Report: Assignment 2
ELL409: Machine Intelligence and Learning

Siddhant Bhakar, *2016MT10413*
Shivam Singla *2016MT10619*
Eshan Balachandar, *2016MT10616*,
Stuti Garg *2018VST9709*

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

I. River Data

We are given a regression task, with a single independent variable (the distance between point of measurement and the river bank) with the goal to predict the oxygen level at the point.

1) Data Preparation: The 10,000 samples provided are divided into a training set (8333 points) and a test set (1667 points).

2) Evaluation Criteria: : We'll try to minimize the Mean Squared Error (MSE) on the test data, which is chosen for it's mathematical convenience in optimization.

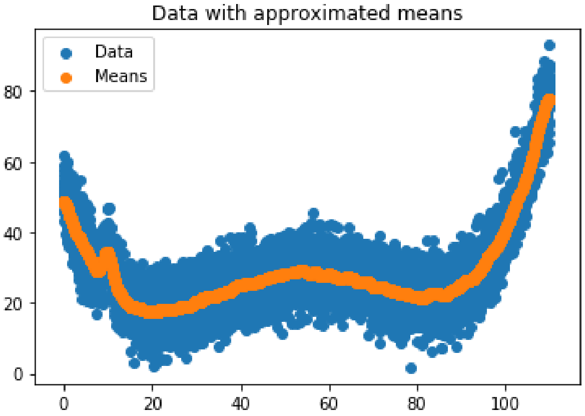

Fig. 1. River Data

3) Data: From the data, we can see that there is a lot of noise. We have roughly approximated the true value at each training point by taking mean of all y values in a small (1m) interval. This gives us an estimate of the true value at each point, along with the noise at that point.

4) Noise: Using the data points about each 1-m interval, we performed a normal-Test and consistently had large p values (mean p-value = 0.5) that imply that the noise does indeed come from a gaussian distribution.

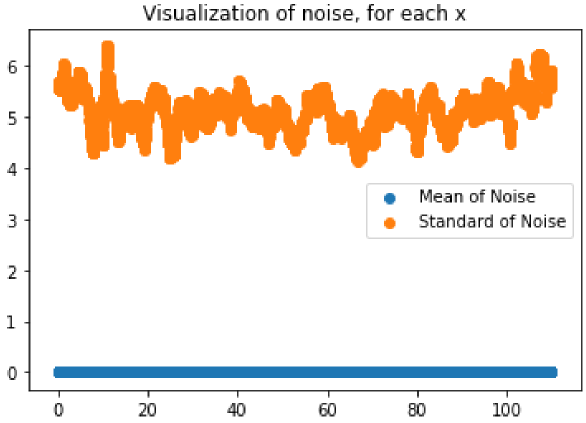

Fig. 2. Noise Visualization

The noises about each point form a gaussian, and we can see that they have 0-mean and more or less the same variance, and we can hence assume that they all must come from the same distribution (Noise is independent of x).
Since the noise is coming from a gaussian distrbution, we can conclude that Ordinary Least Squares with Squared Error Loss will give an optimal result.

5) Kernel: We want to now model the underlying distribution without the noise which looks from fig 1 polynomial. Since the data visibly has three critical points, a good polynomial fit would have at least degree four. The Kernel $\phi = \{1, x, x^2, x^3, x^4\}$ is chosen for subsequent regression.

6) Methods:

1) Normal Solution
2) Gradient Descent
3) Lagrange Interpolation

A. Gradient Descent with Momentum

The loss function with is not convex with respect to a 4-th degree polynomial kernel and hence, the algorithm gets stuck at local optimum's rather easily. This can be seen in the image, which shows different initializations of the gradient descent process converging at different losses (different local optima).

