Indian Institute of Technology Kanpur
CS771 Introduction to Machine Learning, 2017-18-a

QUESTION 1

Assignment Number: 1
Student Name: Akshat Jindal
Roll Number: 150075
Date: September 10, 2017

We realise that for calculating the decision boundary, the decision boundary is basically the line such that all points on the line are equidistant from the two means given to us, i.e (1,0) and (0,1) where distance is defined by the metric in the assignment.

Part1 :

Mathematical Expression:
Let μ_1 be (1,0), μ_2 be (0,1). Let $\mathbf{z}, \in \mathbb{R}^2$ be an arbitrary point.

$$<(z-\mu_1), U(z-\mu_1)> \;=\; <(z-\mu_2), U(z-\mu_2)> \tag{1}$$

$$\begin{bmatrix} z_1-1 & z_2 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} z_1-1 \\ z_2 \end{bmatrix} = \begin{bmatrix} z_1 & z_2-1 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2-1 \end{bmatrix} \tag{2}$$

$$\Rightarrow z_2 = 3z_1 - 1; \tag{3}$$

Part2 :

Mathematical Expression:
Let μ_1 be (1,0), μ_2 be (0,1). Let $\mathbf{z}, \in \mathbb{R}^2$ be an arbitrary point.

$$<(z-\mu_1), U(z-\mu_1)> \;=\; <(z-\mu_2), U(z-\mu_2)> \tag{4}$$

$$\begin{bmatrix} z_1-1 & z_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} z_1-1 \\ z_2 \end{bmatrix} = \begin{bmatrix} z_1 & z_2-1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2-1 \end{bmatrix} \tag{5}$$

$$\Rightarrow z_1 = 1/2; \tag{6}$$

Figure 1: Learning with Prototypes: Left figure is for Part 1, right figure is for Part 2

Indian Institute of Technology Kanpur
CS771 Introduction to Machine Learning, 2017-18-a

Assignment Number: 1
Student Name: Akshat Jindal
Roll Number: 150075
Date: September 10, 2017

QUESTION 2

Designing the Likelihood

Before beginning with the solution, we make the assumption that the training data points given to us are i.i.d Random Variables. This allows us to express the probability of the response vector $y \in \mathbb{R}^d$ as the product of the probability of the individual responses. i.e :

$$P(y|X,w) = \prod_{i=1}^{n} P(y^i|x^i,w) \tag{7}$$

Since we only have to model the responses, we deal with conditional probabilities only.We know from previous experience in the class that the given loss function term arises from a Gaussian Likelihood with mean $= <w,x>$.

Thus, for any constant σ:

$$y^i \sim N(< w,x^i >, \sigma^2) \tag{8}$$

For k as a constant:

$$P(y|X,w) = \prod_{i=1}^{n} P(y^i|x^i,w) \tag{9}$$

$$P(y|X,w) = \prod_{i=1}^{n} k * e^{-(y^i - <w,x^i>)^2/(2\sigma^2)} \tag{10}$$

$$\tag{11}$$

Taking log both sides, the Negative Likelihood Objective becomes:

$$NLL(w) = 1/2\sigma^2 * \sum_{i=1}^{n}((y^i - < w,x^i >)^2) \tag{12}$$

Designing the Prior Distribution

This designing can be thought of as very intuitive. Since our feasible region is only thos w such that their l_2 norm $\leqslant\ r$, we come up with this distribution:

$$P(w) = \begin{cases} \alpha \text{ if } \|w\|_2 \leqslant\ r \\ 0 \text{ if } o.w \end{cases}$$

To make the above a legal probability distro, α is such that $\int_w \alpha = 1\ \forall w$ such that $\|w\|_2 \leqslant\ r$. Let's calculate the MAP estimate to show that this indeed works.

The MAP estimate

$$P(w|X,y) = (P(y|X,w) * P(w))/P(y) \tag{13}$$

Thus clearly:

$$MAP\ Objective = MLE\ Objective + \log P(w) \tag{14}$$

$$Negative\ MAP\ Objective = NLL(w) - \log P(w) \tag{15}$$

$$w_{\hat{MAP}} = argmin\{NLL(w) - \log P(w)\} \tag{16}$$

$$\tag{17}$$

Explanation

Looking at Equation 17, we see that because of the way in which P(w) is defined, $\forall$ w(s) s.t $\|w\|_2 > r$, P(w) becomes 0. This causes the $\log P(w)$ term to tend to $-\infty$. This causes the Negative MAP Objective to tend to $+\infty$ and thus such w(s) will never serve as potential candidates for $w_{\hat{MAP}}$. As for the w(s) s.t $\|w\|_2 \leqslant r$, the $\log P(w)$ term is a constant and the $w_{\hat{MAP}}$ essentially becomes $w_{\hat{MLE}}$. Thus this designing of Likelihood and Prior will lead to the same optimisation problem :)

