Student Name ____________________

Write your answers on the pages; use the back of the pages if needed. Clear hand writing is required; if we cannot read what you write, it would be considered incorrect.

Table of Content

<table>
<thead>
<tr>
<th>Problem</th>
<th>Title</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Clustering in Collaborative Filtering</td>
<td>25</td>
</tr>
<tr>
<td>2</td>
<td>Learning to Rank</td>
<td>20 + 10 extra</td>
</tr>
<tr>
<td>3</td>
<td>Link Analysis</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>Federated Search</td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td>Retrieval Models (contents before midterm)</td>
<td>20</td>
</tr>
</tbody>
</table>
1. **Clustering in Collaborative Filtering [25 points]**

Consider the application of bipartite clustering in support of large-scale collaborative filtering (e.g., for book recommendation in Amazon) where user clusters and item clusters can be jointly induced to address data sparse issues. Design and analyze such a method with the following details.

a) Provide the pseudo code for k-means clustering (Subroutine 1 below), including the steps and the formulae for computation, and the analysis of time/space complexities in big-$O$ notation with brief justification based on your pseudo code. [6 points]

b) Provide the pseudo codes for linking users/items to clusters with weights (Subroutine 2, u2cLink), including the steps and formulae for computation, and the analysis of time/space complexities in big-$O$ notation with brief justification based on your pseudo code. [9 points]

c) For the main program Bipartite Clustering, provide the pseudo code (you can use v2cLink in Subroutine 3 without providing the pseudo code). [10 points]

Use the provided notation; if you need additional symbols, please define them. Also, take the efficiency into account in your algorithm design, i.e., when there is an obvious way to save or to waste the computation, choose the better one.

---

**Subroutine 1. kmeans(S, K)**

Input: $S = \{s_i\}$ is a set of vectors, and $K$ is an integer ($K < |S|$).

Output: $p = \{p_1, p_2, \ldots, p_K\}$, the prototypes of $K$ clusters, and the cluster membership $z = (z_1, z_2, \ldots, z_m)$ where $z_i \in \{1, 2, \ldots, K\}$ is the cluster assignment to vector $s_i \in S$.

Algorithm and complexity analysis:

---

**Subroutine 2. u2cLink (X, z, p)**

Input: $X$, the data matrix whose element $X_{ij}$ is the rating by user $i$ on item $j$; $z = (z_1, z_2, \ldots, z_m)$, the cluster assignment to each items with $z_i \in \{1, 2, \ldots, K_v\}$; $p = \{p_1, p_2, \ldots, p_K\}$, the prototypes of the item clusters;

Output: $U = \{u_1, u_2, \ldots, u_a\}$, the set of dimension-reduced user profiles with $u_i \in R^{K_u}$.

Algorithm and complexity analysis:

---

**Subroutine 3. v2cLink (X, z, p)**

Input: $X$, the data matrix whose element $X_{ij}$ is the rating by user $i$ on item $j$; $z = (z_1, z_2, \ldots, z_m)$, the cluster assignment to each user with $z_j \in \{1, 2, \ldots, K_u\}$; $p = \{p_1, p_2, \ldots, p_K\}$, the prototypes of user clusters
Output: \( V = \{v_1, v_2, \cdots, v_m\} \), the set of dimension-reduced item profiles \( v_j \in \mathbb{R}^{K_v} \).

Algorithm and complexity analysis:
[You do not need to provide the pseudo code here; it is similar to u2cLink.]

\[ \cdots \]

\textbf{Main Bipartite Clustering} \( (X, K_u, K_v) \)

Input:
\( X \), the data matrix whose element \( X_{ij} \) is the rating by user \( i \) on item \( j \);  
\( K_u \) is the number of user clusters, and \( K_v \) is the number of item clusters.

Output:
\( z_u = \{z_{u1}, z_{u2}, \cdots, z_{um}\} \) are the cluster assignment to users and \( z_{ui} \in \{1, 2, \cdots, K_u\} \);  
\( z_v = \{z_{v1}, z_{v2}, \cdots, z_{vm}\} \) are the cluster assignment to items and \( z_{vj} \in \{1, 2, \cdots, K_v\} \).

