CMSC5733 Social Computing

Exercise #3

Deadline: 23:59:59, Oct. 26 (Monday), 2015

Late submission will lead to marks deduction. Days of 1, 2, 3, and 4 or above will cause 10%, 30%, 60% and 100% marks deduction, respectively.

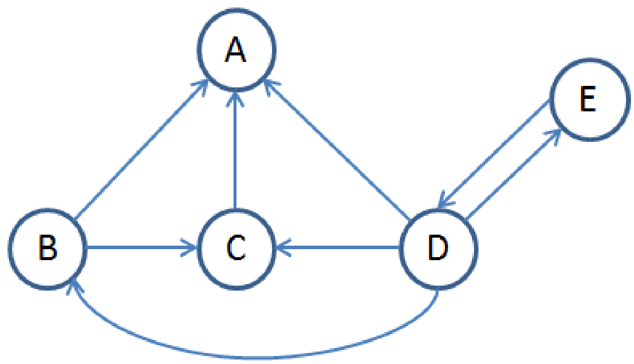
Submission Guidelines: Please send the PDF file to email address cuhk.cmsc5733@gmail.com with your name and student ID.

1. (30pts) Ford-Fulkerson Algorithm.

   Consider the network flow problem with the following edge capacities, c(u, v) for edge (u, v): c(s, 2) = 5, c(s, 4) = 5, c(s, 6) = 2, c(2, 3) = 2, c(2, 5) = 1, c(4, 6) = 2, c(4, 5) = 2, c(6, 5) = 1, c(6, t) = 3, c(5, 6) = 1, c(5, t) = 4, c(3, 5) = 2, c(3, t) = 4.

   (1) Draw the network.
   (2) Run the Ford-Fulkerson algorithm to find the maximum flow. Show each residual graph.
   (3) Show the minimum cut.

2. (30pts)PageRank and HITS.



   The link structure of five web pages is shown in the above figure.

   (1) Suppose d = 0.8, please calculate PageRank score of each state in the first and second iterations. The initiate score of each state is 0.2.
   (2) The initialization of hub score and authority score for each node are both 0.2. Please calculate the hub and authority scores of each state in the first and second iterations.

   Note: please also refer to PageRank in http://en.wikipedia.org/wiki/PageRank or http://beowulf.csail.mit.edu/18.337-2012/MapReduce-book-final.pdf. And please with special attention to dangling nodes. About hits algorithm, please refer to http://en.wikipedia.org/wiki/HITS_algorithm.

3. (40pts) Memory-based Collaborative Filtering.

	I_1	I_2	I_3	I_4	I_5	I_6
U_1	0	2	5	3	1	0
U_2	3	5	4	3	0	2
U_3	4	0	1	4	2	2
U_4	3	0	4	5	5	3
U_5	1	3	5	0	2	2
U_6	3	0	0	0	0	0

The above table shows the ratings of 6 users on 6 items (The value 0 means the user has not rated the item). Please utilize Pearson Correlation Coefficient (PCC) similarity, Cosine similarity and Memory-based CF algorithms introduced in the lecture notes to

(1) find top 2 most similar users of U_3 and estimate U_3's rating on I_2 using PCC similarity and user-based CF.
(2) find top 2 most similar users of U_3 and estimate U_3's rating on I_2 using cosine similarity and user-based CF.
(3) find top 2 most similar items of I_5 and estimate U_2's rating on I_5 using PCC similarity and item-based CF.
(4) find top 2 most similar items of I_5 and estimate U_2's rating on I_5 using cosine similarity and item-based CF.
Note: please also refer to the definition in http://en.wikipedia.org/wiki/Collaborative_filtering, and notice the trick of average calculation. When calculating prediction, please use the following equation in wikipedia.

$$r_{u,i} = \bar{r_u} + k \sum_{u' \in U} simil(u, u')(r_{u',i} - \bar{r_{u'}})$$

4. (extra 20pts) In order to scale the recommender systems, we usually employ SVD-like algorithm to make recommendations. In a system based on SVD-like algorithm, the target is to find two latent matrices to represent users and items through minimizing the following squared error

$$\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij}(R_{ij} - U_i^T V_j)^2 ,$$

where I_{ij} is the indicator of rated item V_j by user U_i, R_{ij} is the rating value (refer to your slides). For the case in question 3, if we want to use SVD-like algorithm to implement our recommender system, which one is better for the following two options? And show the squared errors for the given two options.

1) $U^T =$

```
[[ 1.10551027 -0.42670449  1.55772281]
 [ 1.77669098  0.47945213  0.70527992]
 [ 1.73098233  1.49513475 -0.33416731]
 [ 1.0256562   1.99559686  1.54718149]
 [ 1.00853511  0.0842128   1.68912943]
 [ 1.42324938  1.09823243  0.67127588]]
```

$V^T =$

```
[[ 1.46532483  0.97231578 -0.2860362 ]
 [ 2.03951391  1.51511025  0.45288926]
 [ 1.38862503 -0.37874207  2.15198397]
 [ 1.37980445  1.07252038  0.93106657]
 [ 0.02888709  1.60358726  1.09090268]
 [ 0.75395452  0.59756673  0.65786196]]
```

2) $U^T =$

```
[[-0.20059794  1.90508811]
 [ 1.3302799   1.34492942]
 [ 2.06196556  0.36114035]
 [ 1.94939105  1.59759716]
 [ 0.31149932  1.84784629]
 [ 1.63272368  0.62181092]]
```

$V^T =$

```
[[ 1.76538752  0.14184964]
 [ 2.26467022  1.30820467]
 [ 0.02290252  2.65576854]
 [ 1.39585533  1.44539836]
 [ 1.28545185  0.93067494]
 [ 0.7854192   0.8789373 ]]
```