Dublin City University

School of Computing

CA4009: Search Technologies

Section 8: Recommender Systems

Gareth Jones

November 2016

Introduction

Recommender systems (RSs) predict items that may be of interest to a user.

A RS can be thought of as similar to an IR system with a standing or long-term persistent information need, where the system seeks assess the likely relevance of items as they are passed to the RS.

In this context, a RS generally needs to make a yes/no decision on each item as to whether it is likely to be of interest to the user.

RSs make their recommendations based on user profiles and previous feedback or “ratings” on previously viewed items from the current user and/or feedback and ratings from other users.

One of the challenges of RS is that reliable recommendations can require large amounts of feedback on previous items.

Introduction

RSs fall into two general categories:

- content-based filtering (CBF)
- collaborative filtering (CF)

CBF methods analyse the contents of a set of items rated by the user, and use the contents of the items, as well as the user's ratings of the items.

This information is used to infer a user profile that can be used to recommend further items that at be of interest to the user.

CF methods use ratings from multiple previous users to make their recommendations.

If users similar to you like an item, you might like it too.

Content-Based Filtering

A CBF analyses the features of previously rated items and uses these to build a profile of user interests.

The recommendation process matches the attributes of the profile against each available item, and passes those which it judges to be of potential interest to the user.

CBF builds a personal profile for the current user, and reasons that a particular recommendation based on features of the profile can be shown to the user.

Content-Based Filtering

New items can be recommended without any user rating since the recommendation process is based on the item contents.

Recommendation in CBF is limited only to features which can be identified in the content of the item and the profile.

The system is only able to make reliable recommendations for a user once they have made sufficient rating of items to build a meaningfully representative user profile.

A key challenge of CBF is determined a threshold over which items will be passed to the user.

Initially, the threshold must be set with little information, over time a better threshold can be determined from larger amounts of feedback.

Collaborative Filtering

CF methods are not based on item content.

CF systems provide recommendations based on ratings of items provided by other users who share common interests to the current user.

This makes CF systems potentially useful for recommendation of any type of item, since they do not depend on content of the item.

The user profile in CF is a set of items and their corresponding rating information.

Ratings information is gathered in two ways:

- *explicitly*: the user provides specific rating information for the item;
- *implicitly*: ratings are inferred from user behaviour with the item.

Collaborative Filtering

Two basic approaches to CF:

- Memory-based algorithms

  Compute the similarity between each existing users and the current user, and select the closest neighbours to the current user.

  Prediction is then based on the rating information of these selected nearest neighbours.

- Model-based algorithms

  Construct a model to represent the behaviour of the users to predict ratings.

Collaborative FIltering

Generally memory-based algorithms are simpler than other recommender algorithms.

However, they are much more sensitive than model-based methods to some common problems of recommender systems.

A sufficiently large number of ratings of items by other users is needed to make reliable recommendations.

Note that the contents of the items does not matter. Recommendations are determined by the relationship between the users and the rated items. Thus CF can be applied to the recommendation of items of any type.

Problems of Recommender Systems

- Sparsity of ratings: In most RSs, each user rates only a small subset of the available items, so most items remain unrated. This can make reliable calculation of similarities between users and items unreliable.
- Cold start problem: two types: *User-side* and *Item-side problems*.

  User-side problem relates to the difficult of making recommendations for new users who have so far provide little rating information.

  Item-side problem where new items have not been rated often enough to make recommendations - generally they will not be recommended very often, and so will not build up rating information.
- “Shilling” - spam attacks - attempts to mislead a RS to recommend specific items.

Hybrid Methods

Hybrid approaches combine CBF and CF, and provide better recommendations than either method in isolation.

They can be implemented in several ways:

- by making CBF and CF predictions separately and then combining them;
- adding CBF capabilities to a CF approach or vice versa;
- unifying the approaches into a single model.

Hybrid Models

Hybrid models can reduce problems such as sparsity and cold start.

For instance, where there is insufficient rating information for effective use of CF, CBF can help by comparing the interests of the current user to each item based on their content.

For the item side cold start problem where new items have not been rated, CBF can again help provide recommendations based on the new item's content.

However, cold start is still a problem if the user has not rated enough items to create a meaningful profile.

Comparison with Information Retrieval

There are similarities between information retrieval systems and recommender systems.

The main feature being that they both seek to provide users with items that they will find relevant or interesting.

IR systems respond to a specific user search request to find items relevant to their current active information need.

RSs build standing profiles of user interests or items of interest to the user, and use these to make recommendations of items that the user may find to be of interest.

Combination of IR and RSs

IR and RSs can be combined to integrate the benefits of both technologies in an interactive search application.

A straightforward integration is to use both systems in isolation and combine their output using a data fusion method.

But how do we make recommendations that will be of interest to the current user with their current information need?