Indian Institute of Technology Kanpur
CS771 Introduction to Machine Learning, 2017-18-a

Assignment Number: 1
Student Name: Akshat Jindal
Roll Number: 150075
Date: September 10, 2017

QUESTION 3

Before beginning, we set up the following notation for the objective function:

$$L(w) = \sum_{i=1}^{n}((y^i - < w, x^i >)^2) + \sum_{j=1}^{n} \alpha_j(w_j^2) \tag{18}$$

Designing the Likelihood

We make the assumption that the training data points given to us are i.i.d Random Variables. This allows us to express the probability of the response vector $y \in \mathbb{R}^d$ as the product of the probability of the individual responses. i.e :

$$P(y|X, w) = \prod_{i=1}^{n} P(y^i|x^i, w) \tag{19}$$

Since we only have to model the responses, we deal with conditional probabilities only.We know from previous experience in the class that the given loss function term arises from a Gaussian Likelihood with mean = $<w, x>$.

Thus, for any constant σ:

$$y^i \sim N(< w, x^i >, \sigma^2) \tag{20}$$

For k as any constant:

$$P(y|X, w) = \prod_{i=1}^{n} P(y^i|x^i, w) \tag{21}$$

$$P(y|X, w) = \prod_{i=1}^{n} k * e^{-(y^i - <w, x^i>)^2/(2\sigma^2)} \tag{22}$$

$$\tag{23}$$

Taking log both sides, the Negative Likelihood Objective becomes:

$$NLL(w) = 1/2\sigma^2 * \sum_{i=1}^{n}((y^i - < w, x^i >)^2) \tag{24}$$

Designing the prior

Now, we need to create a prior which will lead to the *feature-regularized* regularization term. We propose that $w \sim$ A MultiVariate Gaussian Distribution with mean $= 0 \in \mathbb{R}^d$ and $\sum$ as the Covariance matrix:

$$w \sim N(0, \sum) \tag{25}$$

Where, $\sum$ is the following diagonal co-variance matrix:

$$D = \begin{bmatrix} (\beta_1)^{-1} & & \\ & \ddots & \\ & & (\beta_d)^{-1} \end{bmatrix}$$

Thus, for k as any constant:

$$P(w) = k * e^{\frac{-(w^T . \sum^{-1} . w)}{2}} \tag{26}$$

The MAP estimate

$$P(w|X, y) = (P(y|X, w) * P(w))/P(y) \tag{27}$$

Thus clearly:

$$MAP\ Objective = MLE\ Objective + \log P(w) \tag{28}$$

$$\Rightarrow w_{\hat{MAP}} = argmax\{\ -1/2\sigma^2 * \sum_{i=1}^{n}((y^i - < w, x^i >)^2) + \log k * e^{\frac{-(w^T . \sum^{-1} . w)}{2}}\} \tag{29}$$

$$\Rightarrow w_{\hat{MAP}} = argmax\{\ -1/2\sigma^2 * \sum_{i=1}^{n}((y^i - < w, x^i >)^2) - \frac{(w^T . \sum^{-1} . w)}{2}\} \tag{30}$$

$$\Rightarrow w_{\hat{MAP}} = argmax\{\ -1/2\sigma^2 * \sum_{i=1}^{n}((y^i - < w, x^i >)^2) - \sum_{i=1}^{d}\frac{\beta_i w_i^2}{2}\} \tag{31}$$

$$\tag{32}$$

Stowing away the constants into β_i terms, and removing the minus sign overall:

$$\hat{w}_{MAP} = argmin\{\ \sum_{i=1}^{n}((y^i - < w, x^i >)^2) + \sum_{i=1}^{d}\alpha_i w_i^2\} \tag{33}$$

$$\tag{34}$$

Thus, the map estimate $\hat{w_{fr}}$(mentioned in question) is the map estimate of our model.

