# HW4 Handwritten Assignment

Lecturor: Pei-Yuan Wu
TAs: Chun-Lin Huang(Problem 1, 2), Yuan-Chia Chang(Problem 3, 4, 5)

November 2023, Third Edition

## Problem 1 (EM algorithm for mixture of Bernoulli model)(1.5%)

Consider the generative model parameterized by $\theta = (\pi_k, \boldsymbol{\mu}_k)_{k=1}^K$, where $\pi_1, ..., \pi_K \in [0,1]$ satisfies $\sum_{k=1}^K \pi_k = 1$, and that $\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K \in [0,1]^D$, so that the probability of generating a $D$-dimensional binary vector $\mathbf{x} = (x^{(1)}, ..., x^{(D)}) \in \{0,1\}^D$ is

$$p(\mathbf{x}; \theta) = \sum_{k=1}^K \pi_k \prod_{j=1}^D \mu_{kj}^{x^{(j)}} (1 - \mu_{kj})^{1-x^{(j)}}$$

In other words, with given $\boldsymbol{\mu}_k$, the elements $x^{(1)}, ..., x^{(D)}$ are independent, where $x^{(j)}$ follows Bernoulli distribution of mean $\mu_{kj}$. Suppose we observe training data of $N$ binary vectors $\mathbf{x}_1, ..., \mathbf{x}_N \in \{0,1\}^D$, derive the E-step and M-step equations of the EM algorithm for optimizing the mixing coefficients $\pi_k$ and the Bernoulli means $\mu_{kj}$ by maximum likelihood.

## Problem 2 (EM algorithm for mixture of exponential model)(1.5%)

Given N samples $x_1, \ldots, x_N \in [0, \infty)$, we would like to cluster them into $K$ clusters. Assume the samples are generated according to Exponential mixture models

$$X \sim \sum_{j=1}^K \pi_j Exp(\tau_j)$$

where $\pi_1 + \ldots + \pi_K = 1$, and $Exp(\tau)$ denotes the exponential distribution with probability density function

$$f_\tau(x) = \begin{cases} (1/\tau)e^{-x/\tau} & , \ x \geq 0 \\ 0 & , x < 0 \end{cases}$$

We would like to apply Expectation Maximization algorithm to find the maximum likelihood estimation of parameters $\theta = \{(\pi_k, \tau_k)\}_{k=1}^K$.

(a) Please write down the E-step and M-step and show that the parameters are updated from $\theta^{(t)} = \left\{(\pi_k^{(t)}, \tau_k^{(t)})\right\}_{k=1}^K$ to $\theta^{(t+1)} = \left\{(\pi_k^{(t+1)}, \tau_k^{(t+1)})\right\}_{k=1}^K$ in the following form:

$$\tau_k^{(t+1)} = \frac{\sum_{i=1}^N \delta_{ik}^{(t)} x_i}{\sum_{i=1}^N \delta_{ik}^{(t)}}, \quad \pi_k^{(t+1)} = \frac{1}{N} \sum_{i=1}^N \delta_{ik}^{(t)}$$

(b) What is the closed form expression of $\delta_{ik}^{(t)}$?

## Problem 3 (Boosting)(0.5%)

1. Consider training a boosting classifier using decision stumps on the data set illustrated in Figure 1:

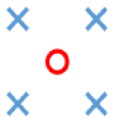

Figure 1: AdaBoost Data set

(a) Which examples will have their weights increased at the end of the first iteration? Circle them.

(b) How many iterations will it take to achieve zero training error? Justify your answers.

2. Suppose AdaBoost is run on $N$ training examples, and suppose on each round that the weighted training error $\epsilon_t$ of the $t$'th weak hypothesis is at most $1/2 - \gamma$, for some number $0 < \gamma < 1/2$. After how many iterations, $T$, will the combined hypothesis be consistent with the $N$ training examples, i.e., achieves zero training error? Your answer should only be expressed in terms of $N$ and $\gamma$. (Hint: Recall that exponential loss is an upper bound for 0-1 loss. What is the training error when 1 example is misclassified?)

