Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 3

QUESTION 1

Student Name: Aman Tiwari
Roll Number: 160094
Date: April 18, 2019

Solution 1:

We need to solve the following optimization problem $\arg\max_{0\leq\boldsymbol{\alpha}\leq C} f(\boldsymbol{\alpha})$ where f is given as,
$f(\boldsymbol{\alpha}) = \boldsymbol{\alpha}^T 1 - \frac{1}{2}\boldsymbol{\alpha}^T \mathbf{G}\boldsymbol{\alpha}$
In coordinate ascent we update a uniformly chosen entry α_n of vector $\boldsymbol{\alpha}$,in this case the function can be written as,
$f(\boldsymbol{\alpha}) = \alpha_n - \frac{1}{2}(G_{nn}{\alpha_n}^2 + 2\sum_{m=1,m\neq n}^{N} G_{mn}\alpha_m\alpha_n)$ + constant
Now $\alpha_n = \alpha_n + \delta_*$, where $\delta_* = \arg\max_\delta f(\boldsymbol{\alpha} + \delta e_n)$
So we can rewrite our objective function in the following manner,
$f(\boldsymbol{\alpha}+\delta e_n) = \delta - \frac{1}{2}(\delta^2 G_{nn} + 2G_{nn}\alpha_n\delta + 2\sum_{m=1,m\neq n}^{N} G_{mn}\alpha_m\delta) = \delta - \frac{1}{2}(\delta^2 G_{nn} + 2\sum_{m=1}^{N} G_{mn}\alpha_m\delta)$
This is a quadratic in δ. Taking the derivative and setting it to zero we get,
$\delta' = (1 - \sum_{m=1}^{N} G_{mn}\alpha_m)/G_{nn}$
Now we have a constraint on the value of δ_*,
$-\alpha_n \leq \delta_* \leq C - \alpha_n$, but the value of δ' computed above may not lie in this range. Since the objective function is quadratic with respect to δ therefore if δ' does not lie in the given range then the value of δ which maximizes the function will lie at the boundary points. Hence we have the following value of δ_*

$$\delta_* = \begin{cases} \delta' & -\alpha_n \leq \delta' \leq C - \alpha_n \\ C - \alpha_n & \delta' \geq C - \alpha_n \\ -\alpha_n & \delta' \leq -\alpha_n \end{cases}$$

The sketch of the coordinate ascent algorithm is as follows:

1. Initialize $\boldsymbol{\alpha}$ as $\boldsymbol{\alpha}^{(0)}$
2. Pick a random i ∈ {1,2,3,...,N} and update $\boldsymbol{\alpha}$ as follows:
   $\alpha_i = \alpha_i + \delta_*$ where δ_* is defined above.
3. Repeat until convergence.

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 3

Student Name: Aman Tiwari
Roll Number: 160094
Date: April 18, 2019

QUESTION 2

Solution 2:

Let P_W represent the sum of squared distances between all pairs of points that are in *different* clusters. Consider the following sum for the given data points,
$S = \sum_{n,m} ||\mathbf{x}_n - \mathbf{x}_m||^2$
Clearly this sum constant for a given set of points . This sum can be written as:
$S = \sum_{n,m} \mathbb{1}[f_n = f_m]||\mathbf{x}_n - \mathbf{x}_m||^2 + \mathbb{1}[f_n \neq f_m]||\mathbf{x}_n - \mathbf{x}_m||^2$
$S = L_W + P_W$
$P_W = S - L_W$
Therefore by minimizing L_W we also maximize P_W.

