

Collaborative Filtering

Introduction

Content filtering:

based solely on matching content of items
to users' information needs

Collaborative Filtering

collect human judgments and match people
who share same information needs and
tastes users share their judgments and
opinions

echoes "word of mouth" principle

Motivations

Advantages over content filtering:

Support for filtering/retrieval of items where contents cannot be easily analysed in an automated manner

Ability to filter based on quality/taste

Recommend items that do not contain content the user was expecting

Applications

The approach has been successful in a number of domains mainly in recommending books/music/films and in e-commerce domains.

www.amazon.com

netflix

lastfm

ebay

ringo/firefly

...

Also been applied to collaborative browsing and searching

In fact, can be applied whenever we have some notion of “ratings” or “likes” or “relevance” of items for a set of users.

Data

Users: a set of user identifiers

Items: a set of item identifiers


Ratings by users of items: numeric values in some predefined range

Can usually view this as a user by item matrix:

Example (ratings from 1 to 5; 0 indicates no rating)

[illegible]

Example (from Resnick et al., 1994)

<i>Message #</i>	<i>Ken</i>	<i>Lee</i>	<i>Meg</i>	<i>Nan</i>
1	1	4	2	2
2	5	2	4	4
3			3	
4	2	5		5
5	4	1		1
6		2	5	

Sample Goals:

Find whether Ken is interested in (likes) message 6

Which users have similar tastes?

Example

[illegible]

Ratings

Explicit:

user usually provides a single numeric value

Universal queries: a gauge set of items is presented to the user for rating.

Choosing this gauge set?

User-selected queries: the user chooses which items to rate (often leaving a sparse ratings matrix with many null values).

Note: User may be unwilling to supply many explicit ratings

Implicit:

User's recommendation is obtained from:
purchase records
web logs
time spent reading an item

This implicit rating is usually mapped into
some numeric scale

For user-user approaches:

Three general steps:

1. Find how similar each user is to every other user (Calculate user correlation)
2. Form groups or neighbourhoods of users who are similar (Select Neighbourhood)
3. In each group, make recommendations based on what other users in the group have rated (Generate Prediction)

Step 1: Find how similar each user is to every other user.

Some Approaches:

- Pearson correlation

- Constrained Pearson correlation

- The Spearman rank correlation

- Vector similarity

Pearson correlation: weighted average of deviations from the neighbours' mean is calculated

$$w_{a,u} = \frac{\sum_{i=1}^m (r_{a,i} - \bar{r}_a) \times (r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i=1}^m (r_{u,i} - \bar{r}_u)^2} \times \sqrt{\sum_{i=1}^m (r_{a,i} - \bar{r}_a)^2}}$$

where for m items:

— $r_{a,i}$ is rating of user a for item i

— \bar{r}_a is the average rating given by user a

$r_{u,i}$ is rating of user u for item i

\bar{r}_u is the average rating given by user u

Vector similarity: uses the cosine measure between the user vectors (where users are represented by a vector of ratings for items in the data set) to calculate correlation.

e.g.

ken = $\langle 1, 5, 0, 2, 4, 0 \rangle$

lee = $\langle 4, 2, 0, 5, 1, 2 \rangle$

meg = $\langle 2, 4, 3, 0, 0, 5 \rangle$

nan = $\langle 2, 4, 0, 5, 1, 0 \rangle$

	<i>K</i>	<i>L</i>	<i>M</i>	<i>N</i>
<i>Message</i>				
1	1	4	2	2
2	5	2	4	4
3			3	
4	2	5		5
5	4	1		1
6		2	5	

Step 2: Form groups or **neighbourhoods** of users who are similar

Some approaches:

Correlation thresholding: where all neighbours with absolute correlations greater than a specified threshold are selected. (say 0.7 if correlations in range 0 to 1)

Best- n correlations: where the best n correlates are chosen.

Size of neighbourhood?

large -> low precision predictions

small -> Few or no predictions

Step 3: For some user (the active user) in a group, make recommendations based on what other users in the group have rated which the active user has not rated.

