Introduction to ML (CS771), Autumn 2020
Indian Institute of Technology Kanpur
Homework Assignment Number 2

Student Name: Rishabh Kumar Chaudhary
Roll Number: 180609
Date: November 27, 2020

QUESTION 1

We are given the loss function

$$\mathcal{L}(\mathbf{w}) = -\sum_{n=1}^{N} \left(y_n \mathbf{w}^T \mathbf{x}_n - log(1 + exp(\mathbf{w}^T \mathbf{x}_n))\right) \tag{1}$$

Taking the first and second derivative, we can easily see that:

$$\nabla \mathcal{L}(\mathbf{w}) = -\sum_{n=1}^{N} \left(y_n \mathbf{x}_n - \frac{exp(\mathbf{w}^T \mathbf{x}_n)}{1 + exp(\mathbf{w}^T \mathbf{x}_n)} \mathbf{x}_n\right) \tag{2}$$

$$\nabla^2 \mathcal{L}(\mathbf{w}) = \sum_{n=1}^{N} \left(\frac{exp(\mathbf{w}^T \mathbf{x}_n)}{(1 + exp(\mathbf{w}^T \mathbf{x}_n))^2} \mathbf{x}_n \mathbf{x}_n^T\right) \tag{3}$$

We know that $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \left(\mathbf{H}^{(t)}\right)^{-1} \mathbf{g}^{(t)}$ is equivalent to

$$\mathbf{w}^{(t+1)} = \arg\min_{w} \left[\mathcal{L}(\mathbf{w}^{(t)}) + \nabla \mathcal{L}\left(\mathbf{w}^{(t)}\right)^T \left(\mathbf{w} - \mathbf{w}^{(t)}\right) + \frac{1}{2}\left(\mathbf{w} - \mathbf{w}^{(t)}\right)^T \nabla^2 \mathcal{L}\left(\mathbf{w}^{(t)}\right)(\mathbf{w} - \mathbf{w}^{(t)}))\right] \tag{4}$$

After substituting all the values, removing the terms which are not dependent on $\mathbf{w}$, we can reduce the above optimization problem to following form:

$$\arg\min_{w} \sum_{n=1}^{N} \frac{exp(\mathbf{w}^{(t)T} \mathbf{x}_n)}{(1 + exp(\mathbf{w}^{(t)T} \mathbf{x}_n))^2} \left(\mathbf{w}^{(t)T} \mathbf{x}_n + \frac{\left(1 + exp(\mathbf{w}^{(t)T} \mathbf{x}_n))\right)^2}{exp(\mathbf{w}^{(t)T} \mathbf{x}_n)} \times \left(y_n - \frac{exp(\mathbf{w}^{(t)T} \mathbf{x}_n)}{1 + exp(\mathbf{w}^{(t)T} \mathbf{x}_n)}\right) - \mathbf{w}^T \mathbf{x}_n\right)^2 \tag{5}$$

Therefore, we can say that:

$$\gamma_n = \frac{exp(\mathbf{w}^{(t)T} \mathbf{x}_n)}{(1 + exp(\mathbf{w}^{(t)T} \mathbf{x}_n))^2} \tag{6}$$

$$\hat{y}_n^{(t)} = \mathbf{w}^{(t)T} \mathbf{x}_n + \frac{\left(1 + exp(\mathbf{w}^{(t)T} \mathbf{x}_n))\right)^2}{exp(\mathbf{w}^{(t)T} \mathbf{x}_n)} \times \left(y_n - \frac{exp(\mathbf{w}^{(t)T} \mathbf{x}_n)}{1 + exp(\mathbf{w}^{(t)T} \mathbf{x}_n)}\right) \tag{7}$$

The value of γ_n makes sense here because it will give more importance to points which are closer to the decision boundary and less importance to points far away. Thus, giving a more robust decision boundary.

Introduction to ML (CS771), Autumn 2020
Indian Institute of Technology Kanpur
Homework Assignment Number 2

Student Name: Rishabh Kumar Chaudhary
Roll Number: 180609
Date: November 27, 2020

QUESTION 2

We will not store any weight vector $\mathbf{w}$ in this algorithm, instead of that, we will store a mistake counter vector $\boldsymbol{\alpha}$ of $n \times 1$ where n is the number of examples.

