CS446: Machine Learning **Fall 2014**

Problem Set 7

Handed Out: November 18^{th}, 2014 *Due: December* 3^{rd}, 2014

- Feel free to talk to other members of the class in doing the homework. I am more concerned that you learn how to solve the problem than that you demonstrate that you solved it entirely on your own. You should, however, write down your solution yourself. Please try to keep the solution brief and clear.
- Please use Piazza first if you have questions about the homework. Also feel free to send us e-mails and come to office hours.
- Please, no handwritten solutions. You will submit your solution manuscript as a single pdf file.
- The homework is due at **11:59 PM** on the due date. We will be using Compass for collecting the homework assignments. Please submit an electronic copy via Compass2g (`http://compass2g.illinois.edu`). Please do NOT hand in a hard copy of your write-up. Contact the TAs if you are face technical difficulties in submitting the assignment.
- You cannot use the late submission credit hours for this problem set.
- No code is needed for any of these problems. You can do the calculations however you please. You need to turn in only the report. Please name your report as ⟨`NetID`⟩`-hw7.pdf`.

1. [EM Algorithm - 70 points]

Given a collection of documents $\{d_1, d_2, \ldots, d_M\}$ where each document consists of words from vocabulary $\{w_1, w_2, \ldots, w_V\}$, we want to cluster this collection into two categories, c_1 and c_2. In this model, words and documents are observed variables and the category assignment of the words are latent variables. Each category has a multinomial distribution over the vocabulary and each document has a binomial distribution over the categories. Let us first introduce the model parameters:

- $P(d_i)$ is the probability of observing a particular document d_i.
- $P(c_k|d_i)$ is the probability that the document d_i has category c_k.
- $P(w_j|c_k)$ is the probability that word w_j appears in the category c_k.

Using these definitions, we can think about the generative process that resulted in the observed collection of documents as follows:

1. Pick a document d_i with probability $P(d_i)$. We generate M documents.
2. For each word position in the document d_i, pick a category c_k with probability $P(c_k|d_i)$.
3. For each word position, we generate a word from $\{w_1, \ldots, w_V\}$ based on the category assignment c_k. That is, we generate a word w_j according to $P(w_j|c_k)$.

One way to estimate the parameters of this model is to use the EM algorithm. We are going to guild you through the steps of the EM algorithm for this model. Please use the notations defined above to answer following questions.

(a) **[10 points]** What is the probability of observing a word in a document (in terms of the variables introduced above), $P(w_j, d_i)$?

Potentially, a clearer way to formulate this question is: What is the probability of observing w_j in d_i at a particular position of d_i?

From the generative process of this model, the probability of picking the document d_i is $P(d_i)$. Given d_i, the category distribution for the word w_j is $P(c_k|d_i)$, which is independent of the category assignment of other words. After picking a category c_k, the probability of seeing the word w_j is governed by $P(w_j|c_k)$. The joint probability of seeing a particular assignment of d_i, c_k, and w_j is thus $P(d_i, c_k, w_j) = P(d_i)P(c_k|d_i)P(w_j|c_k)$. To get $P(d_i, w_j)$, we sum up all possible values of c_k:

$$\begin{aligned} P(w_j, d_i) &= \sum_{k=1}^{2} P(w_j, d_i, c_k) \\ &= \sum_{k=1}^{2} A_i B_{ki} C_{jk} \\ &= A_i \sum_{k=1}^{2} B_{ki} C_{jk} \end{aligned}$$

where $A_i = P(d_i)$, $B_{ki} = P(c_k|d_i)$, and $C_{jk} = P(w_j|c_k)$.

(b) **[10 points]** In the E-step, we estimate the posterior distribution of the latent variables given the current parameters. Derive $P(c_k|w_j, d_i)$.

$$\begin{aligned} P(c_k|w_j, d_i) &= \frac{P(c_k, w_j, d_i)}{P(w_j, d_i)} \\ &= \frac{A_i B_{ki} C_{jk}}{A_i \sum_{k=1}^{2} B_{ki} C_{jk}} \\ &= \frac{B_{ki} C_{jk}}{\sum_{k=1}^{2} B_{ki} C_{jk}} \end{aligned}$$

where $A_i = P(d_i)$, $B_{ki} = P(c_k|d_i)$, and $C_{jk} = P(w_j|c_k)$.