Approaches:

compute the weighted average of user ratings using the correlations as the weights. This weighted average approach makes an assumption that all users rate items with approximately the same distribution.

compute the weighted mean of all neighbours' ratings. Rather than take the explicit numeric value of a rating, a rating's strength is interpreted as its distance from a neighbour's mean rating. This approach attempts to account for lack of uniformity in ratings.

$$P_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^n (r_{u,i} - \bar{r}_u) \times w_{a,u}}{\sum_{u=1}^n w_{a,u}}$$

where for n neighbours:

\bar{r}_a is the average rating given by active user a

$r_{u,i}$ is rating of user u for item i

$w_{a,u}$ is the similarity between user u and a

Note significant correlations ...

Note that the Pearson Correlation formula does not explicitly take into account the number of co-rated items by users.

Thus it is possible to get a high correlation value based on only 1 co-rated item

Often the Pearson Correlation formula is adjusted to take this into account (Herlocker et al., SIGIR, 1999)

Experimental Approach for Testing:

a known collection of ratings by users over a range of items is decomposed into two disjoint subsets:

The first set (usually the larger) is used to generate recommendations for items corresponding to those in the smaller set.

These recommendations are then compared to the actual ratings in the second subset.

The accuracy and coverage of a system can thus be ascertained.

Metrics

The main metrics used to test the predictions produced are:

coverage: a measure of the ability of the system to provide a recommendation on a given item.

accuracy: a measure of the correctness of the recommendations generated by the system.

Statistical accuracy metrics

- are usually calculated by comparing the ratings generated by the system to user-provided ratings.
- the accuracy is usually presented as the mean absolute error (MAE) between ratings and predictions.

Decision support accuracy metrics

- typically, the value of the rating is not that important-it is more important to know if the rating is a good or a bad rating.

Many other approaches:

Machine learning approaches

Bayesian models

Clustering models

Models of how people rate items

Data mining approaches

Hybrid models which combine collaborative filtering with content filtering.

Graph decomposition approaches

Sample Systems

GroupLens (Konstan et al., 1997) Usenet news; implicit+explicit ratings

Ringo (Shardanand & Maes, 1995) Music recommendations; 1-7 scale; new users presented gauge list (125 artists); constrained Pearson.

PTV (Smyth et al. 1999) Personalised TV guides using collaborative and case-based recommendations using ratings and program descriptions

Netflix, last.fm, spotify etc....

Collaborative Filtering Issues: Sparsity of matrix

In a typical domain, there would be many users and many items but any user would only have rated a small percentage of all items in the dataset.

Using a technique such as **Singular Value Decomposition** (SVD) the data space can be reduced and due to this reduction a correlation may be found between similar users who do not have overlapping ratings in the original matrix of ratings.

Collaborative Filtering Issues: Size of matrix

In general, very large.

Affects computational efficiency.

SVD has been used to improve scalability by dimensionality reduction.

Collaborative Filtering Issues: Noise in matrix

Over time would your ratings change for items?

- how to model time dependencies?

Are all ratings honest, reliable?

Collaborative Filtering Issues:

Size of neighbourhood

Affects predictions but no way to know “right” size

Would Visualisation of neighbourhood help?

Would summarisation of main themes/features of neighbourhood help?

Collaborative Filtering Issues

How to gather ratings?

New users; new items

- perhaps use weighted average of global mean and user's (or item's)

User not similar to others

Combining Content and Collaborative Filtering

For most items rated in a collaborative filtering domain, content information is also available:

e.g.

books: author, genre, plot summary, language, etc.

music: artist, genre, sound samples, etc.

films: director, genre, actors, year, country, etc.

Question: Would it be useful to use this information?

Combining Multiple approaches

Different approaches may suffer from different problems.

Usefulness of aggregating across multiple algorithms?

Recasting as a machine learning problem

Can view CF as a classification problem; for an item do we classify it as relevant to a user or not

Exercise: consider any of the ML techniques you may have covered and identify issues that may arise

Recent trends

Much recent work has focused on not only giving a recommendation but attempting to explain the recommendation to the user.

Questions arise in how best to 'explain' or visualise the recommendation.

Larger evaluation issues present

Conclusions

- Traditionally doesn't use content
- Idea is to use social recommendations
- Is predominately used to recommend items for sale, but has potential to be used in other IR domains.
- Suffers from a few problems which could be perhaps overcome to some extent by combining collaborative and content information.

Summary

- Definitions
- How to gather ratings
- Model-based steps; Pearson Correlation and Weighted Average Recommendations
- Experimental Approach for Testing
- Metrics of coverage and accuracy (and MAE)
- Issues