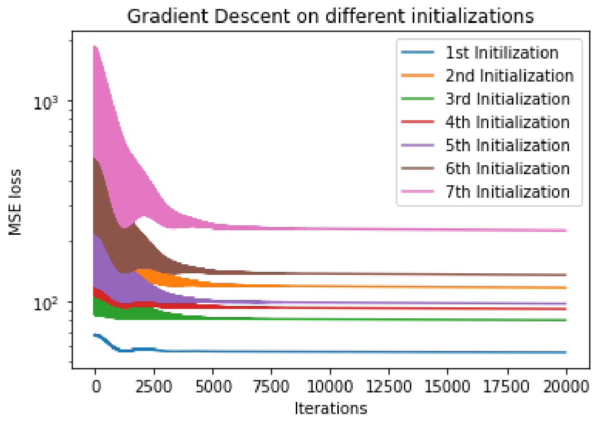

Fig. 3. Gradient Descent Failure

Hence, gradient descent cannot be relied on for properly fitting the 4th degree polynomial. We will hence turn to analytic methods instead.

B. Lagrange Interpolation

Lagrange Interpolation requires 5 points for fitting a 4th degree polynomial. We have given the X values $\{0, 25, 50, 75, 100\}$ and the y values as the mean of data around those 5 points in a 1-m interval. This leads to a excellent fit and a Mean Squared Error of 12.93

C. Normal Formula

Since the no. of points and no. features is very small, the normal formula is a feasible option, that also gives an excellent fit, with a Mean Squared Error of 12.71 on the test set.

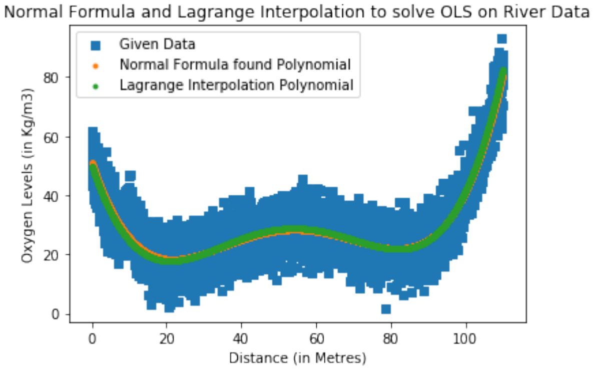

Fig. 4. River Data Prediction

D. Conclusion

We have produced a regressor that has an MSE of just 12.71, and we can presumably do no better due the irreducible noise in the dataset.

II. Blood Test Dataset

As seen in the first assignment, the three classes each correspond roughly to a Gaussian in a three dimensional vector space. We tried three different approaches to solve this multi-class classification problem.

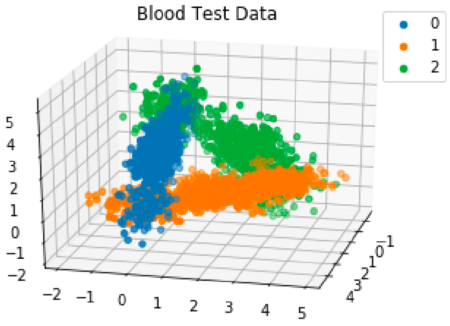

Fig. 5. Blood Test Data

A. Softmax Classifier

Linear Kernel: 80.3%
Quadratic Kernel: 89.66%
This is expected because softmax with linear kernel is optimal only when all the gaussians have the same covariance, which is not the case here. The Quadratic Kernel works for unequal covariances and hence performs as well as the Bayes Classifier sees in the previous assignment.

B. Support Vector Machine – One vs All

The One - vs - All approach to multi-class classification faces a general issue of points being classified as belong to multiple or no classes. In our testing, we marked points as ’good’ if they were classified as belonging to exactly one class and rejected the rest of the points.
We used a polynomial kernel for the soft-margin SVM.
No. of good test points = 2461/3000 (82%)
Accuracy among good points: 94.7%
Accuracy after arbitrarily breaking ties for rejected points: 85.7%

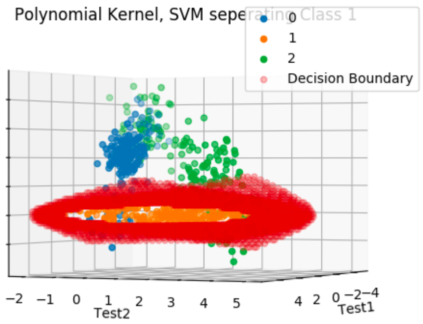

Fig. 6. Class 1 vs Rest

It can be seen that the decision boundary entirely encapsulates the data points, and we cannot do better.