## Problem 4 (Expectation Maximization Interpretation behind Semi-Supervised Learning)(1%)

Given N samples $\mathbf{x}_1, \ldots, \mathbf{x}_N \in \mathbb{R}^m$ as well as their labels $y_1, \ldots, y_N \in \{0, 1, \ldots, K\}$. Consider the generative model where each sample $\mathbf{x}_i$ is generated independently according to Gaussian mixture model $\underline{\underline{\text{that depends on the label } y_i}}$, as represented by random variable

$$X_i \sim \begin{cases} \sum_{j=1}^{K} \pi_j \mathcal{N}\left(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right) & , \; if \;\; y_i = 0 \\ \mathcal{N}\left(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) & , \; if \;\; y_i = k \neq 0 \end{cases}$$

where $\pi_1 + \ldots + \pi_K = 1$, and $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, with probability density function

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^m |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right)$$

We would like to apply Expectation Maximization algorithm to find the maximum likelihood estimation of parameters $\theta = \{(\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\}_{k=1}^{K}$.

1. Please write down the E-step and M-step and show that the parameters are updated from $\theta^{(t)} = \left\{\left(\pi_k^{(t)}, \boldsymbol{\mu}_k^{(t)}, \boldsymbol{\Sigma}_k^{(t)}\right)\right\}_{k=1}^{K}$ to $\theta^{(t+1)} = \left\{\left(\pi_k^{(t+1)}, \boldsymbol{\mu}_k^{(t+1)}, \boldsymbol{\Sigma}_k^{(t+1)}\right)\right\}_{k=1}^{K}$ in the following form:

$$\pi_k^{(t+1)} = \frac{N_k + \sum_{i:y_i=0} \delta_{ik}^{(t)}}{N}$$

$$\boldsymbol{\mu}_k^{(t+1)} = \frac{\sum_{i:y_i=k} \mathbf{x}_i + \sum_{i:y_i=0} \delta_{ik}^{(t)} \mathbf{x}_i}{N_k + \sum_{i:y_i=0} \delta_{ik}^{(t)}}$$

$$\boldsymbol{\Sigma}_k^{(t+1)} = \frac{\sum_{i:y_i=k} \left(\mathbf{x}_i - \boldsymbol{\mu}_k^{(t+1)}\right)\left(\mathbf{x}_i - \boldsymbol{\mu}_k^{(t+1)}\right)^T + \sum_{i:y_i=0} \delta_{ik}^{(t)} \left(\mathbf{x}_i - \boldsymbol{\mu}_k^{(t+1)}\right)\left(\mathbf{x}_i - \boldsymbol{\mu}_k^{(t+1)}\right)^T}{N_k + \sum_{i:y_i=0} \delta_{ik}^{(t)}}$$

where $N_k = \sum_{i:y_i=k} 1$ is the number of samples in class k. Please show your derivations.

2. What is the closed form expression of $\delta_{ik}^{(t)}$? Please show your derivations.

## Problem 5 (Label Propagation Algorithm)(1.5%)

In this problem, we will investigate label propagation algorithm by executing on a toy example. Next, we will show that the algorithm will convergence, which can be expressed analytically.

Let's consider the graph that we have seen in HW3 Problem 2.

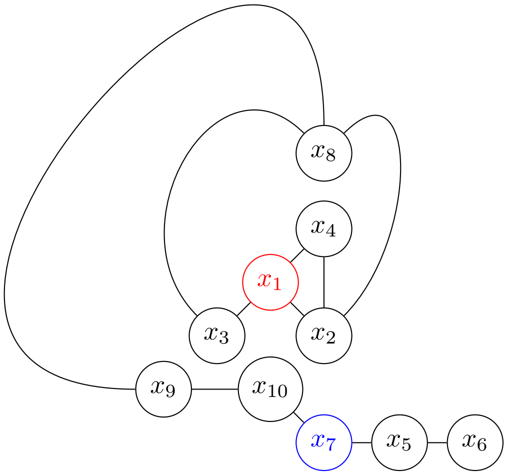

Figure 2: undirected connected graph $G$ with labeled node

We have previously known that $x_1$ node is in Class 1 and $x_7$ node is in Class 2. Now, we want to separate these 10 nodes into Class 1 and Class 2.

