Csci 5525 – Machine Learning

Ajay Sundar Karuppasamy
Homework 1
10/08/2017

3) a. HomeVal50 is a two class dataset. **Fishers' LDA** can be used to effectively classify two normally distributed data in 1D space, with an optimal linear-discriminant. Even for data that are not normally distributed, LDA can be used to determine threshold for separation optimally.

LDA approach projects the 13-dimensional feature space on a smaller subspace k while maintaining the class discriminatory information – unlike PCA. We identify the component axes that maximize the between/and within class variance, but also reducing the computation cost by reducing the dimensions.

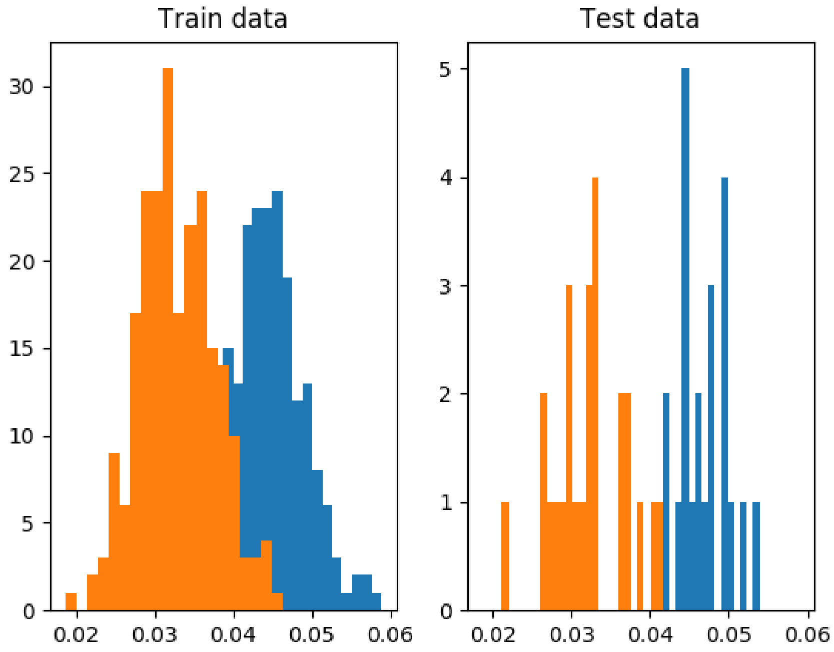
Looking at the eigen values in decreasing order (for one of the folds),

```
1.2958443590881523, 1.1266626138849171e-15,
```

we can see that the second eigen-value is (almost) 0. So, first eigen vector is informative enough to represent the data in the subspace. In fact, the second eigen-value should be 0; it's a very less value due to floating point inprecision. For k=2, there will be only one linear-discriminant.

From the plots of the projections of the HomeVal50 data in the optimal-vector ***$Sw^{-1}.(m1-m2)$*** space, it is clearly evident that the data is well-modeled by the discriminant.

Example test data plot from the 10-fold cross-validation:



3) b. No, Boston50 is a two class data, and cannot be projected to R2. Because, in **LDA the number of linear discriminants is at most (k-1)**, where k is the number of classes, since the 'between scatter matrix Sb' is the sum of k matrices with rank 1 or less.

3) c. Summary of Method:

i. Find mean vectors in *d*-dimensions for all classes in the dataset

ii. Find between and within-class scatter matrices using the below equation:

$$Sw = \sum Si \quad \text{, where} \quad Si = \sum_{x \in D_i}^{n} (x - m_i).(x - m_i)^T$$

and,

$$Sb = \sum_{i=1}^{c} N_i.(m_i - m).(m_i - m)^T$$

where, m and m_i represent the overall mean and sample mean of the individual classes. N_i is the size of the individual classes.

iii. Solving for the eigen-vectors of Sw^{-1}.Sb to obtain the linear-discriminants.

iv. Choosing the top k, eigen-vectors with the largest eigen-values

v. Transform the data to the new sample-subspace using **Y = X x W**

where **X** is the *n x d* dimensional for n samples, and Y is the new *k x d* transformed subspace samples, with **W** eigen-vector matrix of *k x d* dimensions.

vi. We use the projections in Y space, to classify using a Multivariate gaussian classifier.

vii. The apriori, mean and variance of the distribution using the train dataset and fitting the test dataset with the parameters to find the corresponding classes.