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 3

QUESTION 3

Student Name: Aman Tiwari
Roll Number: 160094
Date: April 18, 2019

Solution 3:

Let us denote $\mathbf{x}_n^{miss}$ by $\mathbf{x}_n^m$ and $\mathbf{x}_n^{obs}$ by $\mathbf{x}_n^o$.Also , without loss of generality,let us assume that $\mathbf{x}_n = [\mathbf{x}_n^m, \mathbf{x}_n^o]$ Then the expression for $p(\mathbf{x}_n^m|\mathbf{x}_n^o, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is given as : (using the results from Section 4.3.1 of MLAPP)

$p(\mathbf{x}_n^m|\mathbf{x}_n^o, \mu, \Sigma) = \mathcal{N}(\mathbf{x}_n^m|\boldsymbol{\mu}_{m|o}, \boldsymbol{\Sigma}_{m|o})$

$\boldsymbol{\mu}_{m|o} = \boldsymbol{\mu}_m + \boldsymbol{\Sigma}_{mo}\boldsymbol{\Sigma}_{oo}^{-1}(\mathbf{x}_n^o - \boldsymbol{\mu}_o)$

$\boldsymbol{\Sigma}_{m|o} = \boldsymbol{\Sigma}_{mm} - \boldsymbol{\Sigma}_{mo}\boldsymbol{\Sigma}_{oo}^{-1}\boldsymbol{\Sigma}_{om}$

$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_m \\ \boldsymbol{\mu}_o \end{bmatrix}$

$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{mm} & \boldsymbol{\Sigma}_{mo} \\ \boldsymbol{\Sigma}_{om} & \boldsymbol{\Sigma}_{oo} \end{bmatrix}$

The CLL in this case can be written as :

$\mathcal{Q}(\Theta, \Theta^{old}) = \mathbb{E}[\sum_{n=1}^N \log(\mathbf{x}_n^o, \mathbf{x}_n^m|\Theta)] = \mathbb{E}[\sum_{n=1}^N \log(\mathbf{x}_n|\Theta)] = \sum_{n=1}^N \mathbb{E}[\log(\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}, \boldsymbol{\Sigma}))]$

$\mathcal{Q}(\Theta, \Theta^{old}) = \frac{-N}{2}\log|2\pi\boldsymbol{\Sigma}| - \frac{1}{2}\sum_{n=1}^N \mathbb{E}[(\mathbf{x}_n - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}_n - \boldsymbol{\mu})]$

$\mathcal{Q}(\Theta, \Theta^{old}) = \frac{-N}{2}\log|2\pi\boldsymbol{\Sigma}| - \frac{1}{2}trace(\boldsymbol{\Sigma}^{-1}\sum_{n=1}^N \mathbb{E}[(\mathbf{x}_n - \boldsymbol{\mu})(\mathbf{x}_n - \boldsymbol{\mu})^T])$

$\mathcal{Q}(\Theta, \Theta^{old}) = \frac{-N}{2}\log|2\pi\boldsymbol{\Sigma}| - \frac{1}{2}trace(\boldsymbol{\Sigma}^{-1}\sum_{n=1}^N \mathbb{E}[S(\boldsymbol{\mu})])$

$\mathbb{E}[S(\boldsymbol{\mu})] = \mathbb{E}[\mathbf{x}_n\mathbf{x}_n^T] - \mathbb{E}[\mathbf{x}_n]\boldsymbol{\mu} - \boldsymbol{\mu}\mathbb{E}[\mathbf{x}_n]^T + \boldsymbol{\mu}\boldsymbol{\mu}^T$

So we need to compute the expectations $\mathbb{E}[\mathbf{x}_n]$ and $\mathbb{E}[\mathbf{x}_n\mathbf{x}_n^T]$ to get the expression for CLL.

We can use the mean of the above posterior distribution to get the value of $\mathbb{E}[\mathbf{x}_n]$.

$\mathbb{E}[\mathbf{x}_n] = (\mathbb{E}[\mathbf{x}_n^m, \mathbf{x}_n^o]) = (\boldsymbol{\mu}_{m|o}, \mathbf{x}_n^o)$

To calculate $\mathbb{E}[\mathbf{x}_n\mathbf{x}_n^T]$ we can use the result that $Cov[\mathbf{x}\mathbf{x}^T] = \mathbb{E}[\mathbf{x}\mathbf{x}^T] - \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{x}^T]$.

