First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

Examine the patterns that emerge when considering the taxonomy and identify models of recommender applications in E-commerce

Ionela Maniu, George Maniu

Abstract

Recommender systems are a technology that can help businesses, such systems allow search engines and advertising companies to suggest advertisements or offers to display based on consumer behavior. Product-associated recommendations allow businesses to respond to each customer's current interests and allow the natural associations among different products to guide customers to the right purchase. In this paper, we examine and synthesize data underlying patterns of recommender applications.

1 Introduction

Technology has dramatically reduced the barriers of publishing and distributing information. Now is time of technologies that can help us shift through all the available information to find that which is most valuable to us. In everyday life, people rely on recommendations from other people by spoken words, reference letters, news reports from news media, travel guides, etc. Recommender systems assist and augment this natural social process to help people sift through available books, articles, webpages, movies, music, restaurants, jokes, grocery products, etc. to find the most interesting and valuable information for them [10]. For e-commerce, conventionally, a recommendation system is considered as a success if clients purchase the recommended products. Personalized service requires firms to understand customers and offer goods or services that meet their needs**. Successful firms are those that provide the right products to the right customers at the right time and for the right price.** Recommendation systems apply data mining techniques to determine the similarity among thousands or even millions of data. One such techniques is collaborative filtering(CF). There are three major processes in the recommendation systems: object data collections and representations, similarity decisions, and recommendation computations. Collaborative filtering aims at finding the relationships among the new individual and the existing data in order to further determine the similarity and provide recommendations [13].

2 Examples of recommendation systems

Collaborative filtering has been very successful in both information filtering application and E-commerce applications.

Commercial sites that implement collaborative filtering systems include:

- *Amazon.com* is a commercial book shop/recommendation service.

- *eBay* is a commercial system which allows buyers and sellers to contribute to profiles of other customers with whom they have done business.
- *Rind* is a recommender system to help with buying a PC.
- *Ski-europe.com* recommends ski holidays.

Non-commercial sites that implement collaborative filtering systems include:

- *SurfLen* monitors user browsing and recommends web pages.
- *Tapestry* is an email recommendation system.
- *GroupLens* collaboratively recommends Usenet newsgroup articles.

3 Examine data patterns

3.1 Data Representation

User representation:

- by user attributes : demographic data such as gender, birth date, salary
- by associated items: the products the user has expressed interest in, has given ratings to or actually purchased
- by transactions: attributes extracted from the user's transaction history such as time, frequency, and amount, can partially represent a user's behavior pattern
- by items or item attributes associated with the user reflected in the feedback data- e.g., a user may be characterized as liking romantic stories and favoring low prices based on the attributes of the books she has purchased or simply as a set of books she has purchased

Items representation:

- by item attributes - such as price, content, brand, etc.
- by associated users – the customers who have purchased this item before

There are recommendation algorithms that directly operate on the *user-item interaction matrix* and do not explicitly derive any intermediate user or item representations

Transactions representation :

- by the transaction attributes: such as time, amount, etc.
- by items in the transactions.

Researchers have also included some transaction attributes such as time and place as additional dimensions and support a different type of recommendation that may be based on different combinations of dimensions, such as the recommendation of web content to a particular customer on weekends, or the recommendation of the best time to promote certain products to a particular customer. However, most existing recommender systems focus on the analysis of the two dimensions of users and items. More information about data representation can be found at the adress [14].

Other approach for data representation consist of exploiting product classification taxonomies. Semantic product classification corpora for diverse fields are becoming increasingly popular, facilitating smooth interaction across company boundaries.

Several online retailers employed *product taxonomy* (PT) to give a clear view of their product lines in tree structure. PT is practically represented as a tree and categorizes a set of products at a low level into a more general product at a higher level. The leaves of the tree denote the product instances, and non-leaf nodes denote product classes obtained by combining several nodes at a lower level into one parent node . The root node labeled by 'All' denotes the most general product category

For a given category, attributes are then determined in 3 steps. First, create basic list of attributes, this list is modified through asking users to express their opinions about importance of items. Finally, it's generate final list according to preliminary list and users opinions. There are also determine levels and possible values for final attributes.

Once attributes of categories are determined, we should place this attributes in product taxonomy. There by child nodes will inherit parent node attributes while it has own attributes. Examples and other information can be found in [5] and [16].