<u>Result</u>:

```
Test error-rate for fold-1: 0.0167597765363
Test error-rate for fold-2: 0.0335195530726
Test error-rate for fold-3: 0.0446927374302
Test error-rate for fold-4: 0.0614525139665
Test error-rate for fold-5: 0.0502793296089
Test error-rate for fold-6: 0.0614525139665
Test error-rate for fold-7: 0.0391061452514
Test error-rate for fold-8: 0.0335195530726
Test error-rate for fold-9: 0.0223463687151
Test error-rate for fold-10: 0.0782122905028

Mean test-error 4.01 percent
Std test-error 0.02 percent
```

4) a. Logistic regression:

It is a discriminative model, where we take P(y/x) to solve the problem

We approach to solve the problem by finding the extremum of the log-likelihood,

$$\sum_{n=1}^{N}\{ y_n \boldsymbol{w}^{\boldsymbol{T}} x_n - \log(1+\exp(\boldsymbol{w}^{\boldsymbol{T}} x_n))\}$$

We solve this optimization problem in **w** using Iterative Reweighted Least Squares algorithm, by iteratively updating the weights by solving the weighted linear least squares problem:

$$\boldsymbol{W}^{\boldsymbol{t+1}} = (\boldsymbol{X}^{\boldsymbol{T}} \boldsymbol{R}^{(\boldsymbol{t})} \boldsymbol{X})^{-1} . \boldsymbol{X}^{\boldsymbol{T}} \boldsymbol{R}^{(\boldsymbol{t})} \boldsymbol{y}$$

where **R** is a diagonal matrix of weights, initialized to $r_i = 1$.

And we update the **R** matrix, and thereby the **W** at each iteration with updating,

$$r_i^{(t)} = \frac{1}{max\{\delta, |y_i - X_i . W^t\}|}$$

b. Naive-Bayes with marginal gaussian distributions:

Bayes rule states that,

$$p(k|C_k) = \frac{p(\boldsymbol{x}|C_k)\, p(C_k)}{p(x)}$$

Naive Bayes assumes that the features are independent of each other and hence the apriori can be,

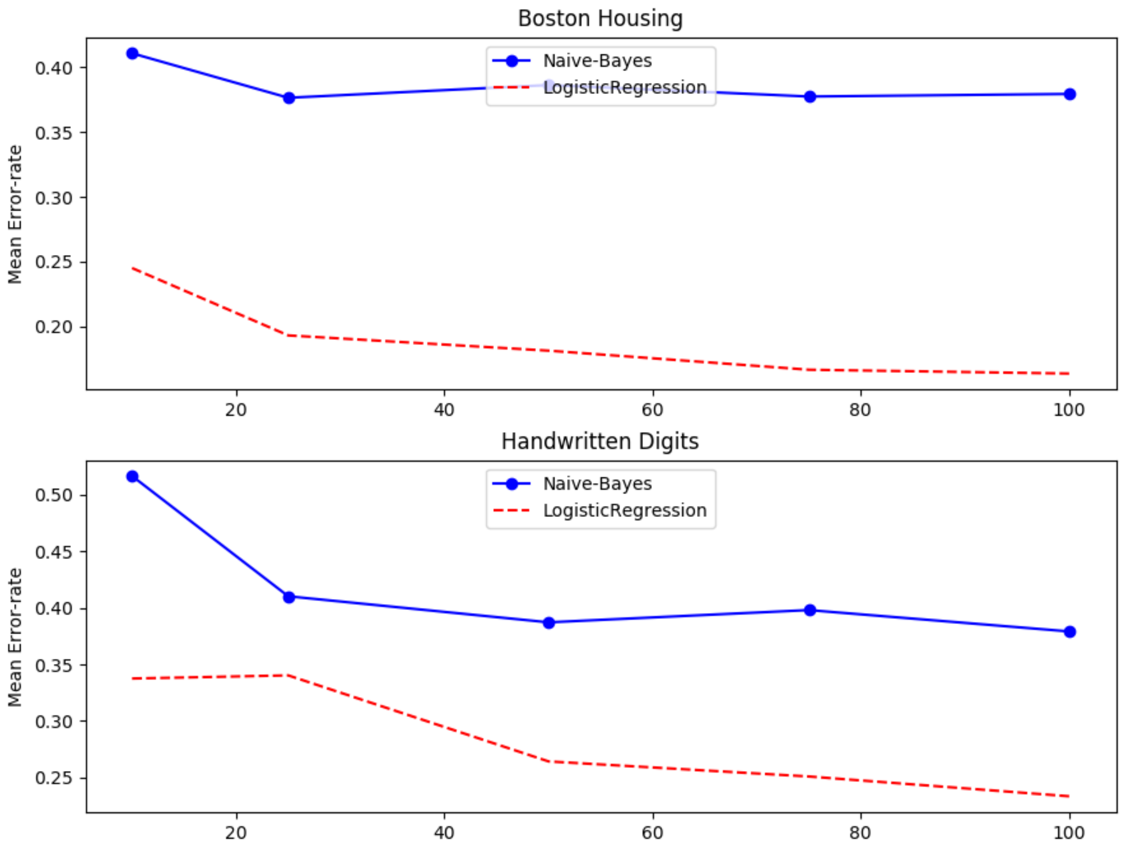
$$p(\boldsymbol{x}|C_k) = \prod_{i=1}^{p} p(x_i|C_k)$$

