CS446: Machine Learning **SPRING 2017**

Problem Set 7

Handed Out: April 20^{th}, 2017 *Due: May* 1^{st}, 2017

- 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]**

   Assume we have a set D of m data points, where for each data point x from D, $x \in \{0,1\}^{n+1}$. Denote the i-th bit of the j-th example as $x_i^{(j)}$. Thus, the index i ranges from $0 \ldots n$, and the index j ranges from $1 \ldots m$.

   Assume these data points were generated according to the following distribution:

   Postulate a hidden random variable Z with values $z = 1, 2$, where the probability of $z = 1$ is α and the probability of $z = 2$ is $1 - \alpha$, where $0 < \alpha < 1$.

   For a specific example $x^{(j)}$, a random value of Z is chosen, but its true value z is hidden. Note that each example $x^{(j)}$ has a fixed underlying z. If $z = 1$, $x_i^{(j)}$ is set to 1 with probability p_i. If $z = 2$, the bit is set to 1 with probability q_i. Thus, there are $2n + 3$ unknown parameters. You will use EM to develop an algorithm to estimate these unknown parameters.

   (a) **[10 points]** Express $\Pr(x^{(j)})$ first in terms of conditional probabilities and then in terms of the unknown parameters α, p_i, and q_i.

   (b) **[10 points]** Let $f_z^{(j)} = \Pr(Z = z \mid x^{(j)})$, i.e. the probability that the data point $x^{(j)}$ has z as the value of its hidden variable Z. Express $f_1^{(j)}$ and $f_2^{(j)}$ in terms of the unknown parameters.

   (c) **[10 points]** Derive an expression for the expected log likelihood ($E[LL]$) of the entire data set D and its associated z settings given new parameter estimates $\tilde{\alpha}, \tilde{p}_i, \tilde{q}_i$.

   (d) **[10 points]** Maximize the log likelihood (LL) and determine the update rules for the parameters according to the EM algorithm.

(e) **[10 points]** Examine the update rules explain them in English. Describe in pseudocode how you would run the algorithm: initialization, iteration, termination. What equations would you use at which steps in the algorithm?

(f) **[10 points]** Assume that your task is to predict the value of x_0 given an assignment to the other n variables and that you have the parameters of the model. Show how to use these parameters to predict x_0. (*Hint:* Consider the ratio between $P(X_0 = 0)$ and $P(X_0 = 1)$.)

(g) **[10 points]** Show that the decision surface for this prediction is a linear function of the x_i's.

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).