(c) **[15 points]** In the M-step, we maximize the expected complete data log-likelihood $E[LL]$ of the entire collection of documents. Derive $E[LL]$. (Please use $n(d_i, w_j)$ to denote the number of occurrences of w_j in document d_i. Note that it's possible that $n(d_i, w_j) = 0$ for some i and j.)

$$\begin{aligned}E[LL] =& E_{c_k}[\log \prod_{i=1}^{M}\prod_{j=1}^{V} P(d_i, w_j, c_k)^{n(d_i,w_j)}]\\ =& E_{c_k}[\sum_{i=1}^{M}\sum_{j=1}^{V} n(d_i, w_j) \log P(d_i, w_j, c_k)]\\ =& \sum_{i=1}^{M}\sum_{j=1}^{V} n(d_i, w_j) \sum_{k=1}^{2} P(c_k|d_i, w_j)[\log A_i + \log B_{ki} + \log C_{jk}]\end{aligned}$$

where $A_i = P(d_i), B_{ki} = P(c_k|d_i)$, and $C_{jk} = P(w_j|c_k)$.

(d) **[20 points]** Solve the optimization problem you formulated in (c) to derive the update rules for $P(d_i)$, $P(c_k|d_i)$ and $P(w_j|c_k)$.

The optimization problem here is

$$\begin{aligned}\max \quad & E[LL]\\ \text{subject to} \quad & \sum_{i=1}^{M} A_i = 1\\ & \sum_{k=1}^{2} B_{ki} = 1, \forall i \in \{1, \dots, M\}\\ & \sum_{j=1}^{V} C_{jk} = 1, \forall k \in \{1, 2\}\end{aligned}$$

Using the Lagrange multiplier method, we get the following objective function with constraints:

$$L = E[LL] + \alpha(1 - \sum_{i=1}^{M} A_i) + \sum_{k=1}^{2} \beta_k(1 - \sum_{j=1}^{V} C_{jk}) + \sum_{i=1}^{M} \gamma_i(1 - \sum_{k=1}^{2} B_{ki})$$

Take partial derivative of L with respect to A_i and set it to zero.

$$\begin{aligned}\frac{\partial L}{\partial A_i} =& \frac{\sum_{j=1}^{V} n(d_i, w_j)}{A_i} - \alpha = 0\\ \Rightarrow A_i =& \frac{\sum_{j=1}^{V} n(d_i, w_j)}{\alpha}\end{aligned}$$

Since $\sum_{i=1}^{M} A_i = 1$, we have $\alpha = \sum_{i=1}^{M}\sum_{j=1}^{V} n(d_i, w_j)$. Therefore,

$$A_i = \frac{\sum_{j=1}^{V} n(d_i, w_j)}{\sum_{i=1}^{M}\sum_{j=1}^{V} n(d_i, w_j)}$$

Similarly, we can get

$$B_{ki} = \frac{\sum_{j=1}^{V} n(d_i, w_j) P(c_k|w_j, d_i)}{\sum_{j=1}^{V} n(d_i, w_j)}$$

$$C_{jk} = \frac{\sum_{i=1}^{M} n(d_i, w_j) P(c_k|w_j, d_i)}{\sum_{j=1}^{V} \sum_{i=1}^{M} n(d_i, w_j) P(c_k|w_j, d_i)}$$

(e) [**15 points**] Examine the update rules and explain them in English. Also, describe in pseudocode how would you run the algorithm: initialization, iteration, and termination. What equations in the previous answers would you use at which steps in the algorithm?

The estimation of $P(d_i)$ is the number of words in d_i over the number of words in the whole collection. $P(w_j|c_k)$ is the proportion of w_j being assigned to c_k over the proportion of all words being assigned to c_k. $P(c_k|d_i)$ is the proportion of words in d_i assigned to c_k over the total number of words in d_i.

The algorithm:

i. Initialize with random values for the parameters, $P(d_i), P(c_k|d_i)$, and $P(w_j|c_k)$, such that the constraints of the optimization problem in question (d) are satisfied.

ii. Compute the posterior distribution of latent variables, $P(c_k|d_i, w_j), \forall i, j, k$, as shown in (b).

iii. Update parameters by the results in (d).

iv. Repeat (ii) and (iii) until convergence.

2. [**Tree Dependent Distributions - 30 points**]

Note: In this problem, we will be looking at tree dependent distributions that will be covered in class soon. You may go through the lecture notes or wait for it to be taught in class before you attempt this problem. A brief introduction is given below.