Consider the transition matrix $\boldsymbol{T}$,

$$\boldsymbol{T}_{i,j} = \frac{\tilde{\boldsymbol{W}}_{i,j}}{\sum_{k=1}^{10} \tilde{\boldsymbol{W}}_{k,j}}$$

, where $\tilde{\boldsymbol{W}}$ is the adjusted adjacency matrix of the graph G, which is defined as

$$\tilde{\boldsymbol{W}} = \boldsymbol{W} + \delta \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix}$$

. $\delta > 0$ is a small number. By adjusting the adjacency matrix, the weight of edge being connected in the original graph G is $1 + \delta$ and the weight of other edges is $\delta$. Through the adjustment, we can prevent that the algorithm runs unsupervised due to the isolated labeled nodes. In the toy example, we set $\delta = 0.01$. $\boldsymbol{T}_{i,j}$ represents the probability that node $j$ will propagate its own state to node $i$. For example, $\boldsymbol{T}_{2,1} = \boldsymbol{T}_{3,1} = \boldsymbol{T}_{4,1} \approx 0.326 \approx \frac{1}{3}, \boldsymbol{T}_{1,1} = \boldsymbol{T}_{5,1} = \boldsymbol{T}_{6,1} = \boldsymbol{T}_{7,1} = \boldsymbol{T}_{8,1} = \boldsymbol{T}_{9,1} = \boldsymbol{T}_{10,1} \approx 0.003$, which shows that node 1 will transfer its label to three neighbors with probability around 1 over 3. Also, it will transfer its label to other nodes(including itself) with probability slightly greater than zero.

1. Please write down the transition matrix $\boldsymbol{T}$.

Next, we define the label matrix sequence

$$\boldsymbol{Y}^t \in \mathbb{R}^{10\times 2} \quad t = 0, 1, ...$$

where the $i_{th}$ row of $\boldsymbol{Y}^t$ means the probability distribution of node $x_i$ at time $t$. In this example, $\boldsymbol{Y}^t_{i,1}$ means the probability that the node $x_i$ lies in Class 1 at time $t$, and $\boldsymbol{Y}^t_{i,2}$ means the probability that the node $x$ lies in Class 2 at time $t$. We initialize $\boldsymbol{Y}^0_{1,1} = 1, \boldsymbol{Y}^0_{1,2} = 0$ because $x_1$ is labeled as Class 1. Also, $\boldsymbol{Y}^0_{7,1} = 0, \boldsymbol{Y}^0_{7,2} = 1$ because $x_7$ is labeled as Class 2. For other nodes i, we initialize $\boldsymbol{Y}^0_{i,1} = \boldsymbol{Y}^0_{i,2} = 0.5$.

After defining label matrix $\boldsymbol{Y}$ and transition matrix $\boldsymbol{T}$, we introduce the algorithm below:

**Algorithm 1** Label Propagation Algorithm in Toy Example

```
Input label matrix $\boldsymbol{Y}^0$, transition matrix $\boldsymbol{T}$, tolerance level $\epsilon$
Output node $x_i \in \{$Class 1, Class 2$\}$     $i = 1, , , 10$
procedure Label Propagation($\boldsymbol{Y}^0$, $\boldsymbol{T}$, $\epsilon$=$10^{-8}$)
    t = 0
    repeat
        t = t + 1
        $\boldsymbol{Y}^t = \boldsymbol{T}\boldsymbol{Y}^{t-1}$                                  ▷ Random walk to its neighbor
        $\boldsymbol{Y}^t_{i,j} = \boldsymbol{Y}^t_{i,j}/(\boldsymbol{Y}^t_{i,1} + \boldsymbol{Y}^t_{i,2})$   $i = 1, ..., 10, j = 1, 2$   ▷ Normalize the probability distribution
        $\boldsymbol{Y}^t_{1,1} = 1, \boldsymbol{Y}^t_{1,2} = 0, \boldsymbol{Y}^t_{7,1} = 0, \boldsymbol{Y}^t_{7,2} = 1$     ▷ Clamp the labeled data
    until $||\boldsymbol{Y}^t - \boldsymbol{Y}^{t-1}||_F < \epsilon$
    If $\boldsymbol{Y}^t_{i,1} > 0.5$ then node $x_i$ lies in Class 1, otherwise node $x_i$ lies in Class 2,   $i = 1, ..., 10$
end procedure
```

There are three main procedures in the loop. First, every node propagates it own state to its neighbors with the transition probability. Next, we normalize the probability distribution for every node. In the last step, we clamp the probability distribution of the labeled data, which prevents the distribution of labeled data being influenced by unlabeled data and accelerates the convergence speed of the algorithm.

