Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 4

QUESTION 1

Student Name: Subham Kumar
Roll Number: 160707
Date: February 23, 2019

Given eigenvector $\mathbf{v} \epsilon R^n$ of the matrix $\frac{1}{N}\mathbf{X}\mathbf{X}^T$ we can use it to the eigenvector $\mathbf{u} \epsilon R^d$ of $\mathbf{S} = \frac{1}{N}\mathbf{X}^T\mathbf{X}$:

$$\frac{1}{N}\mathbf{X}\mathbf{X}^T\mathbf{v} = \lambda\mathbf{v}$$

Multiplying both side by $\mathbf{X}^T$,we get

$$\frac{1}{N}\mathbf{X}^T\mathbf{X}\left(\mathbf{X}^T\mathbf{v}\right) = \lambda\left(\mathbf{X}^T\mathbf{v}\right)$$

This $\mathbf{X}^T\mathbf{v}$ is simply the eigenvector $\mathbf{u}$ of $\mathbf{S}$. The time complexity of the traditional PCA is $\mathcal{O}(D^3)$ whereas this of calculating eigenvectors will have time complexity of $\mathcal{O}(N^3)+\mathcal{O}(DN^2) = \mathcal{O}(DN^2)$.Clearly the later one is better when $D > N$.

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 4

QUESTION 2

Student Name: Subham Kumar
Roll Number: 160707
Date: February 23, 2019

Given the activation function $h(x) = x\sigma(\beta x)$,we can choose suitable values of β to approximate it to get the mentioned activation functions.Here σ denotes the sigmoid function $\sigma(z) = \frac{1}{1+\exp{(-z)}}$.
To get linear activation function set β to be zero.Then $h(x) = \frac{x}{2}$ which is linear.
To get Relu activation function set β to be some very large number.In this case

$$h(x) = \begin{cases} 0 & \forall x < 0 \\ x & otherwise \end{cases}$$

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 4

QUESTION 3

Student Name: Subham Kumar
Roll Number: 160707
Date: February 23, 2019

$$p(y_n|z_n, \mathbf{x}_n) = Bernoulli\left[\sigma\left(\mathbf{w}_{z_n}^T \mathbf{x}_n\right)\right]$$

So,the marginal distribution $p(y_n = 1|\mathbf{x}_n)$ can be written as:

$$p(y_n = 1|\mathbf{x}_n) = \sum_{k=1}^{K} p(z_n = k)p(y_n = 1|z_n = k, \mathbf{x}_n)$$

$$= \sum_{k=1}^{K} \pi_k \sigma\left(\mathbf{w}_k^T \mathbf{x}_n\right)$$

Now consider an equivalent neural network which has an input layer,a hidden layer and an output layer with the following specifications:
1.The input layer will have all the features of an input(say $\mathbf{x}_n$) going into it i.e. it will have D nodes in input layer.
2.The hidden layer will have K nodes.Each node(say k_{th} node) will have pre-activation as $\sum_{d=1}^{D} w_{kd}x_{nd} = \mathbf{w}_k^T \mathbf{x}_n$ i.e. the edge from the input node x_{nd} to the k_{th} node of hidden layer will have weight w_{kd}.Then the non-linear activation used will be sigmoid.Hence the output of this node will $\sigma\left(\mathbf{w}_k^T \mathbf{x}_n\right)$
3.Now for the output layer there will be a single node,where the pre-activation will be $\sum_{k=1}^{K} \pi_k \sigma\left(\mathbf{w}_k^T \mathbf{x}_n\right)$ i.e each node from the hidden layer(say k_{th} node) will have an edge to the output-layer node with weight π_k.The activation used in the output layer will be simply identity activation function.
So the final output of this neural network will be $\sum_{k=1}^{K} \pi_k \sigma\left(\mathbf{w}_k^T \mathbf{x}_n\right)$ which is same as $p(y_n = 1|\mathbf{x}_n)$

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 4

Student Name: Subham Kumar
Roll Number: 160707
Date: February 23, 2019

QUESTION 4

$$\text{Loss Function=Negative of MAP Objective: } = -\log p(\Theta|X)$$

Here $\Theta = \left\{ \{\mathbf{u}_n, \theta_n\}_{n=1}^{N}, \{\mathbf{v}_m, \phi_m\}_{m=1}^{M}, \mathbf{W}_u, \mathbf{W}_v \right\}$

$$-\log p(\Theta|X) = -\log p(X|\Theta) - \log p(\Theta)$$

$$-\log p(\Theta|X) = -\log p(X|\Theta) - \log p(\mathbf{u}) - \log p(\mathbf{v})$$

$$= \sum_{(n,m)\epsilon\Omega} \lambda_x (X_{nm} - (\theta_n + \phi_m + \mathbf{u}_n^T \mathbf{v}_m))^2 + \sum_{n=1}^{N} \lambda_u (\mathbf{u}_n - \mathbf{W}_u \mathbf{a}_n)^T (\mathbf{u}_n - \mathbf{W}_u \mathbf{a}_n) + \sum_{m=1}^{M} \lambda_v (\mathbf{v}_m - \mathbf{W}_v \mathbf{b}_m)^T (\mathbf{v}_m - \mathbf{W}_v \mathbf{b}_m)$$

