RECOMMENDER SYSTEMS

Seminario tenuto presso l'IEI da

Pasquale Pagano

Pisa, 2 Marzo 2000

B5-6

Recommender System

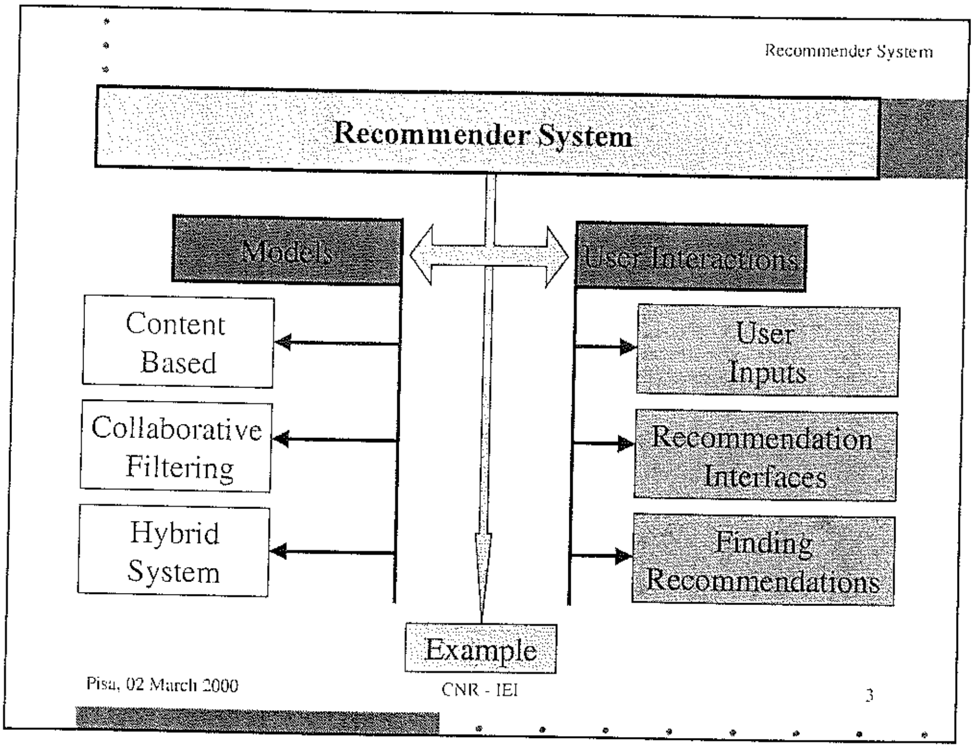
di

Pasquale Pagano

Definition

A recommender system predicts an individual's preferences and makes specific real-time recommendations accordingly.

It does this by learning about each individual's preferences through observing real-time behaviour, such as click-throughs; analyzing the individual profile; recalling past behaviour; and asking the individual to rate a number of relevant items.



Recommender System

Models

Content-based Filtering : the system select the right information for the right people by comparing representations of content contained in the documents to representations of content that the user is interested in.

Collaborative Filtering : the system work by collecting human judgements (known as rating) for items in a given domain; identifies users whose information needs and/or tastes are similar to those of the given user and recommends items they have liked.

Pisa, 02 March 2000 CNR - IEI 4

Models

Hybrid Filtering : the system maintain user profiles based on content analysis, and directly compare these profiles to determine similar users for collaborative recommendation. Users receive items both when they score highly against their own profile, and when they are rated highly by a user with a similar profile.

Content-based Model

This approach has its root in the information retrieval community and employ many of the same techniques.

Passive Environment (static user profile). The users must know in advance how to characterize the information they need before pulling it from the environment.

Active Environment (dynamic user profile). If the user liked a document, weights for the words extracted from it can be added to the weights for the corresponding words in the user profile.

Content-based Model

Shortcoming:

- only a very shallow analysis of certain kinds of content can be supplied.
- when the system can only recommend items scoring highly against a user's profile, the user is restricted to seeing items similar to those already rated (over-specialization). Often this is addressed by injecting a note of randomness.
- a user's own ratings are the only factor influencing future performance, and there seems to be no way to reduce the onerous task without also reducing performance.

Collaborative Model

This approach to recommendation is very different: rather than recommend documents because they are similar to documents a user has liked in the past, the system recommends items other similar users have liked

⇓

- the system does no analysis of the documents, all that is known about an item is a unique identifier
- recommendation for a user are made solely on the basis of similarities to other users

Collaborative Model

Advantages:

- support for filtering documents whose content is not easily analyzed by automated processes (such as movie, music, restaurants, people, politicians….).
- the ability to filter items based on quality and taste.
- the ability to provide serendipitous recommendations - recommending items that are valuable to the user, but do not contain content that the user was expecting.
- the ability to recommend items to a user with dissimilar content to those seen in the past.
- the ability to maintain effective performance given fewer ratings from any individual user.