1. Initialize all the entries of vector $\boldsymbol{\alpha}$ to be 0.
2. For each training example $\mathbf{x}_j$, we will calculate expected output using

$$\hat{y} = sgn(\sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_j)) \tag{8}$$

   If $\hat{y} \neq y_j$ then it is a mistake.

3. If $\hat{y} \neq y_j$, then we know that a mistake has occured, so we increase α_j by 1.

Introduction to ML (CS771), Autumn 2020
Indian Institute of Technology Kanpur
Homework Assignment Number 2

Student Name: Rishabh Kumar Chaudhary
Roll Number: 180609
Date: November 27, 2020

QUESTION 3

We have to solve the following problem

$$\min_{\mathbf{w},v,\boldsymbol{\xi}} f(\mathbf{w},v,\boldsymbol{\xi}) = \frac{||\mathbf{w}||^2}{2} + \sum_{n=1}^{N} C_{y_n}\xi_n$$

$$\text{subject to } 1 - y_n(\mathbf{w}^T\mathbf{x}_n + b) - \xi_n \leq 0 \text{ and } -\xi_n \leq 0 \; n = 1 \ldots N$$

Forming the lagrangian equation gives us

$$\min_{\mathbf{w},v,\boldsymbol{\xi}} \max_{\alpha\geq 0,\beta\geq 0} L(\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta}) = \frac{||\mathbf{w}||^2}{2} + \sum_{n=1}^{N} C_{y_n}\xi_n + \sum_{n=1}^{N}\alpha_n(1 - y_n(\mathbf{w}^T\mathbf{x}_n + b) - \xi_n) - \sum_{n=1}^{N}\beta_n\xi_n$$

Taking derivative of L w.r.t. vw and setting to 0 gives

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \implies \mathbf{w} = \sum_{n=1}^{N}\alpha_n y_n \mathbf{x}_n$$

Taking derivative of L w.r.t. b and setting to 0 gives

$$\frac{\partial L}{\partial b} = 0 \implies \sum_{n=1}^{N}\alpha_n y_n = 0$$

Taking derivative of L w.r.t. ξ_n and setting to 0 gives

$$\frac{\partial L}{\partial \xi_n} = 0 \implies C_{y_n} - \alpha_n - \beta_n = 0$$

Since $\beta_n = C_{y_n} - \alpha_n$ and $\beta_n \geq 0$, we can say that $\alpha_n \leq C_{y_n}$. Substituting these in the Lagrangian gives us

$$\max_{\alpha_n \leq C_{y_n} \forall n=1\ldots N} L(\boldsymbol{\alpha}) = \sum_{n=1}^{N}\alpha_n - \frac{1}{2}\sum_{m,n=1}^{N}\alpha_n\alpha_m y_n y_m \mathbf{x}_m^T\mathbf{x}_n \text{ where } \sum_{n=1}^{N}\alpha_n y_n = 0$$

If we want more increase in loss when some positive value comes out to be negative, we can simple increase the value of C_{+1}. Like this, we can optimize the loss function when we want different loss for positive coming out negative than negative coming out positive and vice-versa.

Introduction to ML (CS771), Autumn 2020
Indian Institute of Technology Kanpur
Homework Assignment Number 2

Student Name: Rishabh Kumar Chaudhary
Roll Number: 180609
Date: November 27, 2020

QUESTION 4

We will store cluster assigned to every point. Mean of each cluster and the number of points in each cluster. Means are initialized randomly and each cluster has 0 points initially.

1. Randomly select a point x_n which has not been assigned any cluster yet.
2. Calculate distance of x_n from all the k means. Assign x_n to the cluster with nearest mean. Let that center be c and number of points in that cluster be v.
3. Increase v by 1.
4. Let the step size be $\eta = \frac{1}{v}$.
5. Calculate the new c using $c_{new} = (1 - \eta)c + \eta x_n$.

The above step size helps us in calculating a center which gives equal importance to all the points in the cluster. So, it is a good choice.