2. Please execute the algorithm. Write down the iteration number $t$, $\boldsymbol{Y}^t$, which nodes lies in Class 1 and which nodes lies in Class 2. Does the result correspond with the graph?

To show the convergence of label convergence algorithm, we consider more general case as the following statement.

Let $(x_1, y_1), ..., (x_l, y_l)$ be labeled data, where $y_i$ takes value in $\{1, .., C\}$, which indicates that $x_i$ lies in Class $y_i$. Also, we have $u$ unlabeled data $(x_{l+1}, y_{l+1}), ..., (x_{l+u}, y_{l+u})$, where $y_j$ is an unknown value, which lies in $\{1, ..., C\}$.

We can construct the transition matrix $\boldsymbol{T}$,

$$\boldsymbol{T}_{i,j} = \frac{\tilde{\boldsymbol{W}}_{i,j}}{\sum_{k=1}^{l+u} \tilde{\boldsymbol{W}}_{k,j}}$$

, where $\tilde{\boldsymbol{W}}_{i,j} > 0, \quad i, j = 1, ..., l + u$.

Also, we define the label matrix sequence

$$\boldsymbol{Y}^t \in \mathbb{R}^{(l+u)\times C} \quad t = 0, 1, ...$$

where the $i_{th}$ row of $\boldsymbol{Y}^t$ means the probability distribution of node $x_i$ at time $t$. $\boldsymbol{Y}^t_{i,j}$ means the probability that the node $x_i$ lies in Class $j$ at time $t$. For $\boldsymbol{Y}^0$, we clamp the first $l$ rows as following,

$$\boldsymbol{Y}^t_{i,j} = \mathbb{1}\{y_i = j\} \quad i = 1, ..., l, j = 1, ..., C$$

, which indicates that $x_i$ must lies in Class $y_i$. For the other rows, we initialize

$$\boldsymbol{Y}^t_{i,j} = \frac{1}{C} \quad i = l + 1, ..., l + u, j = 1, ..., C$$

We execute by the algorithm below.

**Algorithm 2** Generalized Label Propagation Algorithm

```
Input label matrix $\boldsymbol{Y}^0$ and transition matrix $\boldsymbol{T}$
Output $\boldsymbol{Y}^*$
procedure Generalize Label Propagation($\boldsymbol{Y}^0$, $\boldsymbol{T}$, $t = 0$)
    repeat
        t = t + 1
        $\boldsymbol{Y}^t = \boldsymbol{T}\boldsymbol{Y}^{t-1}$                                        ▷ Random walk to its neighbor
        $\boldsymbol{Y}^t_{i,j} = \boldsymbol{Y}^t_{i,j} / \sum_{k=1}^{C} \boldsymbol{Y}^t_{i,k} \quad i = 1, ...l + u, j = 1, ..., C$     ▷ Normalize the probability distribution
        $\boldsymbol{Y}^t_{i,j} = \mathbb{1}\{y_i = j\} \quad i = 1, ..., l, j = 1, ..., C$                 ▷ Clamp the labeled data
    until $\boldsymbol{Y}^t$ converges to $\boldsymbol{Y}^*$
end procedure
```

Once we get $\boldsymbol{Y}^*$, we can conclude that the unlabeled data $(x_i, y_i)$ belong to Class $y_i$, where

$$y_i = \arg\max_{\mathrm{J}}\ \boldsymbol{Y}^*_{i,j}$$

Next, we want to calculate $\boldsymbol{Y}^*$ analytically.

3. Please show that the line 4 and 5 in Algorithm 2 can be combined as

$$\boldsymbol{Y}^t = \overline{\boldsymbol{T}}\boldsymbol{Y}^{t-1}$$

, where $\overline{\boldsymbol{T}}_{i,j} = \boldsymbol{T}_{i,j} / \sum_{k=1}^{l+u} \boldsymbol{T}_{i,k}, i, j = 1, ..., l+u$.