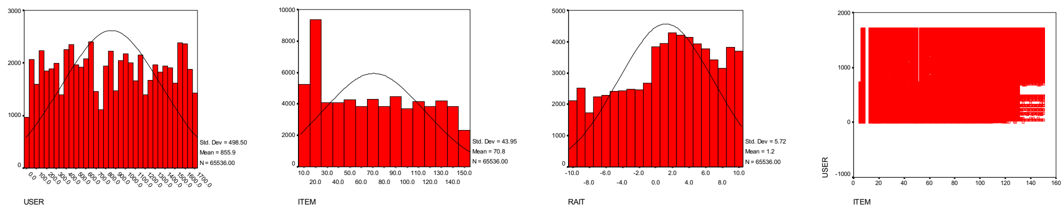
3.2 Data distribution

- **many items.** If there are only a few items to choose from, the user can learn about them all without need for computer support.
- **many ratings per item.** If there are only a few ratings per item, there may not be enough information to provide useful predictions or recommendations.
- **more users rating than items to be recommended.** If there are few ratings per user, you'll need many users. Lots of systems are like this. The ratings distribution is almost always very skewed: a few items get most of the ratings, a long tail of items that get few ratings. Items in this long tail will not be confidently predictable.
- **users rate multiple items.** If a user rates only a single item, this provides some information for summary statistics, but no information for relating the items to each other.
- **homogenous items.** Music albums are like this. Most are similarly priced, similar to buy, of a similar length. Books or research papers are also like this. Items sold at a department store are not like this: some are cheap, some very expensive. For example, if you buy a hammer, perhaps you should not be recommended a refrigerator.

3.3 Experimental study

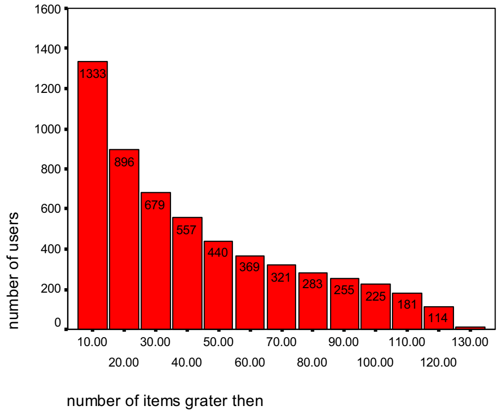
For examine this data pattern (data distribution) we used Jester data set example**.** The dataset comes from Ken Goldberg's joke recommendation website, Jester [11]. In Jester, users rate a core set of jokes, and then receive recommendations about others that they should like. The database we used has 150 jokes, and records of 1708 users. Some users end up reading and rating all the jokes. Jester has a rating scale from -10 to 10, the matrix user-item has 65536 raitings. Ratings are implemented with a slider, so Jester's scale is continuous.

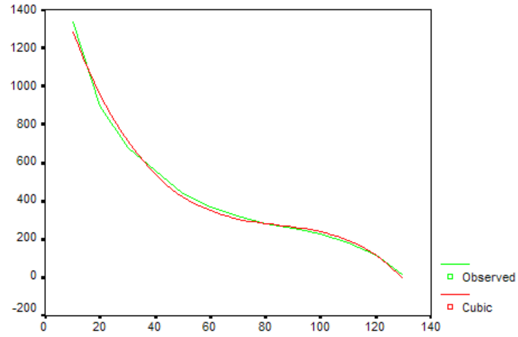
In order to establish data distribution (user, item, raitings) we achieved the histogram chart below and display normal curve and scatter chart :

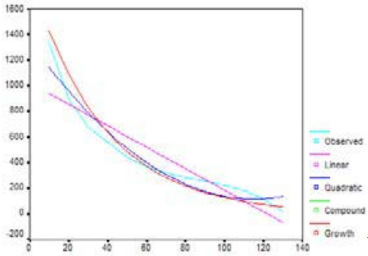


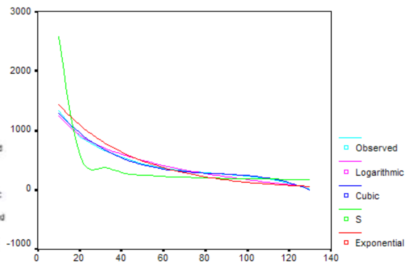
As we can see Jester is a dense dataset (more dense than other datasets) which is why we should try to normalize the data (for example raiting matrix) and after that clustering the data (using clustering algoritms).

