CS771A- Homework Assignment 1

Jaivardhan Kapoor - 150300

August 29, 2016

Question 1

Disance from means

Given training data $\{x_n, y_n\}_{n=1}^N$ for a classification problem with $y_n \in \{-1, +1\}$, the goal is to find $f(\boldsymbol{x})$ such that $y = sign[f(\boldsymbol{x})]$ and $f(\boldsymbol{x})$ can be written as $\sum_{n=1}^N \langle \boldsymbol{x_n}, \boldsymbol{x} \rangle + b$ where $\boldsymbol{x}$ is the test example feature vector.
Let N_+ be the class with $y = +1$ and N_- be the class with $y = -1$. Then,
$\boldsymbol{\mu_+} = \frac{\sum_{y_n \in N_+} \boldsymbol{x_n}}{|N_+|}$ and $\boldsymbol{\mu_-} = \frac{\sum_{y_n \in N_-} \boldsymbol{x_n}}{|N_-|}$
Predict $+1$ if $(\boldsymbol{x} - \boldsymbol{\mu_+})^T(\boldsymbol{x} - \boldsymbol{\mu_+}) < (\boldsymbol{x} - \boldsymbol{\mu_-})^T(\boldsymbol{x} - \boldsymbol{\mu_-})$, and vice versa.
That is,

$$y = sign[(\boldsymbol{x} - \boldsymbol{\mu_-})^T(\boldsymbol{x} - \boldsymbol{\mu_-}) - (\boldsymbol{x} - \boldsymbol{\mu_+})^T(\boldsymbol{x} - \boldsymbol{\mu_+})] = sign[f(\boldsymbol{x})]$$

Substituting the value of $\boldsymbol{\mu_+}$ and $\boldsymbol{\mu_-}$ in $f(\boldsymbol{x})$ and further simplifying, we get

$$\begin{aligned} f(\boldsymbol{x}) &= \frac{1}{N_+^2 N_-^2}\Big(N_+^2(N_-\boldsymbol{x} - \sum_{n\in N_-} \boldsymbol{x_n})^T(N_-\boldsymbol{x} - \sum_{n\in N_-} \boldsymbol{x_n}) - N_-^2(N_+\boldsymbol{x} - \sum_{n\in N_+} \boldsymbol{x_n})^T(N_+\boldsymbol{x} - \sum_{n\in N_+} \boldsymbol{x_n})\Big) \\ &= \frac{1}{N_+^2 N_-^2}\left[\Big(N_+^2 \sum_{i,j\in N_-} \boldsymbol{x_i}^T\boldsymbol{x_j} - N_-^2 \sum_{i,j\in N_+} \boldsymbol{x_i}^T\boldsymbol{x_j}\Big) + \Big(N_+^2\boldsymbol{x}^T(\sum_{j\in N_-} \boldsymbol{x_j}) - N_-^2\boldsymbol{x}^T(\sum_{j\in N_+} \boldsymbol{x_j})\Big)\right] \\ &= \frac{1}{N_+^2 N_-^2}\left(d(const.) + \boldsymbol{x}^T(\sum_{j\in N} \beta_j \boldsymbol{x_j})\right) \end{aligned}$$

where $\beta_j = \begin{cases} N_+^2 & j \in N_- \\ N_-^2 & j \in N_+ \end{cases}$
Rewriting this, we get:

$$f(\boldsymbol{x}) = \sum_{n=1}^N \alpha_n \langle \boldsymbol{x_n}, \boldsymbol{x} \rangle + b$$

where $b = \frac{\left(N_+^2 \sum_{i,j\in N_-} \boldsymbol{x_i}^T\boldsymbol{x_j} - N_-^2 \sum_{i,j\in N_+} \boldsymbol{x_i}^T\boldsymbol{x_j}\right)}{N_+^2 N_-^2}$ and $\alpha_n = \begin{cases} \frac{1}{N_-^2} & j \in N_- \\ \frac{1}{N_+^2} & j \in N_+ \end{cases}$