We split $\overline{\boldsymbol{T}}$ to $\begin{bmatrix} \overline{\boldsymbol{T}}_{ll} & \overline{\boldsymbol{T}}_{lu} \\ \overline{\boldsymbol{T}}_{ul} & \overline{\boldsymbol{T}}_{uu} \end{bmatrix}$, where $\overline{\boldsymbol{T}}_{ll} \in \mathbb{R}^{l \times l}, \overline{\boldsymbol{T}}_{uu} \in \mathbb{R}^{u \times u}$. Also, we split $\boldsymbol{Y}^t$ to $\begin{bmatrix} \boldsymbol{Y}^t_L \\ \boldsymbol{Y}^t_U \end{bmatrix}$, where $\boldsymbol{Y}^t_L \in \mathbb{R}^l, \boldsymbol{Y}^t_U \in \mathbb{R}^u$

4. Please show that after an iteration,

$$\boldsymbol{Y}^t_U = \overline{\boldsymbol{T}}_{uu}\boldsymbol{Y}^{t-1}_U + \overline{\boldsymbol{T}}_{ul}\boldsymbol{Y}^{t-1}_L, t = 1, 2, ...,$$

$$\boldsymbol{Y}^t_L = \boldsymbol{Y}^{t-1}_L, t = 1, 2, ...$$

5. From the above result, we let $\boldsymbol{Y}_L = \boldsymbol{Y}^t_L$ for any $t$. Show that for $t \geq 1$

$$\boldsymbol{Y}^t_U = \overline{\boldsymbol{T}}^t_{uu}\boldsymbol{Y}^0 + \sum_{i=1}^{t} \overline{\boldsymbol{T}}^{i-1}_{uu}\overline{\boldsymbol{T}}_{ul}\boldsymbol{Y}_L$$

6. Please show that $\sum_{i=1}^{u} \overline{\boldsymbol{T}}_{\boldsymbol{uu}i,j} = \gamma_j$, where $0 < \gamma_j < 1$, for $j = 1, ..., u$. Use the fact to derive $\sum_{i=1}^{u} \overline{\boldsymbol{T}}^{\boldsymbol{n}}_{\boldsymbol{uu}i,j} \leq \gamma^n$, for $j = 1, ..., u$, where $\gamma = \max_{j=1,...,u} \gamma_j$. Last, derive $\lim_{n\to\infty} \overline{\boldsymbol{T}}^{\boldsymbol{n}}_{\boldsymbol{uu}} = \boldsymbol{O}$.

7. Define $\boldsymbol{S}_n = \boldsymbol{I} + \overline{\boldsymbol{T}}_{uu} + \overline{\boldsymbol{T}}^2_{uu} + ... + \overline{\boldsymbol{T}}^{n-1}_{uu}, \quad \boldsymbol{S}_n(\boldsymbol{I} - \overline{\boldsymbol{T}}_{uu}) = \boldsymbol{I} - \overline{\boldsymbol{T}}^n_{uu}$. Use the fact to show that $\lim_{n\to\infty} \boldsymbol{S}_n = (\boldsymbol{I} - \overline{\boldsymbol{T}}_{uu})^{-1}$. Combined all the result above, please show that $\boldsymbol{Y}^*_U = (\boldsymbol{I} - \overline{\boldsymbol{T}}_{uu}))^{-1}\overline{\boldsymbol{T}}_{ul}\boldsymbol{Y}_L$. Hence, $\boldsymbol{Y}^* = \begin{bmatrix} \boldsymbol{Y}_L \\ \boldsymbol{Y}^*_U \end{bmatrix}$, which can be obtained analytically regardless of the initial value $\boldsymbol{Y}^0$.

8. Please calculate the analytical solution $\boldsymbol{Y}^*$ on the toy example above. Does the solution correspond to the iteration solution $\boldsymbol{Y}^t$?

# Version Description

1. First Edition: Finish Problem 1 to 5
2. Second Edition: Revise the data description in Problem 5 Generalized Label Propagation.
3. Third Edition: Revise Problem 5 to make the definition more robust.