Algorithm and complexity analysis:
2. Learning to Rank for ad-hoc retrieval [20 pt + 10pt extra]

You have learned that the SVM software can be directly used, or extended, for imposing constraints in learning to rank for IR. Provide answers for the following questions

a) Describe (in English) the intuition in the pairwise constraints for training rankSVM. [4pt]

b) Describe (in English) the intuition in the listwise constraints for training SVM-MAP. [4pt]

c) Provide the precise formulas for the pairwise constraints in rankSVM, assuming feature vectors \( x(q_k, d_i) \) is given for every query-document pair. [4pt]

d) Provide the precise formula for the listwise constraints in SVM-MAP, including the formulation for constructing \( \psi(q_k, l) \in \mathbb{R}^m \), the feature vector of every ranked list given a query. [8pt]

e) Prove that the sum of the slack variables in SVM-MAP is the upper bound of the MAP loss on training queries. [Hint: Based on how the ranked list is chosen given a query.] [10pt extra]

Notation:
- \( D = \{d_1, d_2, \ldots, d_N\} \), the training documents
- \( Q = \{q_1, q_2, \ldots, q_K\} \), the training queries
- \( x(q_k, d_i) \in \mathbb{R}^m \), the feature vector representing the pair of \( q_k \in Q \) and \( d_i \in D \)
- \( D^+_k \equiv \{d_i \in D : d_i \text{ is relevant to } q_k\} \) and \( D^-_k \equiv \{d_j \in D : d_j \text{ is irrelevant to } q_k\} \)
- \( L = \{l\} \), the set of all possible ranking of the documents in \( D \)
- \( l^{(k)} \), a correct ranking of documents for query \( q_k \);
- \( l \in L \setminus l^{(k)} \), a ranked list of documents which differs from \( l^{(k)} \);
- \( \psi(q_k, l) \in \mathbb{R}^m \), the feature vector representation of ranked list \( l \) w.r.t. query \( q_k \);
- \( \Delta(l^{(k)}, l) \), the AP loss of ranked list \( l \) in comparison to the correct ranking \( l^{(k)} \);
- \( w \), the vector of model parameters in rankSVM or SVM-MAP;
- \( \xi_j \) and \( \xi_k \), the slack variables in rankSVM and SVM-MAP, respectively.
3. **Link Analysis [20 pt]**

In PageRank, web pages are iteratively scored until convergence using the rule as:

\[
    r^{(k)} := (\alpha M + \beta E)^T r^{(k-1)}
\]

where \( r \in (0,1)^n \), \( \alpha \in (0,1) \), \( \beta \in (0,1) \) and \( \alpha + \beta = 1 \).

**Answer the following questions** (Define additional symbols as needed.)

a) Specify the probabilistic transition matrix \( M \) and the teleportation matrix \( E \) in PageRank. [4 pt]

b) Show how to modify Formula 1 to calculate the Personalized PageRank (PPR) \( \bar{r}^{(k)}_u \) [5 pt]

c) Show how to modify Formula 1 to calculate the Topic Sensitive PageRank (TSPR) vector \( \bar{r}^{(k)}_t \) for topic \( t = 1, 2, \cdots, m \). [5 pt]

d) Show how to use the TSPR vectors to rank pages with respect to a new query, including the formulae of two alternative methods for estimating \( \Pr(t|q) \). [6 pt]
4. Federated Search [15 points]

Suppose that you have been hired by Thomson-Reuters to create a federated search system that integrates material from its many local and national newspaper, television, and magazine sites. This environment is characterized by rapidly changing content and user interests. For example, “boston bombings” was a popular query last month and “ATM hackers” is a popular query today. Some events will be interesting to a national or international audience while others will only be of interest to only to small numbers of people (e.g., CMU commencement).

What type(s) of resource selection algorithm(s) will be most effective at identifying the right sites for queries about breaking events (i.e., the most current news topics)? What types of information will your solution use, and how will it get it? Your solution needs to react quickly to new events.
5. Retrieval Models [20 points]

a) Assume that you have a small document collection consisting of the 5 documents shown below. What scores will the Okapi BM25 algorithm assign to Doc2 and Doc4 for the query “a”? This is a one-word query, so you can treat idf as a constant. You can also assume that there are no user term weights (i.e., that they are set to 1.0). [6 points]

Doc1: a c d b
Doc2: a d b a d a b a
Doc3: c e
Doc4: a b e a
Doc5: d e

b) What scores will the query likelihood model with no smoothing assign to Doc2 and Doc4? [4 points]

c) What scores will the query likelihood model with Jelinek-Mercer smoothing assign to Doc2 and Doc4? Assume $\lambda = 0.5$. [4 points]

d) Discuss the effects of document length in these three retrieval models. Are any of these models biased towards longer or shorter documents? [6 points]