Subject: Big Data Analytics

Content-Based Recommendations

i. The two basic architectures for a recommendation system:
   - Content-Based systems focus on properties of items. Similarity of items is determined by measuring the similarity in their properties.
   - Collaborative-Filtering systems focus on the relationship between users and items. Similarity of items is determined by the similarity of the ratings of those items by the users who have rated both items.

A. Item Profiles

- In a content-based system, we must construct for each item a profile, which is a record or collection of records representing important characteristics of that item.
- The profile consists of some characteristics of the item that are easily discovered.
- For example, consider the features of a movie that might be relevant to a recommendation system.

- ✓ The set of actors of the movie. Some viewers prefer movies with their favorite actors.
- ✓ The director. Some viewers have a preference for the work of certain directors.
- ✓ The year in which the movie was made. Some viewers prefer old movies; others watch only the latest releases.
- ✓ The genre or general type of movie. Some viewers like only comedies, others dramas or romances.

B. Discovering Features of Documents

- ➢ There are many kinds of documents for which a recommendation system can be useful.
- ➢ For example, there are many news articles published each day, and we cannot read all of them.
- ➢ A recommendation system can suggest articles on topics a user is interested in, but how can we distinguish among topics? Web pages are also a collection of documents.

- These classes of documents do not tend to have readily available information giving features.
- A substitute that has been useful in practice is the identification of words that characterize the topic of a document.

C. Obtaining Item Features From Tags

- There have been a number of attempts to obtain information about features of items by inviting users to tag the items by entering words or phrases that describe the item.
- Thus, one picture with a lot of red might be tagged "Tiananmen Square," while another is tagged "sunset at Malibu."
- The distinction is not something that could be discovered by existing image-analysis programs.

D. Representing Item Profiles

- Ultimate goal for content-based recommendation is to create both an item profile consisting of feature-value pairs and a user profile summarizing the preferences of the user, based on their row of the utility matrix.

- To generalize this vector approach to all sorts of features. It is easy to do so for features that are sets of discrete values.
- For example, if one feature of movies is the set of actors, then imagine that there is a component for each actor, with 1 if the actor is in the movie, and 0 if not.
- We can have a component for each possible director, and each possible genre.

E. User Profiles

- We not only need to create vectors describing items; we need to create vectors with the same components that describe the user's preferences. We have the utility matrix representing the connection between users and items. Recall the nonblank matrix entries could be just 1's representing user purchases or a similar connection, or they could be arbitrary numbers representing a rating or degree of affection that the the user has for the item.
- With this information, the best estimate we can make regarding which items the user likes is some aggregation of the profiles of those items.

- If the utility matrix has only 1's, then the natural aggregate is the average of the components of the vectors representing the item profiles for the items in which the utility matrix has 1 for that user.

F. Recommending Items to Users Based on Content

- With profile vectors for both users and items, we can estimate the degree to which a user would prefer an item by computing the cosine distance between the user's and item's vectors.
- The random-hyperplane and locality-sensitive-hashing techniques can be used to place (just) item profiles in buckets.
- Given a user to whom we want to recommend some items, we can apply the same two techniques – random hyperplanes and LSH – to determine in which buckets we must look for items that might have a small cosine distance from the user.