Dublin City University

School of Computing

CA4009: Search Technologies

Section 8: Recommender Systems

Gareth Jones

November 2019

Introduction

Recommender Systems (RSs) are designed to predict items that may be of interest to a user.

RSs can be thought of as related to IR systems from the perspective that they are both intended to identify items which of interest or use to the user.

RSs can make their recommendations based on the profile of an individual user and their feedback or "ratings" on previously viewed items, and/or feedback and ratings from other users.

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

Introduction

RSs are primarily used by merchants as a method to increase transactions and hence their profits.

RSs can seek either:

- to predict the *rating* value (similar to relevance in IR) that would be assigned to an individual item under consideration by the user, or
- to rank the “top-k” items by their predicted rating values.

The ranking version of the problem is generally easier since only the relative values of the predicted ratings are needed.

in the first case determining the absolute predicted value is important to determine whether to recommend the item or not.

Introduction

Interest in RS technologies developed rapidly with the emergence of e-commerce websites in the 1990s.

GroupLens was established for recommendation of Usenet News articles.

This was followed by book and movie recommendation using *BookLens* and *MovieLens*.

Amazon and *Netflix* have been major drivers in the development of RSs.

Note that as well as providing recommendations, Netflix provide explanations for recommendations, e.g. why specific movies are recommended on the basis of previous viewing choices.

There is general increasing interest in methods for providing explanations for the behaviour of proactive information systems.

Introduction

The goals of an operational RS are typically:

- Relevance: recommend items relevant to the user.
- Novelty: recommend new items relevant to the user which they have not seen before.
- Serendipity: recommend *unexpected* or *surprising* items which the users finds relevant.
- Diversity: if a diverse list of items are recommended, there is a greater chance that one of them will be relevant to the user (and purchased!).

Introduction

novelty and *serendipity* are not the same - the key difference being that with novelty items are new but previously unseen, while with serendipity they are new and *unexpected*, i.e. typically not something that the user will be aware of or previously considered.

Serendipitous items can be recommended based on choices of users with similar interest or purchase profiles to the current user.

Serendipity can increase sales diversity, but increases the chance of recommending non-relevant items (and annoying the users!).

It is generally found that possible annoyance at higher proportion of non-relevant items is outweighed by user satisfaction at seeing relevant items of which they would otherwise be unaware.

Collecting Data to Drive Recommendations

Recommendations are generally based on *ratings* of previously access items.

The role of a RS is to make predictions of ratings for new items.

Ratings can either be:

- *explicit*: actively provided by the user as their rating of an item.
- *implicit*: based on the actions of the user - clicking on an item, dwell time on an item, purchasing an item, etc.

This is similar to the explicit and implicit feedback used in relevance feedback in information retrieval described in the section of the notes on Text Retrieval.

Collecting Data to Drive Recommendations

Both Amazon and Netflix use explicit and implicit ratings.

- Explicit - rate items on a scale 1 – 5.
- Implicit - items that were clicked on or purchased.

Google News Personalisation System

- recommend news items based on click history.

Collecting Data to Drive Recommendations

Facebook Friend Recommendation

- not based on selling products
- friend recommendation encourages the growth of the social network
- referred to as *link prediction*
- larger network encourages increased advertising revenue
- recommendations are based on *structural relationships* rather then rating data.

Categories of Recommender System

RSs fall into three general categories:

- content-based recommender systems (CBRS)
- collaborative filtering (CF)
- knowledge-based recommender systems (KBRS)

CBRS 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 may be of interest to the user.

Categories of Recommender System

CF methods use ratings from multiple previous users to make their recommendations, and take no account of contents.

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

KBRSs methods are based on user interactions with the RS to specify their interests.

The user specification is used with domain knowledge to provide recommendations.

Content-Based Recommender Systems

A CBRS combines rating of previous items by the user, their selection or purchase behaviour and analysis of the contents of items to recommend items.

For example, John rates the movie *Terminator* highly, the details of Terminator can be compared to the descriptions of other movies and matching ones recommended to him.

CBRS essentially builds a personal profile for the current user, and reasons that a particular item matching the features of the profile should be recommended to the user.