Student Name: Rishabh Kumar Chaudhary
Roll Number: 180609
Date: November 27, 2020

QUESTION 5

We know that in kernel k-means, we calculate the kernelized distance between two points using the following equation

$$||\phi(\mathbf{x}_n) - \phi(\boldsymbol{\mu}_k)||^2 = k(\mathbf{x}_n, \mathbf{x}_n) + k(\boldsymbol{\mu}_k, \boldsymbol{\mu}_k) - 2k(\mathbf{x}_n, \boldsymbol{\mu}_k) \tag{9}$$

The problem in the given situation is that we cannot store feature map representation of any point but we can use kernel function on two points to get a value in finite dimensions. Let the number of points in k^{th} cluster be C_k. For the means, we can say that:

$$\boldsymbol{\mu}_k = \phi^{-1}\left(\frac{1}{|C_k|}\sum_{n:z_n=k}\phi(\mathbf{x}_n)\right) \tag{10}$$

We cannot store these means, but we can easily use this representation to compute the distance. By inserting 10 in 9, we can say that:

$$D(\mathbf{x}_n, \boldsymbol{\mu}_k) = ||\phi(\mathbf{x}_n) - \phi(\boldsymbol{\mu}_k)||^2 = K(\mathbf{x}_n, \mathbf{x}_n) + \frac{\sum_{p,q:z_p=z_q=k} K(\mathbf{x}_p, \mathbf{x}_q)}{|C_k|^2} - \frac{2\sum_{p:z_p=k} K(\mathbf{x}_n, \mathbf{x}_p)}{|C_k|} \tag{11}$$

1. **Initialization**: We assign random K points K different clusters.
2. For each point x_n where $n = 1 \dots N$, we can assign cluster by:

$$z_n = \underset{k \in 1\dots K}{\arg\min}\left(K(\mathbf{x}_n, \mathbf{x}_n) + \frac{\sum_{p,q:z_p=z_q=k} K(\mathbf{x}_p, \mathbf{x}_q)}{|C_k|^2} - \frac{2\sum_{p:z_p=k} K(\mathbf{x}_n, \mathbf{x}_p)}{|C_k|}\right) \tag{12}$$

3. We cannot calculate the means implicitly in the given situation.

In normal k-means, cluster means are stored as D dimensional vectors whereas in this kernelized version, they are not explicitly stored, but only used in the kernel function and that too in terms of points in the cluster.

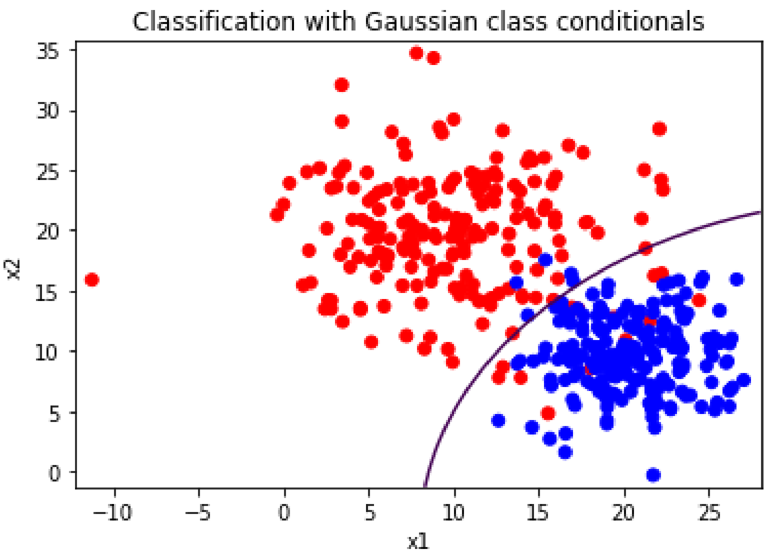
In the normal k-means assignment, mean calculation would take $\mathcal{O}(D)$ time and in the kernelized version, one point would take $\mathcal{O}(N^2)$ time if we have stored the kernel matrix. The kernelized version will be slow generally.