Closed form solution

$$L(w) = \sum_{i=1}^{n}((y^i - < w, x^i >)^2) + \sum_{j=1}^{n}\alpha_j(w_j^2) \tag{35}$$

$$\nabla(L(w)) = 2 * \sum_{i=1}^{n}(y^i - < w, x^i >)\nabla(y^i - < w, x^i >) + \nabla\sum_{j=1}^{n}\alpha_j(w_j^2) \tag{36}$$

$$\nabla(L(w)) = 2 * \sum_{i=1}^{n}(y^i - < w, x^i >)(-x^i) + \nabla\sum_{j=1}^{n}\alpha_j(w_j^2) \tag{37}$$

Let's call A:

$$A = \begin{bmatrix} (\alpha_1)^{-1} & & \\ & \ddots & \\ & & (\alpha_d)^{-1} \end{bmatrix}$$

Also, we call X as the design matrix, i.e X is an N x D matrix, one column for each feature. Putting $\nabla(L(w)) = 0$, we get:

$$X^T y = X^T X w + A w \tag{38}$$

$$\Rightarrow w = (X^T X + A)^{-1} X^T y \tag{39}$$

Thus derived.

□

Indian Institute of Technology Kanpur
CS771 Introduction to Machine Learning, 2017-18-a

QUESTION

4

Assignment Number: 1
Student Name: Akshat Jindal
Roll Number: 150075
Date: September 10, 2017

We knoe the Crammer-Singer formulation $(P1)$ for a single machine learner for multi-classification is

$$\left\{\widehat{\mathbf{W}}, \{\hat{\xi}_i\}\right\} = \underset{\mathbf{W},\{\xi_i\}}{\arg\min} \sum_{k=1}^{K} \left\|\mathbf{w}^k\right\|_2^2 + \sum_{i=1}^{n} \xi_i$$
$$\text{s.t. } \left\langle \mathbf{w}^{y^i}, \mathbf{x}^i \right\rangle \geq \left\langle \mathbf{w}^k, \mathbf{x}^i \right\rangle + 1 - \xi_i, \forall i, \forall k \neq y^i \qquad (P1)$$
$$\xi_i \geq 0, \text{ for all } i$$

On the other hand, (P2) is :

$$\widehat{\mathbf{W}} = \underset{\mathbf{W}}{\arg\min} \sum_{k=1}^{K} \left\|\mathbf{w}^k\right\|_2^2 + \sum_{i=1}^{n} \ell_{\text{cs}}(y^i, \boldsymbol{\eta}^i) \qquad (P2),$$

where $\boldsymbol{\eta}^i = \left\langle \mathbf{W}, \mathbf{x}^i \right\rangle$ and

$$\ell_{\text{cs}}(y^i, \boldsymbol{\eta}^i) = [1 + \max_{k \neq y} \boldsymbol{\eta}_k^i - \boldsymbol{\eta}_y^i]_+$$

We shuffle around the constraint conditions of (P1) as follows:

$$\left\langle \mathbf{w}^{y^i}, \mathbf{x}^i \right\rangle \geq \left\langle \mathbf{w}^k, \mathbf{x}^i \right\rangle + 1 - \xi_i, \forall i, \forall k \neq y^i$$
$$\Rightarrow \xi_i \geq 1 + \left\langle \mathbf{w}^k, \mathbf{x}^i \right\rangle - \left\langle \mathbf{w}^{y^i}, \mathbf{x}^i \right\rangle \forall i, \forall k \neq y^i$$
$$\Rightarrow \xi_i \geq 1 + \boldsymbol{\eta}_k^i - \boldsymbol{\eta}_y^i \quad \forall i, \forall k \neq y^i$$
$$\Rightarrow \xi_i \geq 1 + \max_{k \neq y} \boldsymbol{\eta}_k^i - \boldsymbol{\eta}_y^i \quad \forall i$$

But we know, that $\xi_i \geq 0, \text{ for all } i$, thus

$$\Rightarrow \xi_i \geq \max\{1 + \max_{k \neq y} \boldsymbol{\eta}_k^i - \boldsymbol{\eta}_y^i, 0\} \quad \forall i$$

But this means:

$$\xi_i >= \begin{cases} 0 & \text{if } \boldsymbol{\eta}_y^i - \max_{k \neq y} \boldsymbol{\eta}_k^i > 1 \\ 1 + \max_{k \neq y} \boldsymbol{\eta}_k^i - \boldsymbol{\eta}_y^i & o.w \end{cases}$$

