Joshua Sloan
University of Wyoming
jsloan3@uwyo.edu

Homework 4
COSC 5010 - Data Mining

1 Problems from the Book

1.1 Chapter 8 - Problem 18

None of the 3 methods can guarantee to produce a global minimum (at least from a single trial, perhaps something can be said about a mixed approach with brute force initialization selection). If there were such a method, there would be no reason to use any others. It is certainly possible that a local minimum achieved might be the global minimum, although there is no guarantee that it will be.

That being said, ordinary K-means converges to produce a local minimum.

Bisecting K-means uses the K-Means algorithm locally to bisect individual clusters, however, this does not mean that the result also produces a local minimum. Yet, while the baseline algorithm may not produce a local minimum (in terms of SSE), we are still able to obtain a local minimum by adding the extra refinement step to the algorithm through using the obtained cluster centroids as the initial centroids for the standard K-Means algorithm.

Similarly, although Ward's method uses the same objective function as K-means clustering, it also cannot guarantee a local minimum without further refinement steps being added to the baseline algorithm. In the standard algorithm, minimizing the squared error for the cluster when merging clusters may not correlate to the overall local minimum. Once merged, it cannot be undone. It is very possible that a less ideal merge initially may lead to an overall reduction in SSE. The book suggests that partially clustering the data using another method (such as standard K-means) might help alleviate these issues, but on its own we cannot claim that Ward's method produces a local minimum.

1.2 Chapter 8 - Problem 29

Prove: $\sum_{i=1}^{K} \sum_{x \in C_i} (x - m_i)(m - m_i) = 0$

Proof:
Following the notation from the proof on page 557, we start from:

$\sum_{i=1}^{K} \sum_{x \in C_i} (x - c_i)(c - c_i)$

$= \sum_{i=1}^{K} \sum_{x \in C_i} (xc - c_i c - xc_i + c_i^2)$
(by mutiplying out terms)

$= \sum_{i=1}^{K} \sum_{x \in C_i} xc - \sum_{i=1}^{K} \sum_{x \in C_i} c_i c - \sum_{i=1}^{K} \sum_{x \in C_i} xc_i + \sum_{i=1}^{K} \sum_{x \in C_i} c_i^2$
(by separating summations by operator)

$= \sum_{i=1}^{K} \sum_{x \in C_i} xc - \sum_{i=1}^{K} m_i c_i c - \sum_{i=1}^{K} \sum_{x \in C_i} xc_i + \sum_{i=1}^{K} m_i c_i^2$
(given m_i = the number of objects in the i^{th} cluster)

$= \sum_{i=1}^{K} m_i c_i c - \sum_{i=1}^{K} m_i c_i c - \sum_{i=1}^{K} m_i c_i^2 + \sum_{i=1}^{K} m_i c_i^2$
(by equation 8.2 in the book, stating $\frac{1}{m_i} \sum_{x \in C_i} x = c_i$ S.T. $c_i m_i = \sum_{x \in C_i} x$)

$= 0$
(by cancelling out the identical positive/negative terms)

$\therefore \sum_{i=1}^{K} \sum_{x \in C_i} (x - m_i)(m - m_i) = 0$ □

2 Clustering

$$NMI = \frac{\sum_{h,l} n_{h,l} log \frac{n n_{h,l}}{n_h n_l}}{\sqrt{\sum_h n_h log \frac{n_h}{n}} \sqrt{\sum_l n_l log \frac{n_l}{n}}} \tag{1}$$

where where h and l are clusters from two different clusterings,
n_h and n_l are the number of data points in the clusters h and l respectively,
$n_{h,l}$ is the number of data points in cluster h as well as cluster l,
and n is the size of the dataset.

In order to find the range of values that NMI can take, we must analyze the extreme cases for NMI which constitute the maximum and minimum values for NMI, namely, when both clusterings produces identical non-trivial clusters and when no information can be gained from the clustering results.

Case 1 (identical non-trivial clustering):
In the case that two separate clusterings give the exact same clustering result, we have $h = l$
$\therefore n_h = n_l = n_{h,l}$ S.T.

$$\frac{\sum_{h,l} n_{h,l} log \frac{n n_{h,l}}{n_h n_l}}{\sqrt{\sum_h n_h log \frac{n_h}{n}} \sqrt{\sum_l n_l log \frac{n_l}{n}}}$$

$$= \frac{\sum_{h,l} n_{h,l} log \frac{n n_{h,l}}{n_{h,l} n_{h,l}}}{\sqrt{\sum_h n_{h,l} log \frac{n_{h,l}}{n}} \sqrt{\sum_l n_{h,l} log \frac{n_{h,l}}{n}}}$$

$$= \frac{\sum_{h,l} n_{h,l} log \frac{n}{n_{h,l}}}{\sum_{h,l} n_{h,l} log \frac{n_{h,l}}{n}}$$

and because the bottom term is obtained from the entropy of both clusterings *
(* see http://dmml.asu.edu/smm/chapters/SMM-ch6.pdf for reference),

we know that this term is the sum of negatives and therefore is:

$$\frac{\sum_{h,l} n_{h,l} log \frac{n}{n_{h,l}}}{-\sum_{h,l} n_{h,l} log \frac{n_{h,l}}{n}}$$

Now, we can see that the the difference between terms in the numerator and denominator is that the log terms are inverse fractions (and therefore will give the the positive or negative inverse of the other). In this manner our numerator and denominator will be the same when multiplied by the same scalar value $n_{h,l}$ and can cancel giving us:

$\frac{1}{-(-1)}$ or $\frac{-1}{-(1)}$, such that NMI = 1. Therefore, our upper bound for NMI is 1.

Case 2 (no information can be gained from the clustering results):

In the trivial case where both clusterings produce just one cluster (which includes all the points), we have our top term:

$\sum_{h,l} n_{h,l} log \frac{n n_{h,l}}{n_h n_l} = \sum_{h,l} n_{h,l} log \frac{nn}{nn} = \sum_{h,l} n_{h,l} log 1$

and given that log1 = 0, this top term zeros out such that our obtained NMI = 0,
therefore our lower bound for NMI is 0.

This is easy to see, as if the entire dataset is clustered into 1 giant cluster, then it is like we did not cluster at all, giving us no new information. Alternatively, think of it as the case where the clusterings are completely independent of one another. This could also be just as easily obtained if no two cluster pairs contained the same points such that $n_{h,l}$ was always zero, however, because we are comparing all clusters from each clustering where both were obtained from the same data, this is not possible.

$\therefore$ NMI = [0,1]. □

3 Community Detection

For this problem, please refer to the attached files included in the homework zip file.

- To run my code locally on your machine, refer to the README.txt file
- For my console output results, see the output.txt file
- For a discussion of the results, see the result_analysis.txt file

a)
There are 34 nodes and 78 edges in the Karate Club network. (see python code for implementation)

b)
Refer to python code implementation for pre-processing steps.

c)
See output.txt and result_analysis.txt for console output from python code and a discussion of the results.