Introduction to ML (CS771), Autumn 2020
Indian Institute of Technology Kanpur
Homework Assignment Number 2

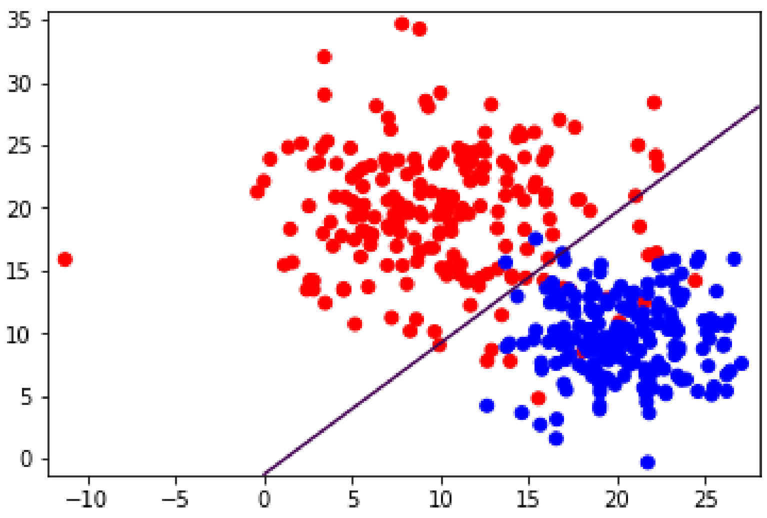
Student Name: Rishabh Kumar Chaudhary
Roll Number: 180609
Date: November 27, 2020

QUESTION 6

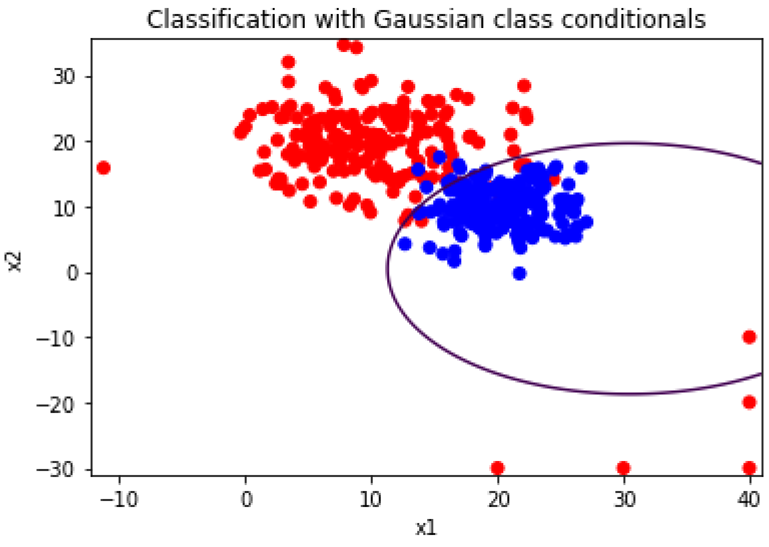
Following is the quadratic decision boundary with class conditionals $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_+, \sigma_+^2\mathbf{I}_2)$ and $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_-, \sigma_-^2\mathbf{I}_2)$ for the first dataset. The red dots are the positive points and blue are the negative ones.



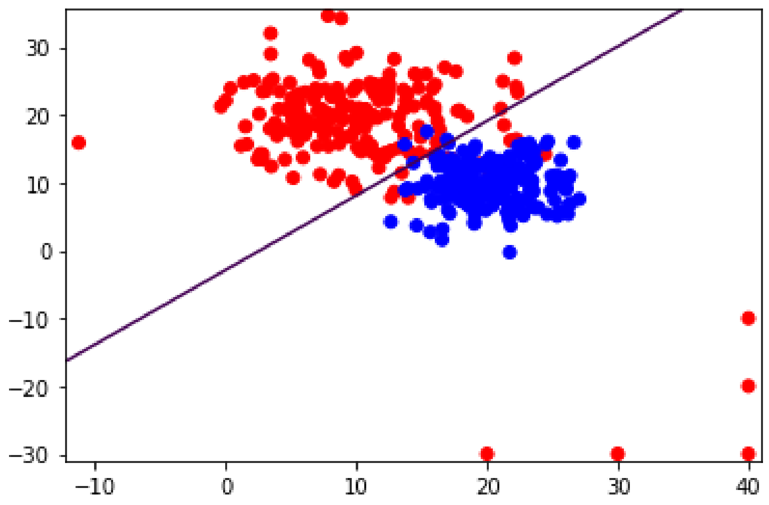
Following is the quadratic decision boundary with class conditionals $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_+, \sigma^2\mathbf{I}_2)$ and $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_-, \sigma^2\mathbf{I}_2)$ for the first dataset. The red dots are the positive points and blue are the negative ones.



