CS 6550: Introduction to Information Retrieval

Assignment 3

UID: u1255635 *Student Name: Hasan Pourmahmoodaghababa*

Question 1.

Both pointwise and pairwise models are learning-to-rank models but with different loss functions.

Pointwise model assumes ranking as a classification problem and considers a loss function for each query-document pair. Then it uses common approaches like regression (to assign a label for a query-document pair) or binary classification techniques (to classify if a query-document pair is relevant or not).

In pairwise model, the loss function is considered for a query and a pair of documents. So, given a query q and two documents d_1, d_2, it determines which one is more relevant to q. Therefore, this model casts the ranking problem to a binary classification problem.

The problem with non-effectiveness of pointwise model is that it considers the ranking problem as a classification problem but ranking is not a classification problem (e.g. the goal in classification is predicting labels and getting best scores but in ranking we should predict sequences, and positions play the key role. Evaluation is also different for classification and ranking.) In fact, classification, and so pointwise model, tries to predict accurate scores not to rank documents correctly. Nevertheless, in pairwise model we evaluate the relevance of two documents each time and opt for the one with more relevance. Thus it is modelling the problem according to relevance (a key factor for ranking) and so it is more effective than the pointwise model that does not care about the relevance.

Question 2.

In retrieval models we studied previously we only have considered text, i.e. signals like word count, document length, position, frequency and corpus length. Thus we have ignored many signals like topic, page quality, user behavior. This means that we have ignored many features and it can affect the performance of models a lot. Therefore, from ranking effectiveness perspective this confirms the need for feature selection in order to capture as many as signals we can.

In terms of system efficiency, feature selection is important as too many features often increases the complexity of model and it means inefficient training. For example, when data is sparse, adding more features can lead to overfitting. So, the performance on test data will be decreased. Therefore, we should select features carefully and discard unuseful features.

Question 3.

One advantage of LSI technique is that it tries to capture certain semantics in a document, that is, it tries to capture the similarity, co-occurrence of terms in a document. However, in TF-IDF model, terms are assumed independent of each other as each word is encoded as a different dimension (it means that two different words are perpendicular to each other and so their cosine similarity is 0, i.e., independent of each other). As another advantage, LSI is more efficient than TF-IDF as it uses a compact representation of data, i.e., the reduced dimensional data.

However, as LSI uses SVD to reduce the dimension (number of latent variables), it loses some information and so precision will be reduced in LSI technique in comparison with TF-IDF. So, it means that TF-IDF is better than LSI in terms of effectiveness. By the way, LSI improves recall.

Question 4.

Question 4.1

User3 and User4 are the most similar users to Bob as I got 0.5196, -0.2887, 0.8769, 0.7947 as similarity value for User1, User2, User3, User4 with Bob, respectively.
For Alice User1 and User3 are the most similar users as I got 0.9391, -0.2236, 0.5283, -0.3078 as similarity value for User1, User2, User3, User4 with Alice, respectively.

Based on user-based nearest neighbor collaborative filtering with Pearson correlation (k = 2) I got the following predicted ratings of movies E and F for Alice and Bob respectively:
Alice: 4.4067, 2.6866
Bob: 2.0779, 1.9303

Question 4.2

Items A and D are most similar ones to item E (similarities I got for A, B, C, D to E are 0.8030, -0.4818, -0.5130, 0.2065 respectively) and items B and C are most similar items to F (similarities I got for A, B, C, D to F are -0.8030, 0.4818, 0.5130, -0.2065, respectively).

Based on item-based nearest neighbor collaborative filtering with Pearson correlation (k = 2) I got the following predicted ratings of movies E and F for Alice and Bob respectively:
Alice: 4.4091, 2.4271
Bob: 1.818, 1.9114

The user-based CF algorithm will recommend movie E to both Alice and Bob. However, the item-based CF algorithm will recommend movie E to Alice but movie F to Bob.

I would say the user-based CF algorithm is better since the number of users in this example is not much bigger than the number of items. Usually the item-based CF algorithm is used when $|\text{users}| \gg |\text{items}|$ in order to deal with inefficiency problem.