Taking the partial derivative of $-\log p(\Theta|X)$ w.r.t. $\theta_n, \mathbf{u}_n, \mathbf{v}_m, \phi_m$ we get the following expressions for updates:

$$\mathbf{W}_u = \sum_{n=1}^{N} (\mathbf{u}_n \mathbf{a}_n^T) \left(\sum_{n=1}^{N} \mathbf{a}_n \mathbf{a}_n^T \right)^{-1}$$

$$\mathbf{W}_v = \sum_{m=1}^{M} (\mathbf{v}_m \mathbf{b}_m^T) \left(\sum_{m=1}^{M} \mathbf{b}_m \mathbf{b}_m^T \right)^{-1}$$

$$\theta_n = \frac{\sum\limits_{m\epsilon\Omega_{\mathrm{r_n}}} (X_{nm} - (\phi_m + \mathbf{u}_n^T \mathbf{v}_m))}{\Omega_{\mathrm{r_n}}}$$

$$\phi_m = \frac{\sum\limits_{n\epsilon\Omega_{\mathrm{c_m}}} (X_{nm} - (\theta_n + \mathbf{v}_m^T \mathbf{u}_n))}{\Omega_{\mathrm{c_m}}}$$

$$\mathbf{u}_n = \left(\lambda_u \mathcal{I}_k + \lambda_x \sum_{m\epsilon\Omega_{\mathrm{r_n}}} \mathbf{v}_m \mathbf{v}_m^T \right)^{-1} \left(\lambda_u \mathbf{W}_u \mathbf{a}_n + \lambda_x \sum_{m\epsilon\Omega_{\mathrm{r_n}}} (X_{nm} - \theta_n - \phi_m) \mathbf{v}_m \right)$$

$$\mathbf{v}_m = \left(\lambda_v \mathcal{I}_k + \lambda_x \sum_{n\epsilon\Omega_{\mathrm{c_m}}} \mathbf{u}_n \mathbf{u}_n^T \right)^{-1} \left(\lambda_v \mathbf{W}_v \mathbf{b}_m + \lambda_x \sum_{n\epsilon\Omega_{\mathrm{c_m}}} (X_{nm} - \theta_n - \phi_m) \mathbf{u}_n \right)$$

ALT-OPT Algorithm:
1.Initialize all the parameters belonging to $\Theta^{(0)}$.Set t=1.
2.
For all $n\epsilon\{1, 2, ..., N\}$ update $\mathbf{u}_n$ as:

$$\mathbf{u}_n^{(t)} = \left(\lambda_u \mathcal{I}_k + \lambda_x \sum_{m\epsilon\Omega_{\mathrm{r_n}}} \mathbf{v}_m^{(t-1)} \mathbf{v}_m^{(t-1)^T} \right)^{-1} \left(\lambda_u \mathbf{W}_u^{(t-1)} \mathbf{a}_n + \lambda_x \sum_{m\epsilon\Omega_{\mathrm{r_n}}} (X_{nm} - \theta_n^{(t-1)} - \phi_m^{(t-1)}) \mathbf{v}_m^{(t-1)} \right)$$

$$\mathbf{W}_u^{(t)} = \sum_{n=1}^{N} \left(\mathbf{u}_n^{(t)} \mathbf{a}_n^T\right) \left(\sum_{n=1}^{N} \mathbf{a}_n \mathbf{a}_n^T\right)^{-1}$$

3.
For all $m \epsilon \{1, 2, ..., M\}$ update $\mathbf{v}_m$ as:

$$\mathbf{v}_m^{(t)} = \left(\lambda_v \mathcal{I}_k + \lambda_x \sum_{n \epsilon \Omega_{\mathrm{c_m}}} \mathbf{u}_n^{(t)} \mathbf{u}_n^{(t)^T}\right)^{-1} \left(\lambda_v \mathbf{W}_v^{(t-1)} \mathbf{b}_m + \lambda_x \sum_{n \epsilon \Omega_{\mathrm{c_m}}} (X_{nm} - \theta_n^{(t-1)} - \phi_m^{(t-1)}) \mathbf{u}_n^{(t)}\right)$$

$$\mathbf{W}_v^{(t)} = \sum_{m=1}^{M} \left(\mathbf{v}_m^{(t)} \mathbf{b}_m^T\right) \left(\sum_{m=1}^{M} \mathbf{b}_m \mathbf{b}_m^T\right)^{-1}$$

4.

$$\theta_n^{(t)} = \frac{\sum_{m \epsilon \Omega_{\mathrm{r_n}}} (X_{nm} - (\phi_m^{(t-1)} + \mathbf{u}_n^{(t)^T} \mathbf{v}_m^{(t)}))}{\Omega_{\mathrm{r_n}}}$$

$$\phi_m^{(t)} = \frac{\sum_{n \epsilon \Omega_{\mathrm{c_m}}} (X_{nm} - (\theta_n^{(t)} + \mathbf{u}_n^{(t)^T} \mathbf{v}_m^{(t)}))}{\Omega_{\mathrm{c_m}}}$$

5.Go to step 2 if not converged yet,set t=t+1.