Content-Based Recommender Systems

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

Recommendation in CBRS 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 ratings of items to build a meaningfully representative user profile.

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

Initially, the threshold must be set with little information, over time a more reliable 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.

Collaborative Filtering

Two basic approaches to CF:

- Memory-based algorithms

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

  Items rated highly by these nearest neighbours can then be recommended to the current user.

- Model-based algorithms

  Use machine learning methods to 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 RSs.

In particular, a sufficiently large number of ratings of items by other users is needed to make reliable recommendations. Model-based methods are more robust to this problem.

Since recommendations are determined by the relationship between the users and the rated items. A user profile may contain multiple item types, e.g. books, household appliances, clothes.

Knowledge-Based Recommender Systems

Knowledge-Based Recommender Systems (KBRSs) are useful for items for purchased infrequently.

- home, car, luxury goods

Sufficient ratings are typically not available for these items to make reliable recommendations to a user using other methods.

Preferences may change over time, e.g. specification of your desired car is likely to change between purchases, so details of previous purchases are unlikely to be helpful.

- for a car: make, model, colour, engine options, etc.

Knowledge-Based Recommender Systems

KBRSs make recommendations based on customer requirements and item descriptions, or use of constraints to specify user requirements.

Knowledge-bases contain data about rules and similarity functions for the matching process.

KBRSs allow the user to *explicity* specific what they want.

- price range, make. model, etc.

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 or sparsely rated.

  This can make reliable calculation of correlations between users and items they are likely to be interested in 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 provided little rating information.

  Item-side problem: where new items have not been rated often enough to make reliable recommendations - generally they will not be recommended very often, and so will not build up rating information.

Hybrid Methods

Hybrid approaches combine CBRS, CF and/or KBRS, and often provide better recommendations than any method in isolation.

They can be implemented in several ways:

- by making CBRS, CF and KBRS predictions separately and then combining them;
- 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, CBRS 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, CBRS 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.

KBRS can be used to specify the types of items that a user is interested in, which could then be used as a filter for recommendations by a CBRS or CF.

Shilling

A significant problem for RSs is “shilling” attacks which attempt to mislead RSs.

Almost anyone can register for and submit reviews to e-commerce websites, such as Amazon, and other interactive websites.

The effectiveness of RSs is dependent on the amount and quality of the data available to it.

Unfortunately, there are motivations for the submission of fake reviews or ratings:

- a manufacturer or author may derive increased sales from the submission of over-positive fake reviews to Amazon.
- a competitor may submit malicious reviews for competitive advantage.

Shilling

A person providing fake information to an RS can be referred to as an *adversary*.

Coordinated fake input to a RS can disrupt the predictions of the RS.

Attacks in this way are referred to as *shilling attacks*.

A single fake review is unlikely to have any impact on the behaviour of the RS (similar to the negligible impact of single fake links in web search, as opposed to an undetected coordinated link farm).

Shilling

Detecting shilling attacks is important in maintaining the integrity of a RS.

There is a trade off between *precision* and *recall* of detecting and removing fake profiles associated with fake reviews and ratings.

- Remove detected fakes too aggressively and real accounts will also be removed, too weakly and some fake accounts will be missed.

Unsupervised: basic idea determine characteristic differences between real and fake profiles, and use these to detect fake accounts.
e.g. identical or very similar profiles may be fakes.

Supervised: machine learning used to train a classifier to detect fake accounts based on examples of real and fake profiles.

Evaluation of Recommender Systems

Evaluation of a RS requires a set of items to be recommended, with ratings of the items by a set of users, in which specific items are rated as relevant to a test user, i.e. how this user would have rated this item if it were recommended to them by an operational system.

As in evaluation of information retrieval, the set of items, the user and their recommendations, should be representative of the environment in which it is planned to operate the RS.

Depending on the task, the evaluation should either:

- Test whether a specific item is correctly recommended to a selected target user.
- Measure the precision and recall of the top-k ranked recommendations.

Comparison with Information Retrieval

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

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

KBRS are more like IR systems in that they are interactive, but the needs are focused on precise user specifications and attributes of defined classes of target items.