Collaborative Model

Shortcoming:

- for a user whose tastes are unusual compared to the rest of the population there will not be any other users who are particularly similar, leading to poor recommendations.
- when a new item appears in the database the system cannot recommend this item until it has enough rating.

A collaborative filtering requires a considerable number of user rating (i.e. requires a considerable number of user).

Collaborative Model

Neighborhood-based Method: these systems predict a user's rating based on the similarity between the rating pattern of the user and those of other users

Pearson correlation coefficient: $s_{ij} = \frac{\sum_k (r_{ik} - \bar{r}_i) * (r_{jk} - \bar{r}_j)}{\sqrt{\sum_k (r_{ik} - \bar{r}_i)^2 * (r_{jk} - \bar{r}_j)^2}}$

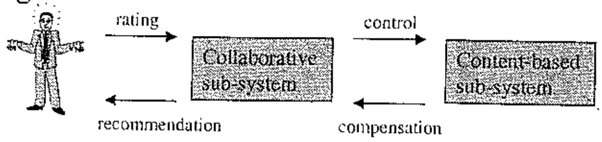
Neighbors of user i : $N(i) = \{j \mid j \in \text{User and } s_{ij} < t\}$

For each item k which is not rated yet by the the user i : $P(i,k) = \frac{\sum_{j \in N(i)} s_{ij} * (r_{jk} - \bar{r}_j)}{\sum_{j \in N(i)} |s_{ij}|}$

Bayesian networks Method; Singular value decomposition with neural net classification; Induction rule learning

Hybrid Model - 1

The content-based and collaborative sub-systems work complementary: the content-based sub-system compensates for the shortage of ratings based on content-based classification, and the collaborative sub-system evaluates and control the compensation based on user rating.



Hybrid Model - 1

Equivalent items: if two items have very similar content, they are very likely to appeal equally to users with similar tastes. Such items are considered "equivalent" and collect in a group.

Virtual users: likes documents with some particular features. A virtual user is embodied as a cluster of items similar to each other and compared with the user in the same way as actual users.

Hybrid Model - 1

The recommendation from the virtual user is merged with those from actual users.

When an item has few ratings, virtual users play a leading role in deciding if it should be recommended.

As the number of ratings on the increases, recommendation from actual users dominate the decision-making, and items which have similar features but have dissimilar tastes will be excluded from the resulted recommendation.

Hybrid Model - 1

Pearson correlation coefficient: $s_{ij} = \frac{\sum_k (r_{ik} - \bar{r}_i) * (r_{jk} - \bar{r}_j)}{\sqrt{\sum_k (r_{ik} - \bar{r}_i)^2 * (r_{jk} - \bar{r}_j)^2}}$

Neighbors of user i : $N(i) = \{j \mid j \in \text{User and } s_{ij} < t\}$

For each item k which is not rated yct by the the user i : $P(i,k) = \frac{\sum_{j \in N(i)} s_{ij} * r_{jk} + \sum_{j \in VN(i)} w_j * \check{S}_{ij}}{\sum_{j \in N(i)} |s_{ij}| + \sum_{j \in VN(i)} |w_j * \check{S}_{ij}|}$

where:

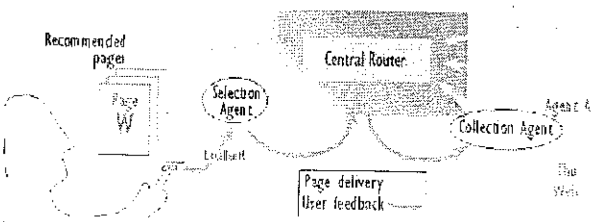
$\check{S}_{ij}$ is the similarity between the user i and the virtual user j

VN(i) is the set of virtual users which are similar to the user i

w_j defines the degree of the effect from the virtual user j on the score compared to an actual user. This weighting factor can be tuned individually for each user according to the value of the collaborative evaluation function.

Hybrid Model - 2

The Collection Stage gathers documents relevant to a small number of topics, computer-generated clusters of interests which track the changing tastes of the user population. These documents are then delivered to a larger number of users via the Selection Stage



Hybrid Model - 2

There are three main components:

- collection agents that find documents for a specific topic
- selection agents that find documents for a specific user
- central router that forwards the documents on to those users whose profiles they match above some threshold.