It is a generative model, where we are assuming model (gaussian distribution) for each univariate feature of form,

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

From the training set samples, we are finding the individual mean, variance and prior and fitting the testing sample for the best matching class as its prediction.

The above two models has been compared with the datasets as below:



---

1) **a**. When loss function is given by the squared-loss:

$$\boldsymbol{E}_{x,y}[L]=\iint L\,.\,p(y|x)\,p(x)\,dy\,dx\,,\,where\,L=(f(x)-y)^2$$

We solve this by taking the derivative inside the integral and equating to 0.
Rewriting the L and expanding the square form:

$$\int_x \boldsymbol{P}_x \int_y \boldsymbol{P}_{y|x}(f_x-\boldsymbol{E}_{y|x}+\boldsymbol{E}_{y|x}-y)^2\,dy\,dx$$

$$\int_x \boldsymbol{P}_x[\int_y \boldsymbol{P}_{y|x}(f_x-\boldsymbol{E}_{y|x})^2+(\boldsymbol{E}_{y|x}-y)^2+2.(f_x-\boldsymbol{E}_{y|x})(\boldsymbol{E}_{y|x}-y)\,dy\,]dx$$

$$\int_y P_{y|x}.\int_x \boldsymbol{P}_x(f_x-\boldsymbol{E}_{y|x})^2\,dx+\iint_{x,y}\boldsymbol{P}_{y,x}(\boldsymbol{E}_{y|x}-y)^2\,dydx+2.[0]$$

The first dot product is 1, second term is independent of f_x and last term evaluates to 0.
So, only the first term decides the solution : which gives optimal solution only when $\mathbf{f_x} = \mathbf{E}_{\mathbf{y|x}}$.

1) b. Similarly, for L as absolute error = |f(x,y) -y|,

We are looking for solution,

$$\underset{x}{\arg\min} = \iint \left|f_{(x,y)} - y\right| dx\, dy$$

Derivative of the inner objective function will give sgn(x). And for signum function to be zero, it has to be at the median - so that the integral has equal positive and negative parts to nullify it to zero.

$$\int_{-\infty}^{\infty} sgn(f_{x,y} - y) = 0$$

$$\Rightarrow \int_{-\infty}^{f_{x,y}} -sgn(f_{x,y} - y) + \int_{f_{x,y}}^{\infty} sgn(f_{x,y} - y) = 0$$

which will be 0, at optimal solution f(x,y).

---

2) P(y|C$_x$) is a probability and so the area over all (integral) y is 1.

So, the overall expected error (for including the region both y==C$_x$ and y!=C$_x$):

$$E[C_j] = \sum_{i=1:M} \int P(C_{j|x})\, p(x)\, dx = \mathbf{1}$$

By definition, given region R$_k$ is the region where conditional probability, p(y) for i=k is maximum, i.e., correct prediction.

From above,

$$\mathbf{1}_{\boldsymbol{x}} = \{ \sum_{i=1:M, i\neq j} \int P(C_{j|x}) P_x\, dx \} + \int_{i=j} P(C_{j|x}) P_x\, dx$$

Relating this for class, y=C$_j$, the above equality can be interpreted as the sum of False Negative and True Positive.

$$\Rightarrow \mathbf{1}_{\boldsymbol{C_j}} = E[C_j] = \{ E[C_{j, y\neq C_j}] + E[C_{j, y+C_j}] \} \Rightarrow \boldsymbol{FN+TP}$$

$$\sum_{i=1:M} \int P(C_{j|x}) p(x) dx = err[y=C_j] + \int_{i=j} P(C_{j|x}) P_x dx$$

$$\Rightarrow err[y=C_j] = \sum_{i=1:M} \int P(C_{j|x}) p(x) dx - \int_{i=j} P(C_{j|x}) P_x dx$$

Thus proved.