C. *Support Vector Machine – One vs One*

The other approach we used was to train 3 one-vs-one soft-margin SVM classifiers and output the class that dominated the other two. While there is a theoretical possibility of cyclic orders (1 beats 2, 2 beats 3 and 3 beats 1), no such case occurs in our data, presumably due to there being no common intersection of all three clusters as seen in fig 5.
Accuracy: 89.66

III. Fashion MNIST

This is another multiclass problem with a large number of features. Unlike the previous assignment, we used applied multi-class FLDA to project the data onto a 9-dimensional subspace (9 because FLDA projects to at most no. of classes - 1 dimensions).

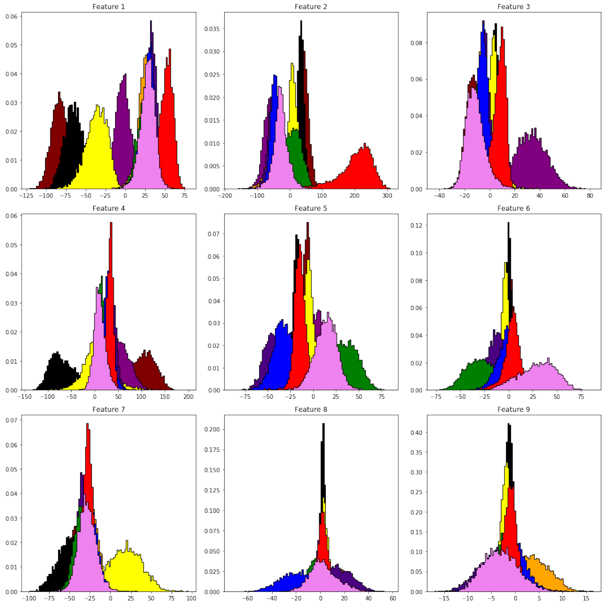

Fig. 7. FLDA projection of FMNIST Data

The features visibly form gaussians, and hence we used a softmax classifier with a quadratic kernel that is theoretically expected to be optimal for gaussian class conditional densities.
Accuracy: 82

A. *conclusion*

The accuracy is less than that achieved by a Bayes Classifer with non-parametric estimation on a PCA-reduced dataset (85%). This can be attributed to both the insufficiency of FLDA to adequately capture key characteristics in just 9 dimensions (compared to PCA, used in Assignment 1, using as many as 84 dimensions) as well as softmax not being optimal when the class conditionals are skewed gaussians (as is the case).

IV. Train Selection Dataset

We tested three classifiers (Perceptron, SVM, Logistic) on the dataset.

A. *Perceptron*

The Perceptron algorithm fails to converge since the data is not linearly separable and in fact has considerable noise.

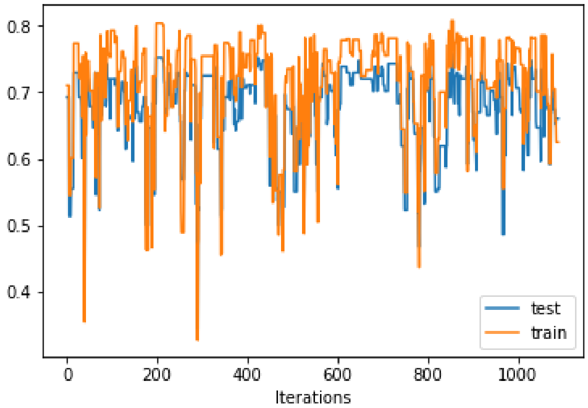

Fig. 8. Varying train, test accuracy

The accuracy can be seen to be strongly fluctuating with each iteration, with rather sharp rises and dips over single iterations. This is possibly due to the presence of rather sharp noise on either side of the classifying decision hyperplane. Best-Case Test Accuracy: 75.1

B. *Logistic Regression*

Accuracy: 76.1%

C. *SVM*

We used a Soft-Margin SVM with a very low C to account for the huge noise, and a polynomial kernel.
Accuracy: 77.8%

D. *Conclusion*

The logistic regression and soft-margin SVM classifiers manage to achieve accuracies that are on par with the Bayesian classifiers used earlier and no better because of lot of irreducible noise in the data and insufficient data points. Perceptron fails because of its assumption of linear seperability of data.