Which implies that: $\xi_i >= \ell_{\text{cs}}(y^i, \boldsymbol{\eta}^i) = [1 + \max_{k \neq y} \boldsymbol{\eta}_k^i - \boldsymbol{\eta}_y^i]_+$
So, we can rewrite (P1) as :

$$\left\{\widehat{\mathbf{W}}, \{\hat{\xi}_i\}\right\} = \underset{\mathbf{W},\{\xi_i\}}{\arg\min} \sum_{k=1}^{K} \left\|\mathbf{w}^k\right\|_2^2 + \sum_{i=1}^{n} \xi_i$$
$$\text{s.t. } \xi_i >= \ell_{\text{cs}}(y^i, \boldsymbol{\eta}^i) = [1 + \max_{k \neq y} \boldsymbol{\eta}_k^i - \boldsymbol{\eta}_y^i]_+ \forall i \qquad (P1)$$

Now realize that since we want to minimise our objective function, in any optimal solution $\left\{\mathbf{W}^0, \{\xi_i^0\}\right\}$ for (P_1), $\xi_i^0 = \ell_{\text{cs}}(y^i, \boldsymbol{\eta}^i) = [1 + \max_{k \neq y} \boldsymbol{\eta}_k^i - \boldsymbol{\eta}_y^i]_+ \ \forall i$

This is because if we take any $\xi_i^1 > \ell_{\text{cs}}(y^i, \boldsymbol{\eta}^i) = [1 + \max_{k \neq y} \boldsymbol{\eta}_k^i - \boldsymbol{\eta}_y^i]_+ \ \forall i$, then we can replace $\xi_i^1 \ \forall i$ by ξ_i^0 in the objective function to squeeze down $\sum_{k=1}^K \left\|\mathbf{w}^k\right\|_2^2 + \sum_{i=1}^n \xi_i^1$ to $\sum_{k=1}^K \left\|\mathbf{w}^k\right\|_2^2 + \sum_{i=1}^n \xi_i^0$

Thus, no matter what $\left\{\mathbf{W}^0\right\}$ you get, the ξ_i^0 will always be $\ell_{\text{cs}}(y^i, \boldsymbol{\eta}^i) = [1 + \max_{k \neq y} \boldsymbol{\eta}_k^i - \boldsymbol{\eta}_y^i]_+ \ \forall i$.
Thus solving P1 is equivalent to solving :

$$\widehat{\mathbf{W}} = \arg\min_{\mathbf{W}} \ \sum_{k=1}^K \left\|\mathbf{w}^k\right\|_2^2 + \sum_{i=1}^n [1 + \max_{k \neq y} \boldsymbol{\eta}_k^i - \boldsymbol{\eta}_y^i]_+$$

which is essentially P2 only.
Thus both the optimization problems are infact the same.

□

Indian Institute of Technology Kanpur
CS771 Introduction to Machine Learning, 2017-18-a

QUESTION 5

Assignment Number: 1
Student Name: Akshat Jindal
Roll Number: 150075
Date: September 10, 2017

Aim

To show that g is a member of the subdifferentials of f at w. i.e.
To prove $\forall w^{'} \in \mathbb{R}^d : f(w^{'}) - f(w) \geq \langle g, w^{'} - w \rangle$
We will prove this inequality by proving it for 1 arbitrary $i \in [1, N]$ and then putting summation sign on both sides of the equation.
Thus, our objective becomes proving, for arbitrary $w^{'} \in \mathbb{R}^d$:

$$[1 - y^i \langle w^{'}, x^i \rangle]_+ - [1 - y^i \langle w, x^i \rangle]_+ \geq \langle h^i, w^{'} - w \rangle \tag{40}$$

For convenience, we define the function $\forall a, b \in \mathbb{R}$:

$$[1 - a.b]_+ = \begin{cases} 0 \text{ if } a.b \geq 1 \\ 1 - a.b \text{ if } a.b < 1 \end{cases}$$

Case 1: $y^i \langle w, x^i \rangle \geq 1$

Subcase A: $y^i \langle w^{'}, x^i \rangle \geq 1$

LHS:

$$[1 - y^i \langle w^{'}, x^i \rangle]_+ - [1 - y^i \langle w, x^i \rangle]_+ \tag{41}$$
$$= 0 - 0 \text{ (} by\ definition\ above \text{)} \tag{42}$$
$$= 0 \tag{43}$$

RHS:

$$\langle h^i, w^{'} - w \rangle \tag{44}$$
$$= 0 \text{ (} by\ definition\ of\ h^i \text{)} \tag{45}$$

Thus, LHS >= RHS.