$\mathbb{E}[\mathbf{x}_n\mathbf{x}_n^T] = \mathbb{E}[\begin{bmatrix} \mathbf{x}_n^m \\ \mathbf{x}_n^o \end{bmatrix}[(\mathbf{x}_n^m)^T, (\mathbf{x}_n^o)^T]] = \begin{bmatrix} \mathbb{E}[\mathbf{x}_n^m(\mathbf{x}_n^m)^T] & \mathbb{E}[\mathbf{x}_n^m](\mathbf{x}_n^o)^T \\ \mathbf{x}_n^o\mathbb{E}[\mathbf{x}_n^m]^T & \mathbf{x}_n^o(\mathbf{x}_n^o)^T \end{bmatrix}$ where

$\mathbb{E}[\mathbf{x}_n^m(\mathbf{x}_n^m)^T] = \mathbb{E}[\mathbf{x}_n^m]\mathbb{E}[\mathbf{x}_n^m]^T + \boldsymbol{\Sigma}_{m|o}$ and $\mathbb{E}[\mathbf{x}_n^m] = \boldsymbol{\mu}_{m|o}$

To get the update equations for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ we can use the results from MLE of multivariate gaussian are replace $\mathbf{x}_n$ by $\mathbb{E}[\mathbf{x}_n]$ and $\mathbf{x}_n\mathbf{x}_n^T$ by $\mathbb{E}[\mathbf{x}_n\mathbf{x}_n^T]$. Hence we get the following results:

$\boldsymbol{\mu} = \frac{1}{N}\sum_{n=1}^N \mathbb{E}[\mathbf{x}_n]$

$\boldsymbol{\Sigma} = \frac{1}{N}\sum_{n=1}^N \mathbb{E}[\mathbf{x}_n\mathbf{x}_n^T] - \boldsymbol{\mu}\boldsymbol{\mu}^T$

The EM algorithm can then be written as:

1. Initialize $\Theta = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ as $\Theta^{(0)}$, set t = 1
2. E Step: Compute $\mathbb{E}[\mathbf{x}_n]$ and $\mathbb{E}[\mathbf{x}_n\mathbf{x}_n^T]$ using the equations given above.
3. M step: Update $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ using the equations given above.
4. Set t = t+1 and go to step 2 if not converged.

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 3

Student Name: Aman Tiwari
Roll Number: 160094
Date: April 18, 2019

QUESTION

4

Solution 4:

We can use $\mathbf{z}_n$ to represent the labels of the unknown examples where $z_{nk} = 1$ if n^{th} example belongs to class k and $z_{nk} = 0$ otherwise. Here $n \in \{N+1, N+2, N+3, ...N+M\}$. In this case the CLL expression can be written as:

$\log p(\mathbf{X}, \mathbf{Z}|\Theta) = \sum_{n=1}^{N} \sum_{k=1}^{K} y_{nk}[\log \pi_k + \log \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)] + \sum_{n=N+1}^{N+M} \sum_{k=1}^{K} z_{nk}[\log \pi_k + \log \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]$

The expected CLL $\mathcal{Q}(\Theta, \Theta^{old})$ is given as:

$\mathcal{Q}(\Theta, \Theta^{old}) = \sum_{n=1}^{N} \sum_{k=1}^{K} y_{nk}[\log \pi_k + \log \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)] + \sum_{n=N+1}^{N+M} \sum_{k=1}^{K} \mathbb{E}[z_{nk}][\log \pi_k + \log \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]$

The expression of $\mathbb{E}[z_{nk}]$ will be same as in the case of Gaussian Mixture Model and is given as:

$\mathbb{E}[z_{nk}] = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{l=1}^{K} \pi_l \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}$

We define γ_{nk} as follows:

$$\gamma_{nk} = \begin{cases} y_{nk} & n \leq N \\ \mathbb{E}[z_{nk}] & N+1 \leq n \leq N+M \end{cases}$$

Now the expression for expected CLL can be written as follows:

$\mathcal{Q}(\Theta, \Theta^{old}) = \sum_{n=1}^{N+M} \sum_{k=1}^{K} \gamma_{nk}[\log \pi_k + \log \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]$

This expression is identical to the CLL expression for the case of Gaussian Mixture Model and hence we can use those results to write the updates for Θ.

