CS446: Machine Learning **Fall 2015**

Problem Set Problem set 7

Shivam Upadhyay *Handed In: December 4, 2015*

1. **EM Algorithm**
For convenience I'll use the notation $P(W = w_j, D = d_i) \equiv P(w_j, d_i)$

a.

$$P(w_j, d_i) = P(d_i)P(w_j|d_i) \qquad \text{[Applying the chain rule]}$$

Now, we know that $P(w_j|d_i)$ is the probability of a word give the document but with the generative model we know that for give value of c_k, the former distribution is given by $P(c_k|d_i)P(w_j|c_k)$ thus marginalising over it will yield the desired distribution.

$$= P(d_i)\sum_{k=1}^{2} P(c_k|d_i)P(w_j|c_k) \qquad \text{[Marginalising over } c_k\text{]}$$

b.

$$P(c_k|w_j, d_i) = \frac{P(w_j, d_i|c_k)P(c_k)}{P(w_j, d_i)} \qquad [P(A|B) = P(B|A)P(A)/P(B)]$$

$$= \frac{P(d_i|c_k)P(w_j|c_k)P(c_k)}{P(d_i)\sum_{k=1}^{2} P(c_k|d_i)P(w_j|c_k)} \quad \text{[by 1 independent assumption between } w_j \text{ and } d_i\text{]}$$

Again,

$$= \frac{P(c_k|d_i)P(w_j|c_k)\cancel{P(c_k)P(d_i)}}{\cancel{P(c_k)P(d_i)}\sum_{k=1}^{2} P(c_k|d_i)P(w_j|c_k)} \quad [P(A|B) = P(B|A)P(A)/P(B)]$$

$$\boxed{P(c_k|w_j, d_i) = \frac{P(c_k|d_i)P(w_j|c_k)}{\sum_{k=1}^{2} P(c_k|d_i)P(w_j|c_k)}}$$

c. As w_j and d_i are observed variables we know that for likelihood of the i^{th} document is given by:

$$\prod_j P(d_i, w_j)^{n(d_i, w_j)}$$

That is product of probability of each word appearing in a document where $n(d_i, w_j)$ is the number of times a particular word appear in that document.
For the entire data $d_1, d_2 \cdot d_m$ the likelihood will be given by product of the probabilities of all the documents.
Therefore,

$$L = \prod_i \prod_j P(d_i, w_j)^{n(d_i, w_j)}$$

So then log-likelihood is:

$$LL = \sum_i \sum_j n(d_i, w_j) log[P(d_i, w_j)]$$

And then expected value of the log-likelihood with respect to the posterior:

$$E[LL] = \sum_i \sum_j n(d_i, w_j) E[log[\sum_k P(d_i)P(c_k|d_i)P(w_j|c_k)]]$$

$$= \textstyle\sum_i \sum_j n(d_i, w_j)[P(c_1|w_j, d_i) log[P(w_j|c_1)P(c_1|d_i)p(d_i)] + \\ P(c_2|w_j, d_i) log[P(w_j|c_2)P(c_2|d_i)P(d_i)]]$$

d. From part c equation is:

$$f : E[LL] = \textstyle\sum_i \sum_j n(d_i, w_j)[P(c_1|w_j, d_i) log[P(w_j|c_1)P(c_1|d_i)P(d_i)] \\ + P(c_2|w_j, d_i) log[P(w_j|c_2)P(c_2|d_i)P(d_i)]$$

With constraints:
$1 : \sum P(c_k|d_i) = 1$
$2 : \sum P(w_j|c_k) = 1$
$3 : \sum P(d_i) = 1$
Embedding the constraints in the equation using Lagrange's multiplier $\lambda_1, \lambda_2, \lambda_3$ we get:

$$\textstyle\sum_i \sum_j n(d_i, w_j) \Bigg[[P(c_1|w_j, d_i) log[P(w_j|c_1)P(c_1|d_i)P(d_i)] \\ + P(c_2|w_j, d_i) log[P(w_j|c_2)P(c_2|d_i)P(d_i)] \Bigg] + \\ \lambda_1 \left[\sum_j P(w_j|c_1) - 1 \right] + \lambda_2 \left[\sum_j P(w_j|c_2) - 1 \right] + \lambda_3 \left[\sum_i P(d_i) - 1 \right]$$