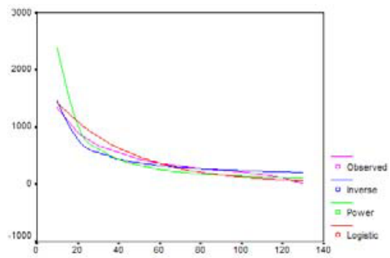
Chart below highlights the fact that the number of users decreases if the number of raiting items increase(with a 10 step), this phenomenon following an cubic curve, result that was obtained after comparison with several types of curves.



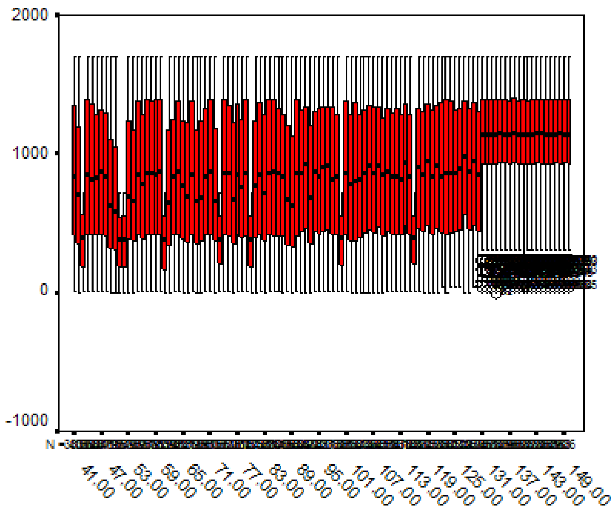






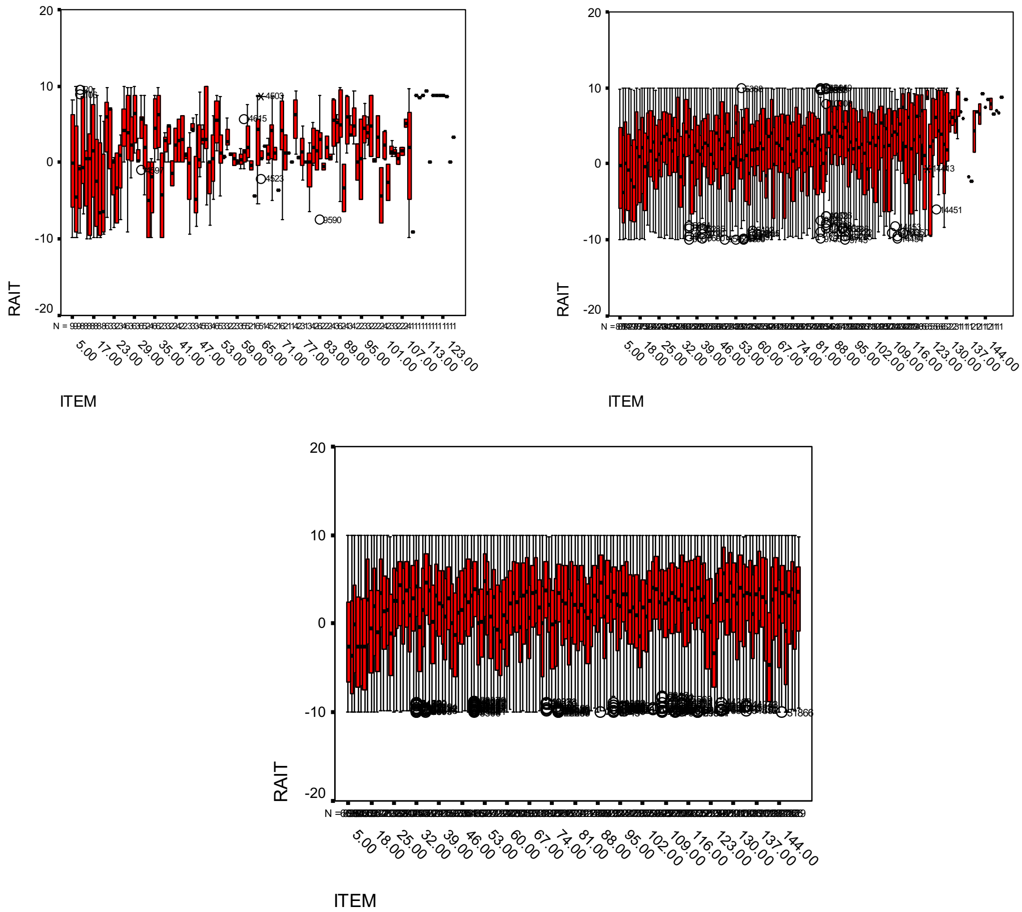


We also can see that those items that are on the last position in the items list are those with the fewest raitings (jokes which the user has not rated from the known ratings of jokes from all users). Consequently this items that have been rated by only few people would be recomanded very rarely even if those few users gave high ratings to them. Also for the 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.



Organizing data into clusters can be a solution in this situation too. In [12], the author proposed a Bregman co-clustering algorithm whose main objective is to find a partition of m rows and n columns of a data matrix into k row clusters and l column clusters such that the distance between the original matrix and the reconstructed matrix is minimized. Another solution is to use user/item profile information when calculating user/item similarity.

Regarding the fact that it is important to be **more users rating than items to be recommended,** it can be seen in the graphs that follow, that increasing the number of users has as consequence the average raitings homogenization and thus increase the accuracy of the recommendation. There were taken into account 3 situations: less than 10 users, less than 100 users, less than 1,000 users.



4 Conclusion

Recommender systems’ performance can be easily affected when there are no sufficient item preferences data provided by previous users. This paper suggests, after examine a part of the patterns of data, another information source, item taxonomies, in addition to item preferences for assisting recommendation making. Item taxonomic information has been popularly applied in diverse ecommerce domains for product or content classification, and therefore can be easily obtained and adapted by recommender systems.

