pixel values here, without accounting for relative positions of those pixels.

II. Blood Test Dataset

A. Class Conditional Densities

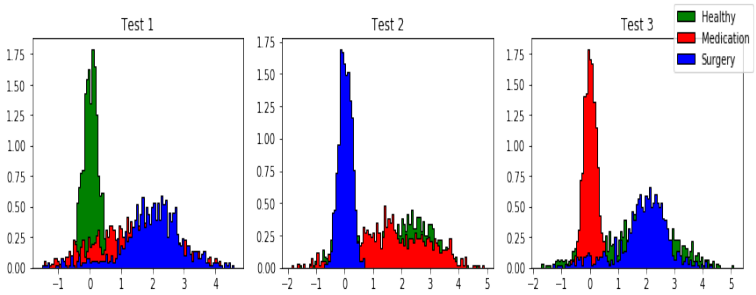

Fig. 4. Class Conditional Densities

The plot shows that along each feature, one class has a sharp normal centered at X=0, while the other two classes have similar spread out normals centered at a greater X. Each class stands out along at least one feature. It is visible that we can achieve good classification on this data set, using MLE.

B. Bayesian Classifier

We will perform a maximum likelihood estimation to find the parameters of the normal distribution (mean, variance).
Result: 90.10% Accuracy
MLE perfectly finds the class conditional densities (from the looks of it) which means that we have a perfect classifier at hand (Bayes Classifier is Optimal). Hence, most of the misclassifications are possibly due to irreducible noise.

C. Naive Bayes Classifier

We can fit a single normal across each feature for each class.
Result: 90.70% Accuracy
Surprisingly, Naive bayes performs just as well as the Bayes classifier, even though the features are NOT independent (We know this by looking at the covariance matrices). Although it may seem counter intuitive, the Naive Bayes error is not correlated with class conditional mutual information between the features[1].

D. Parzen Window Estimation

Just in case the joint distribution isn't a single normal for the classes, we can estimate the class conditional densities using non parametric techniques like Parzen WIndow.
For small h, we expect training error to be zero but a huge test error (Overfitting) but for very large h, we expect each point to be predicted with the class that has the larger prior (the trivial classifier) and hence bad test and training errors.

Result: 89.76% Accuracy
The plot comes out very similar to the over/underfitting

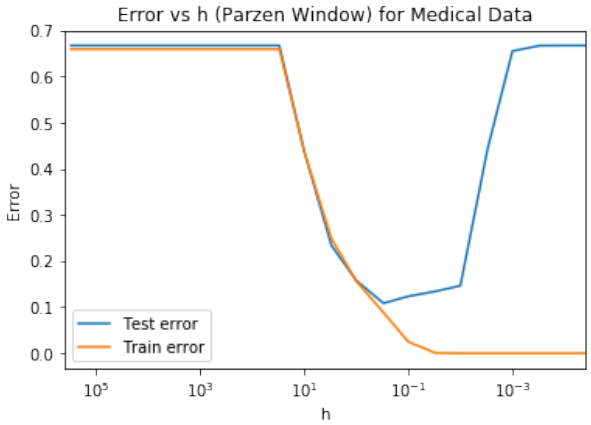

Fig. 5. Error variation with H, Medical Dataset

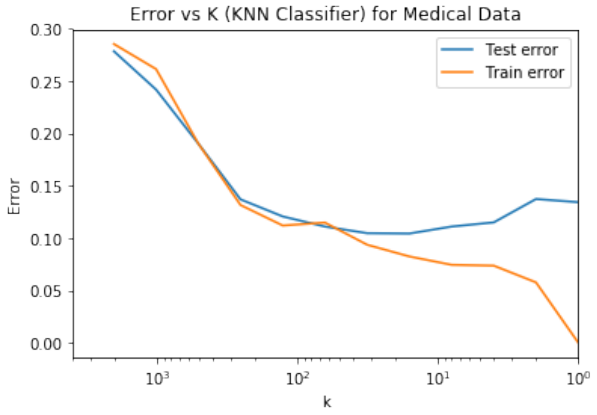

Fig. 6. Error variation with K, Medical Dataset

curves discussed in class.
This doesn't perform better than the MLE estimate, which proves that the gaussian assumption was valid.

E. K-Means Classifier

Each cluster is assigned a class based on the mode of the distribution of classes in the cluster.
Result:
60.8% for K = 3
88.6% for K = 512
Intuitively, K = 3 should work well since there are 3 classes but our resluts show otherwise. When K = 3, the three gaussian means are chosen as the centroids, but this doesn't give a good result. This is because the distance metric here is the same along all axis, and hence the decision boundary is symmetric. This is different from the gaussian decision boundary which was not symmetric because of different variance of each gaussian across different features.
Accuracy peaks for a large number of cluster centers, where this reduces to a KNN classifier.

