CS446: Machine Learning	**Fall 2015**

Problem Set 7

Handed Out: November 19^{th}*,* 2015 *Due: December* 3^{rd}*,* 2015

- 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 face technical difficulties in submitting the assignment.
- You may only use 24 hours of 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]**

   Suppose there is a collection of documents $\{d_1, d_2, \ldots, d_M\}$ where each document consists of words from the vocabulary $\{w_1, w_2, \ldots, w_V\}$. Let D be the random variable representing documents that takes values in $\{d_1, d_2, \ldots, d_M\}$, and let W be the random variable representing words that takes values in $\{w_1, w_2, \ldots, w_V\}$. We observe the documents and words, but each word in each document has an unobserved category, c_1 or c_2, which we represent with the random variable $C \in \{c_1, c_2\}$. The words of the vocabulary are drawn according to some probability distribution given the category, and the categories are drawn according to some probability distribution given the document. Thus, a given document may have multiple categories (either c_1 or c_2 for each word) and the model can be represeneted graphically as

   $$D \rightarrow C \rightarrow W.$$

   The model parameters are:

   - $P(D = d_i)$ is the probability of selecting a particular document d_i.
   - $P(C = c_k | D = d_i)$ is the probability that the document d_i has category c_k at an arbitrary word position.
   - $P(W = w_j | C = c_k)$ 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:

   0. Begin with M completely empty documents.

1. Sample a document d_i with probability $P(D = d_i)$.
2. Sample an unobserved category c_k with probability $P(C = c_k|D = d_i)$.
3. Add to d_i a single word, w_j, sampled with probability $P(W = w_j|C = c_k)$.
4. Repeat steps $1 - 3$ or stop generating words.

Now given this data, i.e. the resulting documents of words after discarding categories, we want to estimate the parameters of our model. One way to do this is to use the EM algorithm. We are going to guide 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 $P(W = w_j, D = d_i)$, the probability of observing the word w_j within document d_i at an arbitrary position?

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

(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.)

(d) **[20 points]** Solve the optimization problem you formulated in (c) to derive the update rules for $P(D = d_i)$, $P(C = c_k|D = d_i)$ and $P(W = w_j|C = c_k)$. (You only need to show the full work for one derivation of your choosing.)

(e) **[10 points]** Examine the update rules and explain what they represent in English.

(f) **[5 points]** Describe in pseudocode how you would run the algorithm: initialization, iteration, and termination.

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

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 conditional 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 the edges explicitly. In this problem, you will 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.*"

(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).