References

[1] Basu, C., Hirsh H., and Cohen, W. , *Recommendation as classification: Using social and content-based information in recommendation*, Workshop on Recommender Systems. AAAI Press, Reston, 11-15, 1998

[2] TSO, H. L. K. and Schmidt-Thieme, *Attribute-Aware Collaborative Filtering*, Annual Conference of the German Classification Society, Magdeburg, Germany, 2005

[3] Ziegler, C., Schmidt-Thieme, L. and Lausen, G., *Exploiting Semantic Product Descriptions for Recommender Systems*, 2nd ACM SIGIR Semantic Web and Information Retrieval Workshop , UK, 2004

[4] http://en.wikipedia.org/wiki/Collaborative_filtering

[5] http://www.amazon.com

[6] L.P. Hung, *A personalized recommendation system based on product taxonomy for one-to-one marketing online*, Expert Systems with Applications 29, pp. 383–392, 2005

[7] S.H. Choi, S. Kang and J.Y. Jeon, *Personalized recommendation system based on product specification values*, Expert Systems with Applications 31 , pp. 607–616, 2006

[8] www.movielens.org

[9] www.grouplens.org
[10] http://www.hindawi.com/journals/aai/2009/421425.html
[11] K. Goldberg, D. Gupta, M. Digiovanni, and H. Narita. Jester 2.0 : *Evaluation of a new linear time collaborative filtering algorithm*. In ACM SIGIR, August 1999. Poster Session and Demonstration.
[12] Arindam Banerjee, Inderjit S. Dhillon, Joydeep Ghosh, Srujana Merugu, and Dharmendra S. Modha. *A generalized maximum entropy approach to bregman co-clustering and matrix approximation*. Journal of Machine Learning Research, 8:1919–1986, August 2007.
[13] http://sir-lab.usc.edu/publications/121new.pdf
[14] http://www.personal.psu.edu/~huz2/Zan/docs/zanhuang.dissertation.pdf
[15] http://www.cs.uni.edu/~schafer/publications/CF_AdaptiveWeb_2006.pdf
[16] http://deposit.ddb.de/cgibin/dokserv?idn=975319213&dok_var=d1&dok_ext=pdf&filename=975319213.pdf
[17] http://www.cs.uni.edu/~schafer/publications/CF_AdaptiveWeb_2006.pdf

Maniu Ionela
Univ. Lucian Blaga Sibiu
Informatics Departament
Dr. Ratiu No. 5-7, Sibiu
Romania
E-mail: mocanionela@yahoo.com

Maniu George
Univ. Spiru Haret Brasov
Management Faculty
Turnului No.7
Romania
E-mail: costelmaniu@yahoo.com