A tree dependent distribution is a probability distribution over n variables, $\{x_1, \ldots, x_n\}$ that can be represented as a tree built over n nodes corresponding to the variables. If there is a directed edge from variable x_i to variable x_j, then x_i is said to be the parent of x_j. Each directed edge $\langle x_i, x_j \rangle$ has a weight that indicates the conditional probability $\Pr(x_j \mid x_i)$. In addition, we also have probability $\Pr(x_r)$ associated with the root node x_r. While computing joint probabilities over tree-dependent distributions, we assume that a node is independent of all its non-descendants given its parent. For instance, in our example above, x_j is independent of all its non-descendants given x_i.

To learn a tree-dependent distribution, we need to learn three things: the structure of the tree, the probabilities on the edges of the tree, and the probabilities on the nodes. Assume that you have an algorithm to learn an *undirected* tree T with all required probabilities. To clarify, for all *undirected* edges $\langle x_i, x_j \rangle$, we have learned both probabilities, $\Pr(x_i \mid x_j)$ and $\Pr(x_j \mid x_i)$. (There exists such an algorithm and we will be covering that in class.) The only aspect missing is the directionality of edges to convert this undirected tree to a directed one.

However, it is okay to not learn the directionality of edges explicitly. In this problem, you would show that choosing any arbitrary node as the root and directing all edges away from it is sufficient, and that two directed trees obtained this way from the same underlying undirected tree T are equivalent.

(a) [**10 points**] State exactly what is meant by the statement: "*The two directed trees obtained from T are equivalent.*"

Two directed trees T_0 and T_1 over variables $x_1, \ldots, x_n$ are equivalent iff the joint probability distributions they represent are the same. In other words:

$$\Pr_{T_0}(x_1, \ldots, x_n) = \Pr_{T_1}(x_1, \ldots, x_n)$$

This implies that for every event E over $x_1, \ldots, x_n$, $\Pr_{T_0}(E) = \Pr_{T_1}(E)$.

(b) [**20 points**] Show that no matter which node in T is chosen as the root for the "direction" stage, the resulting directed trees are all equivalent (based on your definition above).

Let T_i and T_j be the two directed trees obtained by choosing two different roots x_i and x_j ($i \neq j, 1 \leq i, j \leq n$) from the undirected tree T. Denoting $\mathbf{x} = (x_1, \ldots, x_n)$, we would like to show that $\Pr_{T_i}(\mathbf{x}) = \Pr_{T_j}(\mathbf{x})$.
Let π_{x_k} be the parent of node x_k. Note that there is a unique path $\mathcal{P}$ between nodes i and j. Assume for now that the path is of length 1. That is, there is an edge in T between x_i and x_j. Thus, the only difference between T_i and T_j is the direction of this edge (convince yourself of that).

$$\begin{aligned}
\Pr_{T_i}(\mathbf{x}) &= \Pr(x_i) \prod_{\substack{k=1 \\ k \neq i}}^{n} \Pr(x_k \mid \pi_{x_k}) \\
&= \Pr(x_i) \Pr(x_j \mid x_i) \prod_{\substack{k=1 \\ x_k \notin \mathcal{P}}}^{n} \Pr(x_k \mid \pi_{x_k}) \\
&= \Pr(x_i, x_j) \prod_{\substack{k=1 \\ x_k \notin \mathcal{P}}}^{n} \Pr(x_k \mid \pi_{x_k}) \\
&= \Pr(x_j) \Pr(x_i \mid x_j) \prod_{\substack{k=1 \\ x_k \notin \mathcal{P}}}^{n} \Pr(x_k \mid \pi_{x_k}) \\
&= \Pr(x_j) \prod_{\substack{k=1 \\ k \neq j}}^{n} \Pr(x_k \mid \pi_{x_k}) \\
&= \Pr_{T_j}(\mathbf{x})
\end{aligned}$$

Next, notice that if the path $\mathcal{P}$ between x_i and x_j is longer, we maintain the property that the edges not on the path $\mathcal{P}$ have the same directionality in T_i and T_j. We can therefore use the argument above inductively, transforming a tree rooted at x_i to one rooted at x_j by switching the directions of the edges in $\mathcal{P}$ one edge at a time. As shown above, each of these steps maintains the equivalent joint distribution.