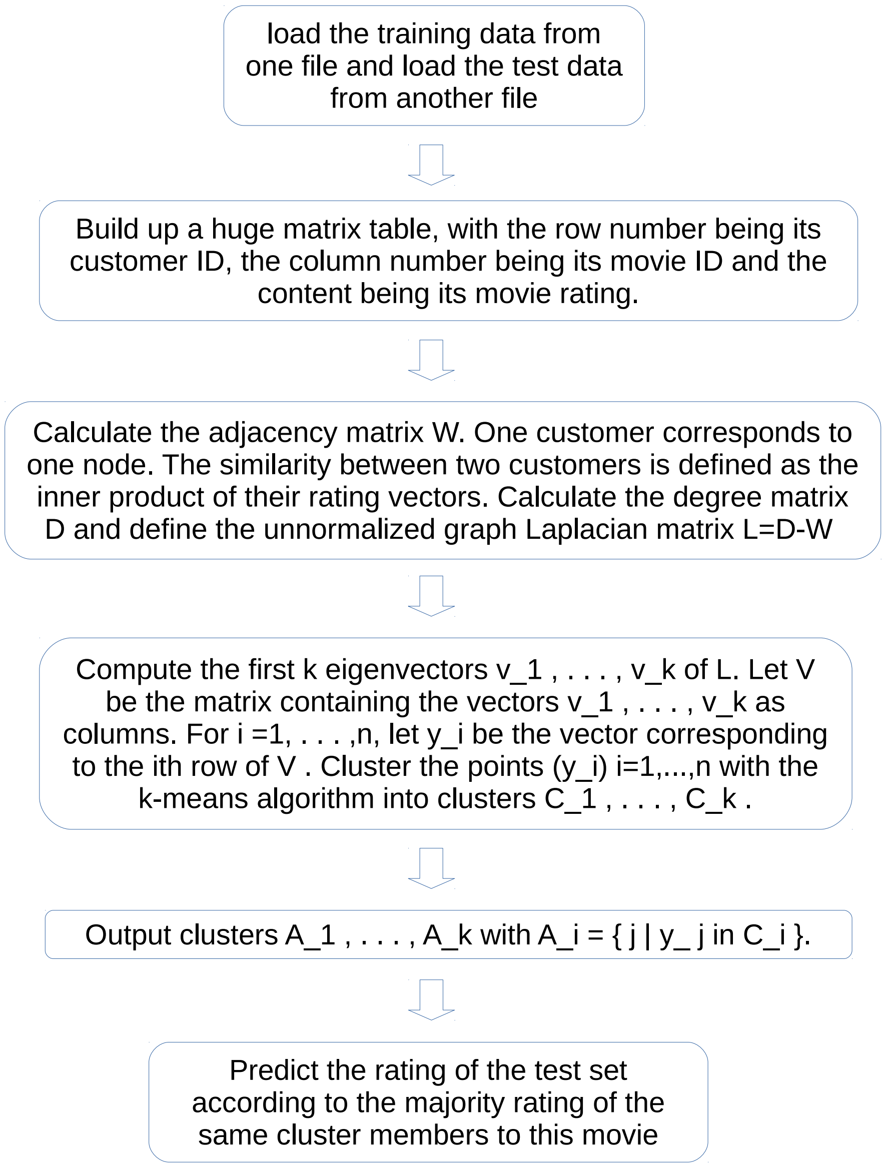
Machine Learning Homework2

Wenjian Hu, Jian Xu

1. A short paragraph over-viewing the reasoning behind your approach:

   We choose to use spectral clustering method to deal with this project.
   Because spectral clustering is simple to implement, can be solved efficiently by standard linear algebra software, and very often outperforms traditional clustering algorithms such as the k-means algorithm.