Question 2

Classes as Gaussians

Given 2 Gaussian distributions as $\mathcal{N}_+(\boldsymbol{x}|\mu_+, \sum)$ and $\mathcal{N}_-(\boldsymbol{x}|\mu_-, \sum)$ and test example $\boldsymbol{x}$, probability that $\boldsymbol{x}$ belongs to class $K (K \in \{+1, -1\})$ is $\mathcal{N}_\mathcal{K}(\boldsymbol{x}|\mu_K, \sum)$.// Since these are disjoint, $\mathcal{N}_+ + \mathcal{N}_- = 1$. For $\boldsymbol{x}$ to belong in class $+1$, $\frac{\mathcal{N}_+}{\mathcal{N}_-} > 1$, and vice-versa.

Taking log on both sides, $sign\Big[log\Big(\frac{\mathcal{N}_+}{\mathcal{N}_-}\Big)\Big]$ determines y. Writing both distributions in exponential form,

we get

$$f(\boldsymbol{x}) = log\Big(\frac{\mathcal{N}_+}{\mathcal{N}_-}\Big) = log\left(\frac{e^{\frac{-1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_+)^T \sum^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_+)}}{e^{\frac{-1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_-)^T \sum^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_-)}}\right)$$
$$= \frac{1}{2}\Big((\boldsymbol{x}-\boldsymbol{\mu}_-)^T \sum{}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_-) - (\boldsymbol{x}-\boldsymbol{\mu}_-)^T \sum{}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_-)\Big)$$

Note that if the covariance matrix is of the form $|\sigma|I$, this reduces to the Distance from Means problem. Expanding terms and removing constant factors from the equation, we get

$$f(\boldsymbol{x}) = 2\boldsymbol{x}^T \sum{}^{-1}(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-) + \big(\boldsymbol{\mu}_-^T \sum{}^{-1}\boldsymbol{\mu}_- - \boldsymbol{\mu}_+^T \sum{}^{-1}\boldsymbol{\mu}_+\big)$$
$$= \boldsymbol{x}^T\Big[2\sum{}^{-1}(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-)\Big] + \big(\boldsymbol{\mu}_-^T \sum{}^{-1}\boldsymbol{\mu}_- - \boldsymbol{\mu}_+^T \sum{}^{-1}\boldsymbol{\mu}_+\big)$$
$$= \boldsymbol{w}^T\boldsymbol{x} + b$$

where $\boldsymbol{w} = \Big[2\sum{}^{-1}(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-)\Big]$ and $b = \big(\boldsymbol{\mu}_-^T \sum{}^{-1}\boldsymbol{\mu}_- - \boldsymbol{\mu}_+^T \sum{}^{-1}\boldsymbol{\mu}_+\big)$

Question 3

Importance-Weighted linear regression

Let $\boldsymbol{X} = \big[\boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_N}\big]^T$ and correspondingly $Y = \big[y_1, y_2, ..., y_N\big]^T$. Also $\langle \boldsymbol{x_n}, y_n\rangle$ carries penalty c_n. Let C be a NXN diagonal matrix with $C_{ii} = c_i$.Define empirical loss function

$$\mathcal{L} = \sum_{n=1}^{N} c_n(\boldsymbol{w}^T\boldsymbol{x_n} - y_n)^2$$

where $\boldsymbol{w}$ is the parametric vector.
Our goal is to optimize $\boldsymbol{w}$ such that $\mathcal{L}$ is minimum. writing the above equation in vector form, we get

$$\mathcal{L} = (X\boldsymbol{w} - Y)^T C(X\boldsymbol{w} - Y)$$

to minimize $\mathcal{L}$, we use matrix calculus. Recall that $\nabla_x \boldsymbol{x}^T A\boldsymbol{x} = 2A\boldsymbol{x}$, and $\nabla_x \boldsymbol{b}^T\boldsymbol{x} = \boldsymbol{b}$. Applying these results on the expanded Loss function $\mathcal{L}$, we get

