Using Wikipedia to Boost Collaborative Filtering Techniques

Gilad Katz[1,2] katzgila@bgu.ac.il

Nir Ofek[1,2] nirofek@bgu.ac.il

Bracha Shapira[1,2] bshapira@bgu.ac.il

Lior Rokach[1,2] liorrk@bgu.ac.il

Guy Shani[1] shanigu@bgu.ac.il

[1] Department of Information Systems Engineering, Ben Gurion University, Israel

[2] Deutsche Telekom laboratories at Ben Gurion University, Israel

ABSTRACT
One important challenge in the field of recommender systems is the sparsity of available data. This problem limits the ability of recommender systems to provide accurate predictions of user ratings. We overcome this problem by using the publicly available user generated information contained in Wikipedia. We identify similarities between items by mapping them to Wikipedia pages and finding similarities in the text and commonalities in the links and categories of each page. These similarities can be used in the recommendation process and improve ranking predictions. We find that this method is most effective in cases where ratings are extremely sparse or nonexistent. Preliminary experimental results on the MovieLens dataset are encouraging.

Categories and Subject Descriptors
H.3.3 [**Information Systems**]: Information Search and Retrieval

General Terms
Algorithms, Experimentation

Keywords
Recommender Systems, Wikipedia, Collaborative Filtering, Cold Start Problem

1. INTRODUCTION
A major task of recommender systems is the prediction of item ratings. For example, the online video rental service NetFlix displays by each newly released movie a predicted rating for the movie, helping the customer decide whether to rent that movie.

Perhaps the most common approach for recommender systems is collaborative filtering (CF) [9]. CF methods predict users' interest in specific items based on their past ratings and the ratings of other (similar) users. This information is used to calculate the similarity among items or users.

RecSys'11, October 23–27, 2011, Chicago, Illinois, USA.

Two important drawbacks of CF methods are data sparsity and the cold-start problem;The user-rating matrix is in many cases sparse and it is difficult to find items that were jointly rated by many users because of the diversity of the item set and user tastes; The cold start problem refers to new items, where ratings are difficult to predict before sufficient data has been collected [1-2].

In this paper, we boost the available data by integrating data from Wikipedia - a free encyclopedia built collaboratively. We utilize several different relations between Wikipedia items in order to provide traditional collaborative filtering techniques with additional information. We use three different Wikipedia resources; the page text, its categories, and links between Wikipedia articles.

Our method is a hybrid of CF and content-based recommendation [3], where the content is a result of Wikipedia users collaboration. We use the Wikipedia content for computing the similarity between items when the existing data on the rating matrix is too sparse. We explain how these similarities can be employed in an item-item CF framework.

We evaluate the performance of the proposed method over the MovieLens dataset, simulating various sparsity levels. We compare the performance using separately the three data sources - text, links and categories - and their combinations. We show that when the sparistiy level is high, integrating Wikipedia content significantly improves recommendation results.

2. Background
The idea of integrating external sources to boost collaborative filtering was already explored in previous studies, differing on the data sources they use and on the method used to integrate these sources.

The external sources used are extremely versatile. For example, [4] uses organizational social networks in order to compute similarities between users. Possible indicators of groups include attending the same events, co-authored paper, and being members of the same projects. This correlation is used to create profiles which are then integrated with a web based recommender system.

Similar to our approach, other works use content information gathered on items in order to calculate their correlations. For example, [5] integrates information from a set of knowledge sources, including Wikipedia, and generates a graph of linguistic terms. The generated model enables identifying similar items for any given item. However it only uses certain terms and does not use all the user-generated text. In addition, our work investigates the utilization of other Wikipedia attributes: categories and links.

Semantic relations extraction from concepts in Wikipedia was used by [6] in order to represent domain resources as a set of interconnected Wikipedia articles. This was achieved by using the hyper-links in Wikipedia pages, which were represented in an item similarity matrix. The authors used only the links of the Wikipedia pages and ignored its text and categories.

Exploiting Wikipedia links and categories was also used in the work presented in [7] on modeling interconnected network of concepts. A concepts network was used to enrich the semantics of the given user's preference. This work differs from ours as it uses only the hyperlinks of Wikipedia pages as well as probabilistic models to generate user-user similarity.

Unstructured textual information from IMDB was used in [3], where it was integrated with EachMovie's user-rating dataset. A Bayesian algorithm was then used to classify ratings into one of the six class labels 0-5 (representing a user's satisfaction with an item). The classification result, based on bag-of-words approach, was used to boost pure collaborative filtering technique. This work differs from ours in the fact that it only utilizes text and ignores other types of data. Furthermore, we use different methods to calculate the item similarity.

3. Wikipedia-based Item Similarity

The proposed method consists of two offline preprocessing steps: identifying Wikipedia item data pages and the enrichment of the user-item ranking matrix with artificial ratings using Wikipedia content. We then use an online rating prediction.