2. A diagrammatic summary of your approach:

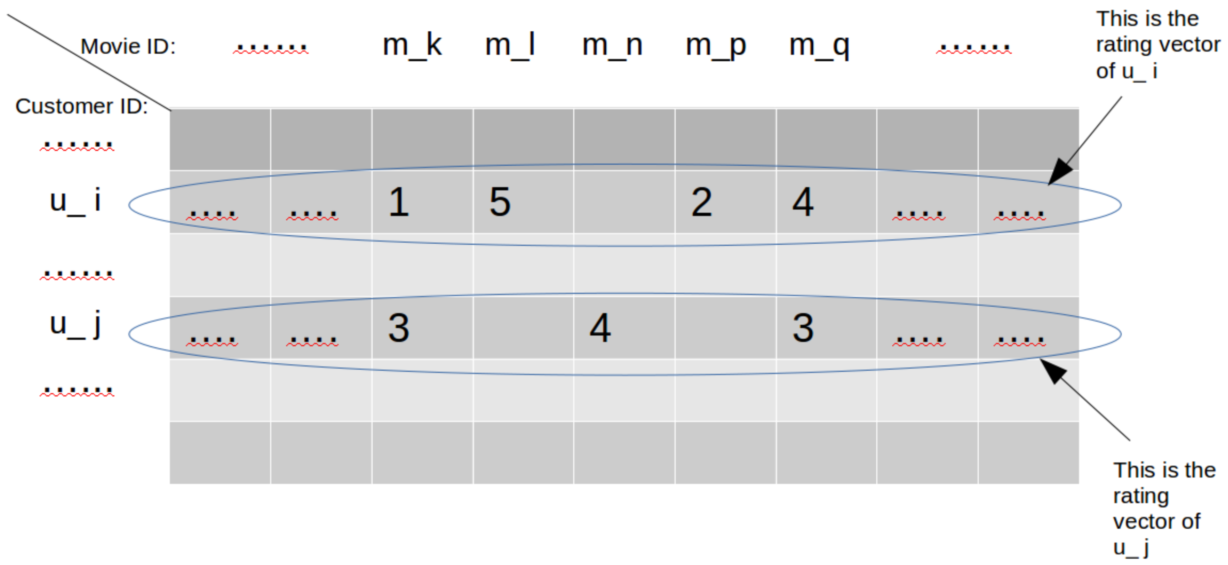

load the training data from one file and load the test data from another file
Build up a huge matrix table, with the row number being its customer ID, the column number being its movie ID and the content being its movie rating.
Calculate the adjacency matrix W. One customer corresponds to one node. The similarity between two customers is defined as the inner product of their rating vectors. Calculate the degree matrix D and define the unnormalized graph Laplacian matrix L=D-W
Compute the first k eigenvectors v_1 , . . . , v_k of L. Let V be the matrix containing the vectors v_1 , . . . , v_k as columns. For i =1, . . . ,n, let y_i be the vector corresponding to the ith row of V . Cluster the points (y_i) i=1,...,n with the k-means algorithm into clusters C_1 , . . . , C_k .
Output clusters A_1 , . . . , A_k with A_i = { j | y_ j in C_i }.
Predict the rating of the test set according to the majority rating of the same cluster members to this movie

3. An at most two page detailed writeup on your approach:

The training data set consists of 1 million transactions/records (user-movie pairs) of features/attributes: <movie id, customer id, rating, date recommended> as your training set. To use the spectral clustering method, we first need to build up the corresponding undirected graph $G = (V, E)$.

Here we define one customer ID as one vertex v_i. Then we can build up the vertex set $V = \{v_1, ..., v_n\}$ by searching all non-repetitive customers in the training data. Each edge between two vertices v_i and v_j carries a non-negative weight $w_{ij} \geq 0$. Then the weighted adjacency matrix of the graph is the matrix $W = (w_{ij})_{i,j=1,...,n}$. If $w_{ij} = 0$ this means that the vertices v_i and v_j are not connected. As G is undirected we require $w_{ij} = w_{ji}$. In our approach, we define the non-negative weight w_{ij} as the inner product of two customer's rating vector:

$$w_{ij} = \frac{\vec{u_i} * \vec{u_j}}{||\vec{u_i}|| * ||\vec{u_j}||}$$


Movie ID:
....
m_k
m_l
m_n
m_p
m_q
....
Customer ID:
This is the rating vector of u_ i
u_ i
1
5
2
4
u_ j
3
4
3
This is the rating vector of u_ j

Then we can define the degree matrix, which is a diagonal matrix with the degrees d_1,..., d_n on the diagonal and the degree of a vertex $v_i \in V$ is defined as:

$$d_i = \sum_{j=1}^{n} w_{ij}$$

After we get the adjacency matrix W and the degree matrix D, we can define the unnormalized graph Laplacian matrix L:

$$L = D - W$$

Then we can compute the first k eigenvectors $v_1, ..., v_k$ of L. Let V be the matrix containing the vectors $v_1, ..., v_k$ as columns. For $i = 1, ..., n$, let y_i be the vector corresponding to the ith row of V . Cluster the points $(y_i)_{i=1,...,n}$ with the k-means algorithm into clusters $C_1, ..., C_k$ and output clusters $A_1, ..., A_k$ with $A_i = \{j|y_j \in C_i\}$.

Cluster A_m with n members
u_{m_1}, u_{m_2}, u_{m_3},....,u_{m_n}
....
Cluster A_p with q members
u_{p_1}, u_{p_2}, u_{p_3},....,u_{p_q}

After we cluster different customers, we can then predict the rating for a particular customer ID and a particular movie ID in the test data set.

To give an accurate prediction, we first need to figure out which cluster the test data customer ID belongs to. Then we can predict the rating of this test data according to the average rating of this particular movie ID from the same cluster members who also watched this movie before.

Until now, we can already use the above approach to make predictions for the test data set. But we still need to discuss some of the issues which come up when actually implementing spectral clustering.

(a) Choosing the number k of clusters is a general problem for all clustering algorithms, and a variety of more or less successful methods have been devised for this problem. One tool which is particularly designed for spectral clustering is the eigengap heuristic, which chooses the number k such that all eigenvalues $1, ..., k$ are very small, but $k + 1$ is relatively large.

(b) In the test set, we might meet some moive IDs which have never been seen by other people before (in the training set) or we might meet some customer IDs which have never appeared before (in the training set). In these cases, it seems impossible to give a prediction based on the training data, so we will simply return an average rating of all movies and all users from the training set.

4. For the test set on the website. A 250,000 line file (named preds.txt) with one prediction/score per line. The prediction can be an integer or real number b/w 1 and 5. I will use a quadratic loss function to score how accurate your approach is:

   The prediction file is already attached, with the file name ”test data predictions”. I also attach my python code, with the file name ”spectral clustering.py”.