Technical University of Munich
Department of Informatics
Prof. Dr. Stephan Günnemann

Exercise to the lecture
***Mining Massive Datasets* in WS16/17**

Aleksandar Bojchevski (bojchevs@in.tum.de)
http://www.kdd.in.tum.de/mmds

Sheet No. 7

Exercise 1 - Spectral Clustering

You are given the graph on slide 85 (chapter 6: Graphs/Networks). How do the first 3 eigenvectors change when increasing the weight between node $v6$ and $v9$. How does the spectral embedding look like? How does this change affect the final clustering?

Exercise 2: Modularity Consider the definition of modularity slide 94 (chapter 6: Graphs/Networks).

a) Can you reformulate the objective function (maximizing modularity) as a constrained trace minimization problem? Hint: Use a cluster indicator matrix.

b) Given your reformulation in a) propose a relaxation of the constraints to obtain an efficient solution to the problem.

Exercise 3: Probabilistic Models

a) Consider the generalization of slide 100 (chapter 6: Graphs/Networks): A coin flip where we observe n times a 1 and m times a 0. Prove that the maximum likelihood estimate corresponds to $n/(n+m)$.

b) Consider the equation on slide 112 (chapter 6: Graphs/Networks). We want to maximize the probability $p(A|B, z)$ using alternating optimization based on the following algorithm:
step (a): Fix all values of z_i, update B.
step (b): Fix B
  step (b1): fix all z_i except one (e.g. z_j is not fixed), update the value of z_j.
  step (b2): repeat step (b1) for all possible z_j
step (c): repeat from (a) until convergence

i) How to find the optimal parameters for the matrix B in step (a)? Is your approach efficient (complexity)?

ii) How to find the optimal value of z_j in step (b1)?

iii) Is it easy to update all z_i simultaneously? How would you do it or what are potential problems?

Project task 4 - Song Clustering

For this task we are going to perform unsupervised songs clustering via spectral clustering on the Last.fm similarity graph. Your tasks are as follows:

a) Similarly to task 3 parse the json files to extract the graph and form the adjacency matrix W. However, unlike the previous task here we are going to construct a **weighted symmetric** matrix W, which means we will be working with a weighted undirected graph.

To form the matrix W for each pair of songs i and j set $W_{ij} = W_{ji} = max(s_{ij}, s_{ji})$. Note that here while parsing you might find for example pairs where there is only an edge $i \rightarrow j$ (or only $j \rightarrow i$), in that case set the weight according to $W_{ij} = W_{ji} = s_{ij}$ (or similarity s_{ji}).

b) Perform spectral clustering on the graph:
   - You are **not** allowed to use an existing implementation of spectral clustering like for example the one in sklearn.cluster.SpectralClustering
   - Expose a parameter k (default $k = 10$) controlling the number of clusters.
   - Support both the normalized L_{sym} and unnormalized L graph Laplacian

c) Calculate the ratio-cut corresponding to the minimal value according to the the constrained relaxation version of the problem.

d) Calculate the ratio-cut given the hard assignment clustering you obtained in b).

e) Visualize the distribution of tags for each cluster by plotting a histogram of tags per cluster.

Note: Use sparse matrix format for storing W and efficient eigenvalue decomposition for sparse matrices (scipy.sparse.linalg.eigsh).

The deadline for this project task is **07.02.2017 23:55**.