Following is the quadratic decision boundary with class conditionals $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_+, \sigma_+^2\mathbf{I}_2)$ and $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_-, \sigma_-^2\mathbf{I}_2)$ for the second dataset. The red dots are the positive points and blue are the negative ones.



Following is the quadratic decision boundary with class conditionals $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_+, \sigma^2\mathbf{I}_2)$ and $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_-, \sigma^2\mathbf{I}_2)$ for the second dataset. The red dots are the positive points and blue are the negative ones.



Following are the plots and decision boundaries using SVM model. These models were generated using sci-kit learn and plotted using mlextend.

In general, generative model seems to learn better decision boundaries. If we see according to the dataset, we should use quadratic curve for the first dataset and SVM for the second dataset.

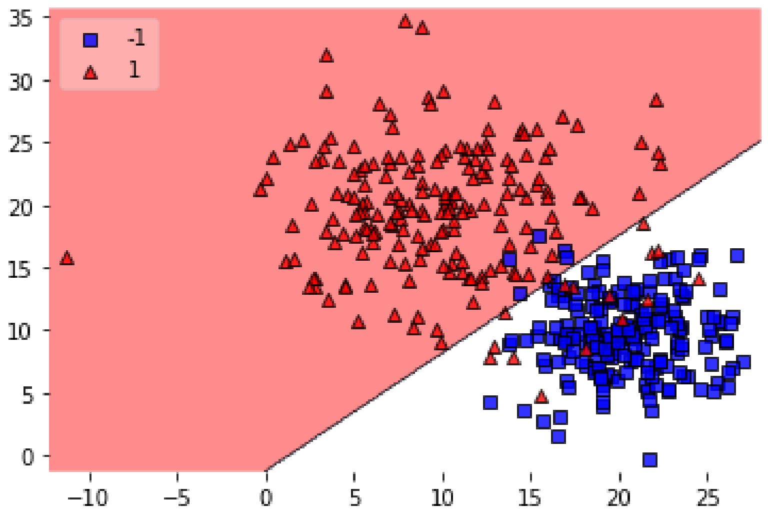


Figure 1: First dataset

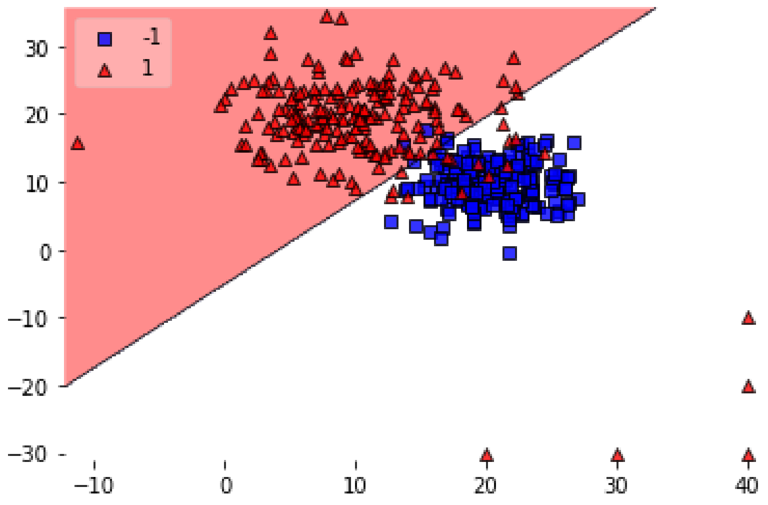


Figure 2: Second dataset