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = 2X^T CX\boldsymbol{w} - 2CX^T Y = 0$$

Since $\mathcal{L}$ is of quadratic form, $\mathcal{L} = 0$ gives us the global optimum(minimum).

$$X^T CX\boldsymbol{w} = CX^T Y\boldsymbol{w} = (X^T CX)^{-1}CX^T Y$$

Similarly, adding l_2 norm results in

$$\mathcal{L} = (X\boldsymbol{w} - Y)^T C(X\boldsymbol{w} - Y) + \lambda \boldsymbol{w}^T\boldsymbol{w}$$

which on minimisation gives

$$\boldsymbol{w} = (X^T CX + \lambda I_D)^{-1}CX^T Y$$

where D is the dimensions of the feature vector. Also note that we have assumed that the bias units and bias parameters have already been added to the feature vector X and the parameter vector $\boldsymbol{w}$.

Question 4

Noise as regularizer

For a standard non-regularised linear regession model, we define the loss function as

$$\mathcal{L} = \sum_{n=1}^{N} c_n(\boldsymbol{w}^T\boldsymbol{x_n} - y_n)^2$$

and minimise $\mathcal{L}$ with respect to $\boldsymbol{w}$. Note that this is identical to maximising

$$\prod_{n=1}^{N} \mathcal{N}(\boldsymbol{w}^T \boldsymbol{x} | \boldsymbol{w})$$

. This is the maximum likelihood function and translates to

$$\prod_{n=1}^{N} \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(\boldsymbol{w}^T \boldsymbol{x}_n - y_n)^2}{2\sigma_0^2}} = e^{-\frac{\Big(-(X\boldsymbol{w}-Y)^T(X\boldsymbol{w}-Y)\Big)}{2\sigma_0^2}}$$

Also, after adding prior $\frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-\boldsymbol{w}^T\boldsymbol{w}}{2\sigma^2}}$ the MAP function becomes

$$e^{\frac{\Big(-(X\boldsymbol{w}-Y)^T(X\boldsymbol{w}-Y)\Big)}{\sigma_0^2} + \frac{-\boldsymbol{w}^T\boldsymbol{w}}{2\sigma^2}}$$

Taking logarithm of the MAP function and maximizing this leads to the empirical loss function

$$\mathcal{L} = (X\boldsymbol{w}-Y)^T C (X\boldsymbol{w}-Y) + \lambda \boldsymbol{w}^T \boldsymbol{w}$$

where $\lambda = \frac{\sigma_0^2}{\sigma^2}$

As we have already seen t=in the previous question, this is the optimisation function of regularised Linear Regression and thus the prior in this case acts as the regulariser, befitting our intution.