Every agent maintains a profile based on words contained in documents which have been rated.

Hybrid Model - 2

The user's ratings are used to update their personal selection agent's profile and are also forwarded to the originating collection agents, which will use them to adapt their profiles.

Additionally any highly rated documents are passed directly to the user's nearest neighbors. These collaborative recommendations are processed by the receiving user's selection agent in the same way as the documents from the central router.

Hybrid Model - 2

Unpopular collection agents (whose documents are not seen by many users) or unsuccessful ones (who receive low median scores) are regularly weeded out.

A new collection agent is inserted to represent the best information needs not covered by any collection agents.

⇩

The collection agents specializations need not be fixed in advance, but are determined dynamically and change over time.

The collection agents automatically identify emergent communities of interest

User Inputs

Preference judgements can be explicit statements recorded from the user or implicit measures that are inferred from available data on user activity.

IMPLICIT:

- Purchase Data: which products a user has purchased;
- System Data: data already collected for other purposes - web logs;
- Other Data: time spent reading; URL references.

⇨ The recommendation is generated without explicit effort by the user. (Automatic recommendation)

User Inputs

EXPLICIT:

- Likert: what a user says he thinks of a product, typically on 1-5 or 1-7 scale with high value representing a strong interest in an item and low values representing a strong disinterest.
- Text: Written comments intended for other customers to read. Usually not interpreted by the computer system.
- Editor's Choice: Selections within a category made by human editors, usually employed by the system site, though independent editors are possible in principle.

⇒ The users takes explicit effort to seek out recommendations that will fit his/her interests. (Manual recommendation)

Finding Recommendations

The systems can utilize different methods for allowing users to access the recommendations.

Organic Navigation: recommendations appear as part of the document information page. These recommendations can consist of additional items to consider, average ratings, or a list of other user comments.

Through the course of normal navigation of the site, users are provided with a recommendations.

Request Recommendation List: users can access recommendations based on their previously recorded likes/dislikes. To do so, they simply have to request these recommendations from the system.

Finding Recommendations

Selection Options: users must truly interact with the system in order to receive recommendations. Typically, users choose from a set of predefined criterion/options upon which to base their recommendations. Users of these systems can select from a finite list of keywords, format, length and genre options to define a recommendation criteria.

Keyword/freeform: users provide a set of textual keywords upon which to retrieve future recommendations. This method requires the user to know specially what types of things they are interested in.

Recommendation Interfaces

The method selected may well depend on how the system wants the user to use the recommendation.

Browsing: Recommendations are returned with immediate links to the items being recommended. Recommended browsing helps the E-Commerce site by converting browsers to buyers. It does so by helping the users narrow down their choices and feel more confident in their decision to buy by providing organized access to the recommendations.

Similar Item: Systems attempt to expose customers to items they may have forgotten about, or of which they may have simply been unaware. The items displayed can be entirely selected based on the item(s) in which a user has shown interest.

Recommendation Interfaces

Email: Recommendations can be delivered directly to customers through email. This approach enables site to attract users into their store before other stores with the same product can reach those users.

Text Comments: Systems can provide users with recommendations based directly on the text comments of other users. This not only helps convert browsers into buyers, but should increase loyalty to a site by providing impartial information on the items.

Average Rating: Systems can provide users with recommendations based directly on the rating opinions of other users. By aggregating these ratings into an average rating the system provide users with a check on the quality of an item.This should increase loyalty to a site by providing impartial information on the items.

Recommendation Interfaces

Top-N: Once each site has learned details about a user's like and dislikes, each is able to provide the user with a personalized list of the top number of unrated items for that users. The system can provide a single set of items that might interest a given client without distracting them with items they with items they will not be interested in.

Ordered Search Results: Systems can provide a less restrictive variation of the Top-N approach. While Top-N limits the predictions to some predefined number, ordered search results allow the user to continue to look at items highly likely to be of interest to them.