Taking the partial derivative w.r.t $P(d_i)$ and setting it to 0:

$$\textstyle\sum_j \left(n(d_i, w_j)P(c_1|w_j, d_i) \frac{1}{P(d_i)} + n(d_i, w_j)P(c_1|w_j, d_i) \frac{1}{P(d_i)} \right) + \lambda_3 = 0 \quad (1)$$

we know that $P(c_1|w_j, d_i) + P(c_2|w_j, d_i) = 1$

$$\rightarrow \sum_j n(d_i, w_j) \frac{1}{P(d_i)} + \lambda_3 = 0$$

$$\rightarrow \sum_j n(d_i, w_j) = -P(d_i)\lambda_3 \qquad (1)$$

summing across all the documents:

$$\rightarrow \sum_i \sum_j n(d_i, w_j) = \sum_i -P(d_i)\lambda_3$$

Since, $\sum_i P(d_i) = 1$

$$\rightarrow \lambda_3 = -\sum_i \sum_j n(d_i, w_j) \qquad (2)$$

Putting value from (2) to (1):

$$\rightarrow \sum_j n(d_i, w_j) = -P(d_i)\Big(-\sum_i \sum_j n(d_i, w_j)\Big)$$

$$\rightarrow \boxed{P(d_i) = \frac{\sum_j n(d_i, w_j)}{\sum_i \sum_j n(d_i, w_j)}}$$

similarly,

$$\boxed{P(w_j|c_k) = \frac{\sum_i n(d_i, w_j)P(c_k|d_i, w_j)}{\sum_i \sum_j n(d_i, w_j)P(c_k|d_i, w_j)}}$$

$$\boxed{P(c_k|d_i) = \frac{\sum_j n(d_i, w_j)P(c_k|d_i, w_j)}{\sum_j n(d_i, w_j)}}$$

e. $P(w_j|c_k)$ is given by iterating through all the documents, and counting how many times w_j has appeared with category c_k divided by the total number of appearances of c_k in all of the documents.
$P(c_k|d_i)$ is given by iterating through the given document d_i, and counting how many times the category c_k appeared, and dividing it by the total number of words in the document.
$P(d_i)$ is given by total number of words in document d_i divided by total number of words in all the document.

f. **Algorithm:**

```
1. Set the parameters to some guessed values $P(d_i)$, $P(c_k|d_i)$, $P(w_j|c_k)$
2. While(not converged)
3.      Find posterior $P(c_k|w_j, d_i)$ of the latent variable, using part b.
4.      Calculate the expected value of the log-likelihood, with respect to the
        posterior, using current parameters and the equation from part c.
5.      Maximize the expected value, using part d.
6.      Set parameters to the new values.
7. Return final parameters.
```

2. **Tree Dependent Distributions**

   a. "The two directed trees obtained from T are equivalent" It means that the choice of root node is irrelevant for the final joint probability distribution of the tree i.e $P(D|T_1) = P(D|T_2)$ Generally, it means:
   $\forall x_r$ (root node) , in the graph, all $P(x_r)\prod_{x_i \in T-\{x_r\}} P(x_i)P(x_i|Parent(x_i))$ are equal.

   b. Chow-Liu Algorithm is used for finding the tree that maximises the likelihood of the given data. It works by maximising the sum of information gain which in turn reduces the distributional distance for given data.
   Information gain is given by $I(x,y) = \sum_{x,y} P(x,y)log\frac{P(x,y)}{P(x)P(y)}$ Given a DAG ,we see that if we were to use either $P(x_i|x_j)$ or $P(x_j|x_i)$, it wouldn't matter, since information gain I is same for both cases. And since we know that either one of the two (for all i and j) will be included in the calculation, the resulting joint probability distribution is the same, no matter where the starting root is. It can be further understood by taking the set S of all the parents in the given tree, we know that probability of any data point is given by:

   $$P(x_r)\prod_{\forall edges(i,j)} P(x_i|x_j) = \prod_{\forall edges(i,j)} P(x_i,x_j)\frac{P(x_r)}{P(x_i)}$$

   Using the chain rule and moving in $P(x_r)$, we get the above equation. Now lets S be the set of all parent nodes (nodes which act as source of the edge in a directed graph). Now this set is the denominator in the above fraction $\forall edges(i,j)p(x_i)$ Let there be n_i edges being associated to i^th node, moreover a node can be a root or source of other edges.
   Scenario 1: When the i^th node is root, then it will be added $n_i - 1$ times in the set for the edges originating from it -1 as it is the root(in the above equation it also gets cancelled by the numerator x_r.
   Scenario 2: When the i^th node is not a root, then it will be added $n_i - 1$ times in the set for the edges originating from it -1 for one incoming edge from the root(As it is a DAG).
   Thus no matter how we choose the origin the set S remains same i.e. the number of edges associated with a node remains same and hence all the trees are quivalent.