Subcase B: $y^i \langle w^{'}, x^i \rangle < 1$

LHS:

$$[1 - y^i \langle w^{'}, x^i \rangle]_+ - [1 - y^i \langle w, x^i \rangle]_+ \tag{46}$$
$$= 1 - y^i \langle w^{'}, x^i \rangle - 0 \text{ (} by\ definition\ above \text{)} \tag{47}$$
$$\tag{48}$$

Note: LHS > 0 as $y^i \langle w^{'}, x^i \rangle < 1$
RHS:

$$\langle h^i, w^{'} - w \rangle \tag{49}$$
$$= 0 \text{ (} by\ definition\ of\ h^i \text{)} \tag{50}$$

Thus, LHS >= RHS.

Case 2: $y^i\langle w, x^i\rangle < 1$

Subcase A: $y^i\langle w^{'}, x^i\rangle \geq 1$

LHS:

$$[1 - y^i\langle w^{'}, x^i\rangle]_+ - [1 - y^i\langle w, x^i\rangle]_+ \tag{51}$$
$$= 0 - (1 - y^i\langle w, x^i\rangle) \text{ (} by\ definition\ above \text{)} \tag{52}$$
$$= y^i\langle w, x^i\rangle - 1 \tag{53}$$

RHS:

$$\langle h^i, w^{'} - w\rangle \tag{54}$$
$$= \langle -y^i x^i, w^{'} - w\rangle \text{ (} by\ definition\ of\ h^i \text{)} \tag{55}$$
$$= y^i\langle x^i, w\rangle - y^i\langle x^i, w^{'}\rangle \text{ [} y^i\ is\ constant\ and\ dot\ product\ can\ be\ split \text{]} \tag{56}$$
$$= y^i\langle w, x^i\rangle - y^i\langle w^{'}, x^i\rangle \tag{57}$$

So, we observe:

$$y^i\langle w^{'}, x^i\rangle \geq 1\ [We\ are\ in\ subcase\ A] \tag{58}$$
$$\Rightarrow -y^i\langle w^{'}, x^i\rangle <= -1 \tag{59}$$
$$\Rightarrow y^i\langle w, x^i\rangle - y^i\langle w^{'}, x^i\rangle <= y^i\langle w, x^i\rangle - 1 \tag{60}$$
$$\Rightarrow RHS <= LHS \tag{61}$$

Thus, LHS >= RHS.

Subcase B: $y^i\langle w^{'}, x^i\rangle < 1$

LHS:

$$[1 - y^i\langle w^{'}, x^i\rangle]_+ - [1 - y^i\langle w, x^i\rangle]_+ \tag{62}$$
$$= y^i\langle w, x^i\rangle - y^i\langle w^{'}, x^i\rangle \text{ (} by\ definition\ above \text{)} \tag{63}$$
$$\tag{64}$$

RHS:

$$\langle h^i, w^{'} - w\rangle \tag{65}$$
$$= y^i\langle x^i, w\rangle - y^i\langle x^i, w^{'}\rangle \text{ (} by\ definition\ of\ h^i \text{)} \tag{66}$$
$$= y^i\langle w, x^i\rangle - y^i\langle w^{'}, x^i\rangle \tag{67}$$

Thus, LHS >= RHS.
Thus, for all cases, we proved that :

$$[1 - y^i\langle w^{'}, x^i\rangle]_+ - [1 - y^i\langle w, x^i\rangle]_+ \geq \langle h^i, w^{'} - w\rangle \tag{68}$$

Thus:

$$\sum_{i=1}^{n}[1-y^i\langle w^{'},x^i\rangle]_+ - \sum_{i=1}^{n}[1-y^i\langle w,x^i\rangle]_+ \geq \sum_{i=1}^{n}\langle h^i, w^{'}-w\rangle \tag{69}$$

$$\sum_{i=1}^{n}[1-y^i\langle w^{'},x^i\rangle]_+ - \sum_{i=1}^{n}[1-y^i\langle w,x^i\rangle]_+ \geq \langle\sum_{i=1}^{n} h^i, w^{'}-w\rangle \tag{70}$$

$$\sum_{i=1}^{n}[1-y^i\langle w^{'},x^i\rangle]_+ - \sum_{i=1}^{n}[1-y^i\langle w,x^i\rangle]_+ \geq \langle g, w^{'}-w\rangle \tag{71}$$

$$f(w^{'}) \ - f(w) \geq \langle g, w^{'}-w\rangle \tag{72}$$

Since the $w^{'}$ we chose was arbitrary, the result has been proven $\forall w^{'} \in \mathbb{R}^d$