F. K Nearest Neighbours Classifier

We can expect this classifier to give similar accuracy compared to K-Means Classifier with a large number of clusters.
Result: 88.80% Accuracy
Also, Large K leads to underfitting (output = argmax(priors)) while small K leads to overfitting. [Fig 6]

III. Train Selection Dataset

For this problem, the data is categorical, and also very incomplete, since it is impractical to expect a 100% accuracy on whether someone will board based on just the ticket details. Such a decision needs more features based on the life conditions of those people. This is clearly reflected in people with the same feature vectors both boarding and not boarding. This is the irreducible noise.
Since the data is categorical, we explored different ways of mapping them to the real line (One Hot Encoding and Ordinal Numbering)
Also, there are only 3000 data points, which is small in comparison to the number of distinct feature points.

A. Class Conditional Densities

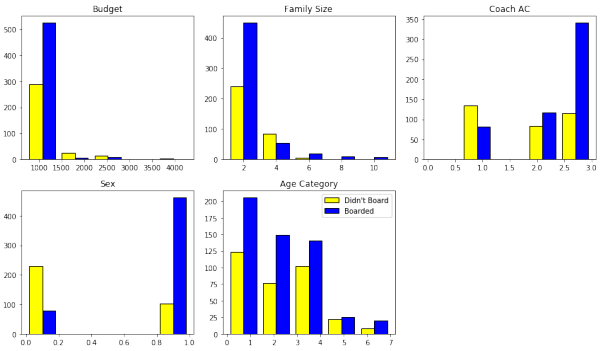

Fig. 7. Class Conditional Densities

Priors: 59% of the people board while 41% do not. So a trivial classifier can be expected to have at least 59% accuracy.
By looking at the plots, we can see that some features do not convey any useful information on their own. For example, across each age category, the percentage of people boarding compared to those who don't is essentially the same. On the other hand, features such as gender and the preferred class are very useful discriminants.
In particular, by classifying all males as having boarded and all females as not, we can achieve a 72% accuracy.

B. Bayes Classifier

We expect Bayes Classifier to give optimal performance, if we are able to correctly predict the class conditional densities.

1) Parzen-Window Estimation: For small h, this will give point estimates that are equal to the mode of the distribution at the given point for very large n. Also, we can do no

better than this estimate since the Bayes Classifier is optimal.
Result: 76.25% Accuracy

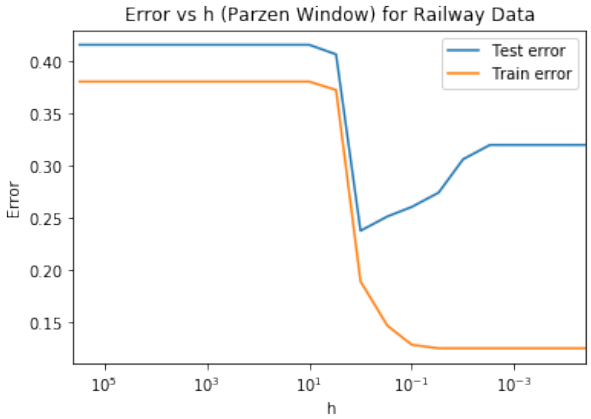

Fig. 8. Error Variation with h

For large h, the classifier classifies all data with the class with the larger prior (Boarded) and hence gives the same accuracy as the trivial classifier. [Underfitting]
For large h, the test and training accuracies are different because of the small data size (Law of large numbers doesn't hold). For small h, the classifier works great on the training data [12.4% error] but doesn't generalize well onto the test data. [Overfitting]

C. K - Means Classifier

We can expect this classifier to give the same result (and it does) as the bayes classifier with the parzen window estimate since both will classify based on the mode of the class distribution at each distinct data point.
Result: 76.25% Accuracy for 100 clusters

D. K Nearest Neighbour Classifier

Result: 74.88% for K = 30
KNN is not a good classifier for such data because for small K, there will be insufficient sampling about some data points while for large K, the classifier will start accounting for points that belong to a different cluster. Since we must fix K throughout the algorithm, the result will be sub-par.

E. Conclusion

Due to the presence of irreducible noise in the data set (as evidenced by Training error being non-zero for the bayes classifier) as well as the small size of the data set, we probably can not do any better than the 76.25% accuracy achieved by the K-Means and bayes classifiers.

References

[1] Irina Rish, Joseph Hellerstein, Jayram Thatcher, IBM Watson Research: An analysis of data characteristics that affect Naive Bayes Performance
[2] S.H. Al-Harbi, V.J Rayward Smith: Adapting K-means for supervised Classification.