Example

Amazon

Services	Recommendation Interface	Recommendation Technology	Finding Recommendation
Customer who Bought	Similar Item (books, authors)	Collaborative Filtering (item to item correlation) *Purchase data*	Organic Navigation
Eyes	Email (new item)	Content-Based	Keywords (author, title, subject, ...)
Delivers	Email	Content-Based	Selection options
Book Matcher	Top-N List	Collaborative Filtering (people to people correlation) *Likert*	Request List
Customer Comments	Average Rating Text Comments	Collaborative Filtering (Aggregated Rating) *Likert, Text*	Organic navigation

Example

CDNow

Services	Recommendation Interface	Recommendation Technology	Finding Recommendation
Album Advisor	Similar Item (album, artist) Top-N List	Collaborative Filtering (item to item correlation) *Purchase data*	Organic Navigation Keywords
My CDNow	Top-N List	Collaborative Filtering (people to people correlation) *Likert (own it and like it ; own it and dislike it)*	Organic Navigation Request List

Example

eBay

Services	Recommendation Interface	Recommendation Technology	Finding Recommendation
Feedback Profile	Average Rating Text Comments (buyers and sellers)	Collaborative Filtering (Aggregated Rating) *Likert, Text*	Organic Navigation

Levis

Services	Recommendation Interface	Recommendation Technology	Finding Recommendation
Style Finder	Top-N List	Collaborative Filtering (people to people correlation) *Likert*	Request List

Example

Moviefinder

Services	Recommendation Interface	Recommendation Technology	Finding Recommendation
Match Maker	Similar Item (mood, theme, genre or cast)	Collaborative Filtering (item to item correlation) *Editors choice*	Navigate to an item
We Predict	Top-N List OrderedSearchRes. Average Rating	Collaborative Filtering (people to people correlation) *Aggregated Rating* *Likert*	Keywords Selection options Organic Navigation

Example

Reel

Services	Recommendation Interface	Recommendation Technology	Finding Recommendation
Movie	Similar Item (movie)	Collaborative Filtering (item to item correlation) *Editor's choice*	Organic Navigation
Movie Map	Ordered Search Result	Content-Based *Editor's choice*	Keywords

Conclusion

- A Recommender System allows the customers to satisfy the own information needs in a reasonable amount of time.
- A Recommender System create customizable products and services.
- The Recommender Techniques are part of personalization on a site, because they help the site adapt itself to each user.
- Recommender Systems automate personalization on the Web, enabling individual personalization for each user.

References

- An Algorithmic Framework for Performing Collaborative Filtering - Herlocker, Konstan, Borchers, Riedl - 1999
- Recommender Systems in E-Commerce - Schafer, Konstan, Riedl - 1999
- Recommendation Engine - Net Perceptions - www.netperceptions.com
- Content-Based, Collaborative Recommendation - Balabanovic, Shoham - 1997
- Social and Content-Based Information Filtering for a Web Graphics Recommender System
- Active Recommendation Project - Rocha, Joslyn, Kantor - www.c3.lanl.gov
- TalkMine and the Adaptive Recommendation Project - Rocha - 1999

References

- Recommender Systems for Evaluating Computer Messages - Avery, Zeckhauser - 1997
- Recommender Systems - Resnick, Varian - 1997
- A System for Sharing Recommendations - Terveen, Hill, Amento, McDonald, Creter - 1997
- Collaborative value filtering on the Web - Authors omitted
- Combining Collaborative Filtering with Personal Agents for Better Recommendations - Good, Shafer, Konstan, Borchers, Sarwar, Herlocker, Riedl - 1999
- Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System - Sarwar, Konstan, Borchers, Herlocker, Miller, Riedl - 1998

References

- CiteSeer: An Autonomous Web Agent for Automatic Retrieval and Identification of Interesting Publications - Bollacker, Lawrence, Giles - 1998
- A System for Automatic Personalized Tracking of Scientific Literature on the Web - Bollacker, Lawrence, Giles - 1999
- A Supra-Classifier Architecture for Scalable Knowledge Reuse - Bollacker, Ghosh - 1998
- Text categorization Through Probabilistic Learning: Applications to Recommender Systems - Bennett - 1998
- Combining Social Networks and Collaborative Filtering - Kautz, Selman, Shah - 1997
- Applying Collaborative Filtering to Usenet News - Konstan, Miller, Maltz, Herlocker, Gordon, Riedl - 1997

References

- Experiences with GroupLens: Making Usenet useful again - Miller, Riedl, Konstan - 1996
- A System for Automatic Personalized Tracking of Scientific Literature on the Web - Bollacker, Lawrence, Giles - 1999
- User Preferences When Searching individual and Integrated Full-Text Databases - Park - 1999
- The Effects of Singular Value Decomposition on Collaborative Filtering - Pryor - 1998
- Distributing Information for Collaborative Filtering on Usenet Net News - Maltz - 1994
- Sluice: A Java-Based Framework for Collaborative Interactive Modular Visualization Environments - Isenhour, Shaffer, Begole, Nielsen, Abrams - 1998

References

- Learning To Surf: Multi-agent Systems For Adaptive Web Page Recommendation - Balabanovic - 1998