3.1 Assigning Wikipedia pages to data items

In order to use the information contained in Wikipedia, we first need to identify the pages describing the items. We currently focus on movies ratings, but our method can be generalized to other domains that are described on Wikipedia, such as books, music and so forth. This is a challenge, because of possible ambiguities – some movies are named after books, while others have the names of objects ("Boomerang" being one example), and some have adjective names (e.g. "Big").

This was done by the following heuristic steps:

a) Generating several variations of each movie's name (with and without the year, the removal of ", the" and ", a" etc).

b) Compare the generated names with corresponding page titles in Wikipedia.

c) Choose the page with the largest number of categories that contain the word "film" (e.g., "Films shot in Vancouver").

Using this technique, we were able to assign 1512 items out of the 1682 (89.8%) contained in the MovieLens database to their corresponding Wikipedia pages. The remaining 170 movies were assigned no pages and therefore their rating did not benefit from the proposed method.

3.2 Predicting Item Ratings

We now describe how we compute item rating predictions. This process consists of the following steps; First, we use several data sources to calculate item-item similarity using Wikipedia; Then, we create a similarity metric that combines the existing user ratings and the data collected from Wikipedia; Finally, we use the combined item-item similarity metric to compute ratings for unknown items and insert them to the user-item rating matrix in order to reduce sparsity. We now review each step in detail.

a) Similarity computation:

We use three Wikipedia features to calculate the similarity between a pair of items – text, categories and links.

Text similarity: we extract the text from each movie page, and use a bag-of-words approach (combined with stemming and stop-words removal) to represent the text [8]. We then compare similarity between movies based on a cosine measure [9] over the bag-of-words. Thus, movies that are described using many identical words are considered similar.

C*ategory similarity:* in Wikipedia, items can be assigned a set of categories (or tags). In the movie domain these categories can be "American films", "Actions films", or "Films shot in Los Angeles". We compute movie similarity based on the number of joined categories of two movies.

Links similarity: in Wikipedia, many pages contain links to other Wikipedia pages, forming a graph of linked pages. In this work we use only single indirection of links. That is, we count identical outgoing links from two movie pages. For example, Batman and Con Air (both action movies) have high links similarity.

It is often noticed in Wikipedia studies that such links may contain much noise. We thus consider only the links that are in the "plot" and "cast" paragraphs. Note that the Wikipedia categories are also implemented as links to category pages. We do not consider categories as links.

We compute each of the three similarity measures to produce three item-item similarity matrices. When combining these different matrices, a calibration is required in range and scale. We use here the following (ad-hoc) similarity calibration method:

The text similarity computed using the cosine score over the bag-of-words produces values in the range [0,1]. The values of the two other similarities are natural numbers (counts). We choose to truncate counts higher than 5, which are very rare, as these already signify high similarity for both links and categories. Thus, we remain with values in the range of [1,5]. We then use the following transformation on the textual similarity values (the values were set empirically):

$$f(x) = \begin{cases} x >= 0.9 \Rightarrow 5 \\ 0.9 > x >= 0.7 \Rightarrow 4 \\ 0.7 > x >= 0.5 \Rightarrow 3 \\ 0.5 > x >= 0.3 \Rightarrow 2 \\ x < 0.3 \Rightarrow 0 \end{cases}$$

An obvious next step for future research is to learn the calibration using machine learning techniques, but for now this simple ad-hoc method serves us well.

b) Combine the metrics into a unified item-item similarity metric

At this point there are three item-item similarity metrics. Our goal is to combine them into a single item-item similarity metric that can be used to compute ratings predictions. We currently combine the metrics using a simple weighted average approach. For every pair of items *i* and *j* we use the following formula:

$$final\_sim(i,j) = \frac{\sum_{m=1}^{n} sim_m(i,j) * weight_m}{\sum_{m=1}^{n} weight_m}$$

where i and j are the items whose similarity we attempt to calculate and m is a similarity metric. At this point, all similarity matrices were given identical weight (assigning the weights smartly was also reserved for future research).

c) Compute ratings for unknown items

We now attempt to use the item-item similarity matrix in order to add additional ratings to the user-item rating matrix. The ratings are added in the following manner: for each missing user rating, we find all the items whose similarity (in the matrix created in the previous section) to the analyzed item is greater than 0. We then use the following formula to calculate each item's artificial rating:

The rating of each unrated item i for user u is calculated by the set of items K that were rated by the user u in the following way:

$$\forall \; rating(u,i) = \frac{\sum_{k \in K} rating(u,k) * sim(k,i)}{\sum_{k \in K} sim(k,i)}$$

After the completion of this step, the missing values of user-item rating matrix are filled with the calculated values thus reducing the sparsity of the matrix. The method does not provide ratings for every item; only to those user-items pairs for which there are sufficiently similar (defined by a threshold) items that were also ranked by the user.

We would like to emphasize that *the ratings calculated here are not the ratings presented to the user*. These ratings are used only to reduce the sparsity of the matrix, thus improving the performance of the standard CF. The prediction itself is done by a standard CF system (see section 3.3). Experiments we conduct show that this approach produces better results than using our method to directly generate the ratings.

