Statistical Natural Language Processing, SS 16
Assignment 08

Ho Vinh Thinh (2562630)
Prabal Agarwal (2561419)
Nurzat Rakhmanberdieva (2561148)

Problem 1

Results for the EM algorithm are as follows:

```
Iteration - 580, log likelihood value = -203026.609524
('ordered list of senses within cluster', [('HARD1', 819), ('HARD3', 93), ('HARD2', 90)])
('ordered list of senses within cluster', [('HARD1', 1900), ('HARD2', 146), ('HARD3',
88)])
('ordered list of senses within cluster', [('HARD1', 736), ('HARD2', 266), ('HARD3', 195)])
```

Hence, as observed from the composition of the three clusters, the EM algorithm does not clusterize the data well.

Does EM find the global optimum?

EM does not guarantee to find a global optimum. If we take the derivative of likelihood at converged point, it is either maxima or saddle point. There can be a number of local maxima values of the objective function. Because of such local optima points, there is a possibility that the algorithm terminates before reaching the global optima.

In this case, the number of clusters was set to the number of senses provided in the corpus. In a real unsupervised scenario, this is not given. How would you decide on the number of clusters?

When the number of clusters not given, Bayesian Information Criterion approach can be used.
When fitting models, it might be reasonable to add more parameters as it increases the likelihoods. But this will result in over-fitting and model will only work for particular case. That's why it is wise to use BIC.

$$BIC = -2 \ln L + k * \ln(n) \tag{1}$$

We choose different values of K and compute BIC. The goal is to minimize BIC, which penalizes model for complexity, and doing so, it avoids over-fitting and finds the most suitable size of K which is cluster number.

For all other tasks in this assignment, code provided.

Bonus

Explain how the information in matrix C could be used to train a classifier (e.g. Support vector machine or KNN). What would be your features and targets?

C is the matrix of vocab instances and context where they are present. The matrix C could be used in support vector machine to build the boundary that separates classes well.

words in vocabulary can be denoted as terms which are features
contexts can be denoted as documents which are targets that need to be correctly classified

This term*doc matrix will be projected onto the new vector space. Using kernels with different pairwise similarity for these matrix entries, we can draw decision boundaries for classes.

For example: X = $(X_1, X_2, ... X_n)^T$
X vector are words in particular context. Using them, we will build the support vector classification model by choosing β values that reduces model error rate with provided information about the true label of given observations.