□

Indian Institute of Technology Kanpur
CS771 Introduction to Machine Learning, 2017-18-a

Assignment Number: 1
Student Name: Akshat Jindal
Roll Number: 150075
Date: September 10, 2017

QUESTION 6

Part1

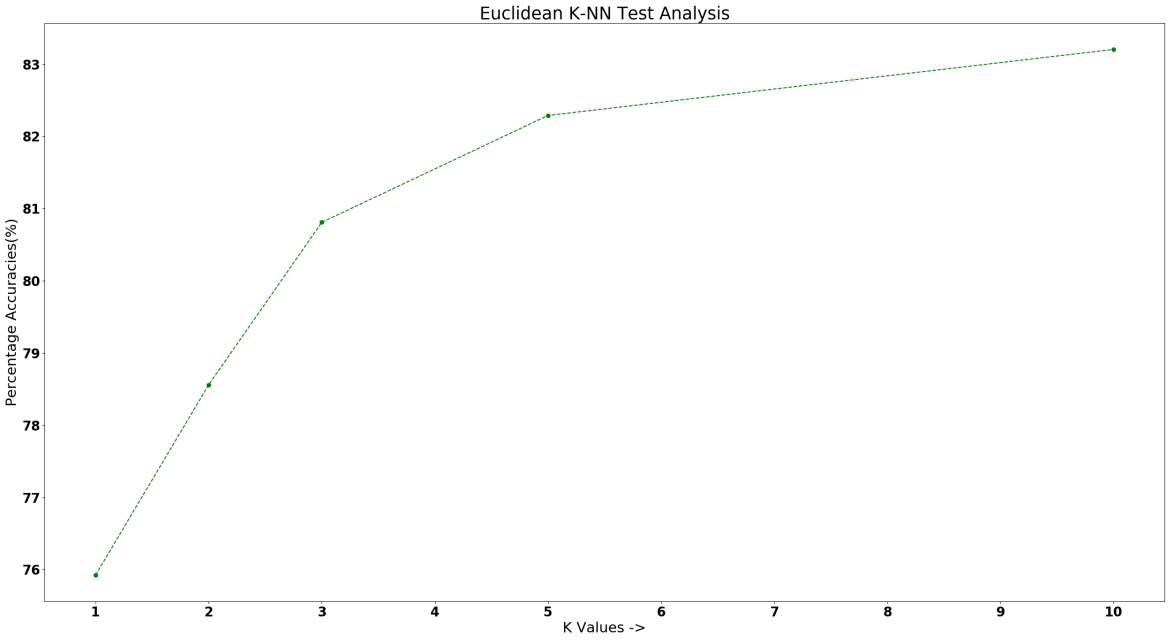

Figure 2: Graph depicting test set accuracies using Euclidean distance metric

We observe that as the value of k increases, the percentage accuracies keep increasing. Here are the precise values I obtained:

$$k = 1 : 75.925\% \tag{73}$$
$$k = 2 : 78.56\% \tag{74}$$
$$k = 3 : 80.81\% \tag{75}$$
$$k = 5 : 82.29\% \tag{76}$$
$$k = 10 : 83.205\% \tag{77}$$

This can be attributed to the fact that increasing k helps in preventing over fitting. As the value of k increases, the areas assigned to each class become smoother, preventing the decision boundary from tailoring itself too much to training data. This improves the capability of the algorithm to generalise and hence the performance improves as k increases.

Part2

The validation technique used was held out validation. I took the training data from the train.dat file. Then I randomly shuffled the entire data and picked the first 40,000 training points for my Xtr and Ytr. The rest 20,000 data points acted as the cross validation points. The prediction was then done for the following values of k: 1,5,8,12,15,19,20.

On measuring the accuracies for each k, I realised that k=12 gave me the highest accuracy percentage.So ideally k=12 is the best value I obtain.But I soon realised that running LMNN for k=12 was not feasible. So, instead I observed that uptill k=5, the surge in accuracy was very high, after which, even though the percentage accuracies did improve, the increase was very little. So, I set k=5 as my choice of k for LMNN training.

Part3

Using the above set value of k and choosing only 10,000 out of the 60,000 training points to train my LMNN model, I built the model and saved in the model.npy file. The decision to take 1/6 of the data was propelled by the fact that running LMNN for the entire dataset was not feasible on my machine.

The test accuracy obtained for the model thus learnt was 82.78 %