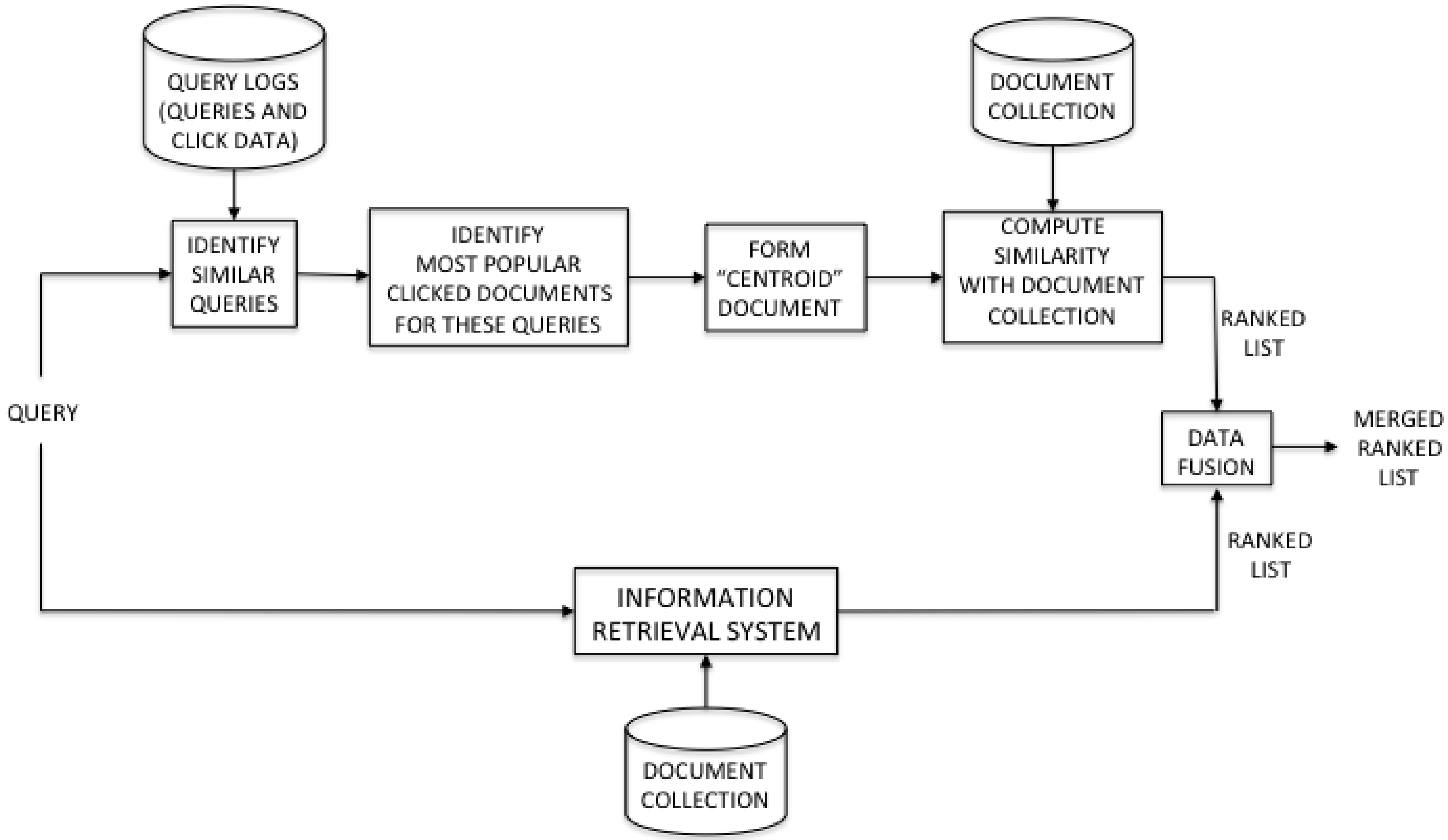
Combination of IR and RSs

Build topical recommenders using previous queries and relevance information.

One possible approach using CBF:

- When a query is entered examine query log of previous user queries and form clusters of similar or identical queries.
- Select the most commonly clicked documents for the queries in the cluster.
- Use term selection methods from query expansion with pseudo relevance feedback to select terms from these documents to form stronger description of the current query (a “centroid’ document”).

Combination of IR and RSs



Combination of IR system with content-based recommender

Combination of IR and RSs

- Compute the similarity between the expanded representation of the query and each available item in the collection.
- Produce ranked list of items as the output of the CF system.
- In parallel, enter the original query into a standard IR system with the same items as the search collection. Produce a ranked list IR output.
- Merge the outputs of the IR system and the CF system by summing the two lists using Data Fusion (see previous notes on Text IR).

Combination of IR and RSs

Combination approach enables rating/relevance information for this query (or similar queries) from multiple previous searchers to contribute to the ranking of items for the current user.

Can be useful to help disambiguate likely interpretation of ambiguous queries. Items relating to the most common interpretations will tend to dominant the ranking.

- In a deployed system, other methods might need to be included to formally ensure diversity of results.