The update equations are given as:

$\boldsymbol{\mu}_k^{(t)} = \frac{1}{N_k} \sum_{n=1}^{N+M} \gamma_{nk}^{(t)} \mathbf{x}_n$

$\boldsymbol{\Sigma}_k^{(t)} = \frac{1}{N_k} \sum_{n=1}^{N+M} \gamma_{nk}^{t} (\mathbf{x}_n - \boldsymbol{\mu}_k^{(t)})(\mathbf{x}_n - \boldsymbol{\mu}_k^{(t)})^T$

$\pi_k^{(t)} = \frac{N_k}{N+M}$

Here $N_k = \sum_{n=1}^{N+M} \gamma_{nk}$

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 3

Student Name: Aman Tiwari
Roll Number: 160094
Date: April 18, 2019

QUESTION

5

Solution 5:
This model is trying to use a mixture of weight vectors to model the given data. In general it may not be possible for a single weight vector model the data effectively. This method in a way divides the data into different classes and models the data of each class using a different weight vector. Hence it is expected to work better than a standard probabilistic model.

The ALT-OPT algorithm in this case is given as follows:

1. Initialize Θ as $\hat{\Theta}$.
2. For $n = 1, ...N$ find the best z_n: $\hat{z_n} = \arg\max_k p(\mathbf{y}_n, z_n = k|\hat{\Theta})$
   $\hat{z_n} = \arg\max_k p(z_n = k|\hat{\Theta})p(\mathbf{y}_n|z_n = k, \hat{\Theta})$
   $\hat{z_n} = \arg\max_k \pi_k \mathcal{N}(\mathbf{w}_k^T\mathbf{x}_n, \beta^{-1})$
3. Given $\mathbf{Z} = \{\hat{z_1}, ...\hat{z_n}\}$, re-estimate Θ using MLE.
   $\hat{\Theta} = \arg\max_\Theta \log p(\mathbf{Y}, \hat{\mathbf{Z}}|\Theta)$
   After simplification we get the following expression:
   $\hat{\Theta} = \arg\max_\Theta \sum_{n=1}^N \sum_{k=1}^K \hat{z}_{nk}[\log(\pi_k) + \log \mathcal{N}(\mathbf{w}_k^T\mathbf{x}_n, \beta^{-1})]$
   The update for π_k will be same as in the case of generative classification model. To find the update for $\mathbf{w}_k$ we can write the expression in the following manner:
   $\hat{\mathbf{w}_k} = \arg\max_{\mathbf{w}_k} \sum_{n:\hat{z}_{nk}=1} \log \mathcal{N}(\mathbf{w}_k^T\mathbf{x}_n, \beta^{-1})$
   This is same as the objective function for linear regression problem , the difference is that only a subset of the examples contribute. Hence the update equations are given as:
   $\hat{\pi_k} = \frac{N_k}{N}$
   $\hat{\mathbf{w}_k} = (\sum_{n:\hat{z}_{nk}=1} \mathbf{x}_n\mathbf{x}_n^T)^{-1} \sum_{n:\hat{z}_{nk}=1} y_n\mathbf{x}_n$
4. Go to step 2 if not yet converged

If $\pi_k = \frac{1}{K}$ then the update for z_n is given as:
$\hat{z_n} = \arg\max_k \mathcal{N}(\mathbf{w}_k^T\mathbf{x}_n, \beta^{-1})$
This update chooses the $\mathbf{w}_k$ which results in highest probability or least square error for the data point $(\mathbf{x}_n, y_n)$.