Question 5

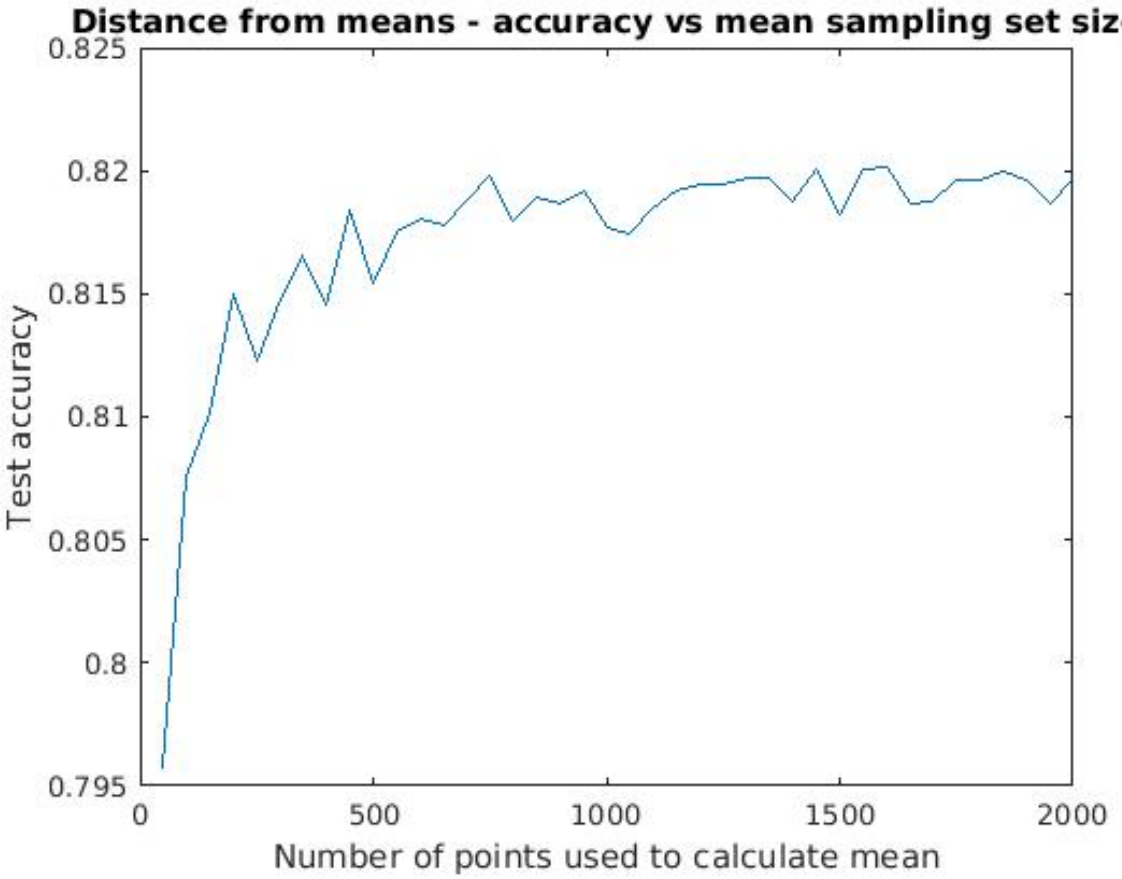
Decision trees for regression

We basically have to make our selection purer as we go down the tree. Suppose X is the feature vector matrix with rows as feature vector(which may consist of real valued features as well as labeled features). If all of the features are labeled features, we split the data according to the feature which on splitting reduces the variance of the data the most. That is, the entropy, or randomness of the real valued prediction is interpreted by its variance. So the feature $\mathcal{F}$ which minimises the variance of the resulting split subclasses of the data the most is selected for that node.
If the feature is a real valued one, we calculate a boundary value for the values of that feature, for which the variance is minimized for the resulting subclasses of X. In this way we tackle splitting real valued features as well as discrete valued ones, and complete the tree(doing pruning of branches either during the proces or after).
This is just a crude algorithm, and the accuracy of it I can barely predict. However, This is one way of "purifying" the features.

Question 6

Programming assignment- Distance from means classifier on MNIST dataset

This question requires a programming element. The program is written in MATLAB. The goal is to use the *Distance From Means* algorithm, as discussed in class to classify handwritten digits as a part of the MNIST dataset provided. The code is available on this Github repository.
The algorithm is a primitive one that uses the concept of representing one class with the mean of its constituents, and predicts new datapoints' classes by computing euclidean distances from the means as a measure of the similarity from the classes. The size of subset of each class used to compute the mean is varied and the corresponding test accuracy is plotted against it, as shown in the following figure.



We notice that as the mean sampling size increases, the performance, on average, increases(barring some abrupt fluctuations due to random sampling,which have been smoothed out by averaging accuracy over multiple iterations). This may be explained on the basis that the increased number of data points better approximate the underlying central tendency of the class. As a result, better results are obtained when the sampling set is larger.