CSC2515 – Assignment #3

Due: Nov.20, 3pm at the **START** of class
Worth: 18%
Late assignments not accepted.

1 Mixture of Base Rates (7% + up to 5% bonus)

In this question you will derive the EM algorithm for a very simple model called "mixture of base rates", which is the unsupervised equivalent of Naive Bayes classification. We model some discrete data $\mathbf{x}$ by assuming that there is an *unobserved* cluster variable k and that given this cluster variable, all the components x_i of $\mathbf{x}$ are conditionally independent. The equation is:

$$p(\mathbf{x}) = \sum_k p(k) \prod_i p(x_i|k) = \sum_k \alpha_k \prod_i \prod_{j \in V_i} a_{ijk}^{[x_i=j]}$$

where α_k is the prior probability of cluster k, a_{ijk} are the model parameters ($\sum_j a_{ijk} = 1 \ \forall i, k$), V_i is the set of possible values for the discrete variable x_i (e.g. if x_i is binary then $V_i = \{0, 1\}$), and the indicator $[x_i = j]$ is 1 if x_i takes on value j and 0 otherwise.

- Write down the complete (joint) log likelihood $p(\mathbf{x}, k)$.
- Derive the E-step of the EM algorithm by writing down the expression for the posterior probability of cluster k, $p(k|\mathbf{x}^n)$ given a particular datapoint $\mathbf{x}^n$. Hint: use Bayes' rule.
- Write down the expected complete log likelihood using the posterior probability you just derived: $\sum_n \sum_k p(k|\mathbf{x}^n) \log p(\mathbf{x}^n, k)$
- Derive the M-step of the EM algorithm by taking the derivative of this expected complete log likelihood with respect to the parameters a_{ijk}. Don't forget to enforce the normalization constraint that $\sum_j a_{ijk} = 1 \ \forall i, k$.
- **Bonus1**: Also find the updates for the priors α_k.
- **Bonus2**: Assuming you have computed $q_k = \log \alpha_k + \sum_i \sum_{j \in V_i} [x_i = j] \log a_{ijk}$, how would you compute $p(k|x)$ in a numerically stable way using `logsum`?
- **Bonus3**: Fill in the ??:
  For continuous vector valued data $\mathbf{x}$, the analog of this model is a "?? of ?? with diagonal ??".

2 Fully Observed Trees (11%)

In this assignment you will train a fully observed tree model on word-document vectors taken from USENET articles.

2.1 What to do

- Using the data provided in `a3newsgroups.mat`, train a fully observed tree model with the optimal structure. Training the model structure should give you an *undirected* tree. Using this, you can pick a root node arbitrarily and form a *directed tree.*
- The matrix `documents` contains one column per USENET article.
  Each of the 100 rows is a binary variable indicating whether a certain keyword appears in the posting or not. The cell array `wordlist` has the 100 words corresponding to these rows. The data has 100 (binary) features and 16242 training cases.

2.2 What to hand in

- Any *compact, clear and unambiguous* representation of the optimal (undirected) tree structure found by your algorithm. For example, you might print out all 99 pairs of words `(word_i--word_j)` corresponding to edges chosen by the tree learning algorithm. Or you might want to draw a picture of the graph (see below).
  DO NOT hand in lists of numbers corresponding to the indices of pairs of features chosen – convert everything back to words using the `wordlist` cell array. Since the order of the words is arbitrary, the number of the word index is meaningless, and printing it out tells someone else nothing about your model.
- After finding the optimal tree structure for the 100 variables, you may choose any node you wish as the root and form a directed graph. Different choices of root node will result in different output when you visualize a directed tree, but of couse the undirected model structure always remains the same. Chose a root word. Hand in a *compact, clear and unambiguous* representation of a directed tree structure created by directing all arcs in the optimal structure away from a your chosen root. For example, you might print out a breadth-first list of directed links `(word_i->word_j)` in the tree. Alternately, you might want to draw a picture (see below).
- A histogram (with at least 100 bins) of the log likelihoods of the training cases under your optimal model. Notice that these likelihoods are the same no matter how you chose to convert your undirected tree into a directed tree.
  Compute and print out the min,max,mean and median log likelihood.
  Which training case (give its number) has the worst (lowest) log likelihood?
  Print out the list of keywords appearing in this posting.

2.3 Some hints

- Be careful when computing the mutual information weights for pairs of variables that have zero counts for some of their joint settings. In this case the mutual information should be zero, but a careless calculation will lead to division by zero or log of zero errors in the code.
- To save you some trouble, I've written the function `mwst.m` for you, which finds minimum or maximum weight spanning trees given a weight matrix.
- Here is a trick for computing the 2 by 2 joint count table of two binary column vectors `v1` and `v2`:
  ```
  counts12 = reshape(hist(v1+2*v2,0:3),2,2);
  ```
  but make sure you understand it before using it!
- The data is almost a megabyte in MATLAB , stored as a sparse matrix, so be careful about making copies of it in memory. You should be able to do everything you need without making another (sparse or nonsparse) copy of the whole dataset.
- In case you are interested, the array `newsgroup` tells you the toplevel newsgroup category of each posting, `1=comp.*, 2=rec.*, 3=sci.*, 4=talk.*`, although we won't use it in this assignment.

2.4 Reminders

Amongst other things, your code will need to:

- Compute the marginal count for every feature being on or off summed across all documents.
- Compute the joint counts for every pair of features being on together, off together, on-off or off-on, summed across all documents.
- Convert these counts into mutual information weights.
  Be careful to do this properly for joint counts which are zero!
- Find the best spanning tree over the features given these weights. Depending on how you compute the weights, you will either need a minimum or maximum weight spanning tree.
  (This will in turn affect how you call `mwst.m` if you choose to use it.)
- Select a root for the tree and direct all edges away from it to get a directed graph.

2.5 Automatic Graph Layout

- If you want to be really fancy, you can try using the function `dottree.m` and the automatic graph layout programs `dot` and `neato` to draw pictures of your final structures. It is very cool looking, but you will not get any more or any fewer marks for using this method or a simpler method to display your results.
  Caution: do not waste too much time trying to get this to work, since it doesn't get you any more marks. It is just for fun. Get the rest of the assignment finished first.
- Call `dottree(tree,'fname',wordlist)` and it will write out two files, `fname.u` and `fname.d`.
  Now call the `neato` and `dot` programs to convert these files into pictures.
- Information and downloads for `dot` and `neato` are available at `http://www.graphviz.org/`. The programs are already installed in `/local/bin/` on CDF and in `/pkgs/graphvis/linux/bin/` on CSLAB.