The EM algorithm is given as follows:

1. Initialize Θ as $\Theta^{(0)}$
2. $\mathbb{E}[z_{nk}^{(t)}] = \frac{\pi_k^{(t-1)}\mathcal{N}(\mathbf{x}_n^T\mathbf{w}_k^{(t-1)}, \beta^{-1})}{\sum_{l=1}^K \pi_l^{(t-1)}\mathcal{N}(\mathbf{x}_n^T\mathbf{w}_l^{(t-1)}, \beta^{-1})}$
3. $\gamma_{nk} = \mathbb{E}[z_{nk}]$ and $N_k = \sum_{n=1}^N \gamma_{nk}$, restimate Θ via MLE
   $\pi_k^{(t)} = \frac{N_k}{N}$
   $\mathbf{w}_k^{(t)} = (\sum_{n=1}^N \mathbf{x}_n\mathbf{x}_n^T\gamma_{nk})^{-1} \sum_{n=1}^N y_n\mathbf{x}_n\gamma_{nk}$
4. Set t = t + 1 and go to step 2 if not yet converged

Let us consider the expression for $\mathbb{E}[z_{nk}] = \frac{\pi_k \mathcal{N}(\mathbf{x}_n^T \mathbf{w}_k, \beta^{-1})}{\sum_{l=1}^{K} \pi_l \mathcal{N}(\mathbf{x}_n^T \mathbf{w}_l, \beta^{-1})}$. Divide the numerator and denominator by $\pi_k \mathcal{N}(\mathbf{x}_n^T \mathbf{w}_k, \beta^{-1})$ to get:

$$\mathbb{E}[z_{nk}] = \frac{1}{\sum_{l=1}^{K} (\pi_l/\pi_k) \exp(((y_n - \mathbf{w}_k^T \mathbf{x}_n)^2 - (y_n - \mathbf{w}_l^T \mathbf{x}_n)^2))\beta)}$$

The solution that we get from ALT-OPT for $\hat{z_n}$ will be a one hot vector where $z_{nk} = 1$ if $\pi_k \mathcal{N}(\mathbf{w}_k^T \mathbf{x}_n, \beta^{-1})$ is maximum among all values of k. Now let us consider the EM case when $\beta \to \infty$.
The value of k for which $\mathcal{N}(\mathbf{w}_k^T \mathbf{x}_n, \beta^{-1})$ is maximum or $(y_n - \mathbf{w}_k^T \mathbf{x}_n)^2$ is minimum will cause all the terms in the denominator to go to zero expect one term which will be 1. Hence the expression will be 1 for this value of k. For any other value of k at least one value in the denominator blows to ∞ hence the expression goes to zero. So we see that EM chooses the value of k which maximizes the probability $\mathcal{N}(\mathbf{w}_k^T \mathbf{x}_n, \beta^{-1})$.Also as $\beta \to \infty$ this is same as choosing the value of k which maximizes $\pi_k \mathcal{N}(\mathbf{w}_k^T \mathbf{x}_n, \beta^{-1})$. This is same as what ALT-OPT does. Hence EM converges to ALT-OPT when $\beta \to \infty$.

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 3

Student Name: Aman Tiwari
Roll Number: 160094
Date: April 18, 2019

QUESTION
6

Solution 6:
Problem 1:

1. As the value of lambda increases the RMSE also increases.The RMSE values for different values of lambda are as follows:
   $\lambda = 0.1$ RMSE = 0.2653
   $\lambda = 1$ RMSE = 7.250
   $\lambda = 10$ RMSE = 92.801
   $\lambda = 100$ RMSE = 207.579

2. As the number of landmark points increases the RMSE error decreases. The values of $L \geq 50$ seem good enough for the data. The RMSE values for different values of L are as follows:
   $L = 2$ RMSE = 238.00
   $L = 5$ RMSE = 207.93
   $L = 20$ RMSE = 42.01
   $L = 50$ RMSE = 2.136
   $L = 100$ RMSE = 0.6735

Problem 2:

1. In this case since the distance of points belonging to the two clusters are different we can simply use the distance of the points form origin as a feature to cluster the points.

2. The data points for the second cluster are more spread out as compared to the data points first cluster . So a single landmark point in the outer cluster will not work well since it is not a good representation of the points of outer cluster. On the other hand since data points of the inner cluster are not much spread out hence even a single point from this cluster can act as a good representation of the cluster. So when the landmark is from the inner cluster the algorithm works well but if the landmark is from outer cluster it does not work well.