3.3 The Prediction Phase

As mentioned before, all predictions are calculated using item-item collaborative filtering. When providing a prediction for a user-item rating the CF algorithm relies, whenever possible, on the "real" ratings of the training set rather than on the artificial ratings. The reason for this is that the artificial ratings are merely an estimation and therefore are less accurate.

If a user u rated a sufficient number of items to exceed a predefined threshold, the original user-item rating matrix is used by the CF algorithm for predicting the (u,i) rating.

If the number of "real" ratings of user u is smaller than the predefined threshold, the original user-item matrix is enriched by our similarity model. If user rated no items, all operations are performed on a matrix that includes the artificial ratings.

4. Evaluation

We evaluated the proposed method using the MovieLens dataset. This dataset contains 943 users and 1682 items.

In order to evaluate the contribution of the proposed method, we have compared it to a standard item-item similarity recommender system, using adjusted cosine similarity in order to calculate the similarity among items and a weight sum prediction in order to calculate the prediction [10].

We used learning set sizes ranging between 5% and 80% of the total provided ratings, providing a sparsity of 99.68% to 95%, respectively. The learning sets were generated by randomly choosing, for each user, X% of its provided ratings.

We compared the performance based on the original matrix (containing only the original training data) to the matrix generated by our proposed methods. As can be seen in Figures 1 and 2, some of the proposed methods provided substantial improvement.

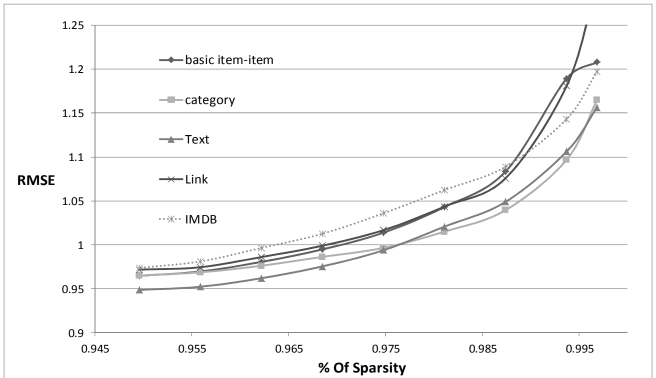

Figure 1: Comparison of individual pseudo rating methods

From figure 1 it can clearly be seen that the text and category-based methods provide a considerable improvement, while the links similarity does not. In addition, it is easy to see that the IMDB-based similarity yields significantly worse results than the common item-item similarity method.

IMDB's information on movies is typically shorter than the information found in Wikipedia (sometimes by orders of magnitude). In figure 2 we present the lengths of the movie descriptions in both sources (only movies for which we had descriptions both in Wikipedia and IMDB are presented).

We believe that IMDB's poor performance is due to the nature of the content. IMDB's plot information is short and uninformative. Wikipedia, on the other hand, usually provides ample information that includes detailed explanations on various aspects of the movie. For example, the movie "Toy Story" is described by 146 words in IMDB, while in Wikipedia it is described by approximately 6000.

Figure 3 presents the performance of the proposed methods when combining several item-item similarities (since the links similarity failed to produce significantly better results on its own, we have also combined only the text and categories similarities).

From Figure 3 it can be easily observed that the combination of several item-item similarity methods provides an additional boost to the performance of the model. In addition, as shown in Figure 3, the combined methods perform better than any method (text, category or links) on its own.

Figure 4 presents the relative improvement of the three similarity method and the category+link method over the basic item-item similarity.

From these experiments we draw the following conclusions:

1. It is observed that using the text and category information contained in Wikipedia significantly improves the accuracy of recommendations. This holds when using them separately and together.
2. The improvement is most significant in high sparsity.
3. The links similarity provides little improvement and on some sparsity levels actually produces inferior results. We assume that this is due to the high "noise" level of the links. For example, the movie "titanic" Wikipedia page links to the National Hockey League's page – one of the events in which the movie was promoted.

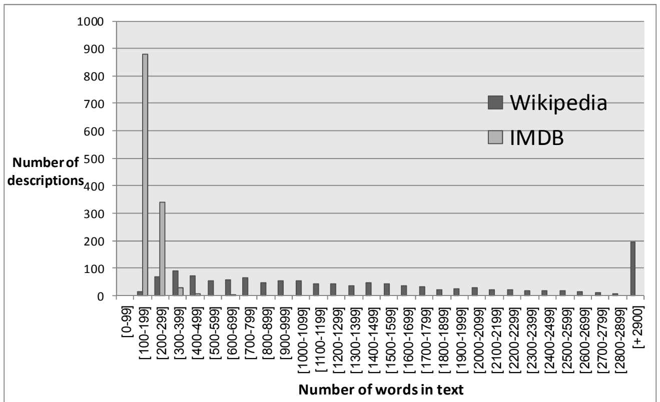

Figure 2: Comparison of words per movie

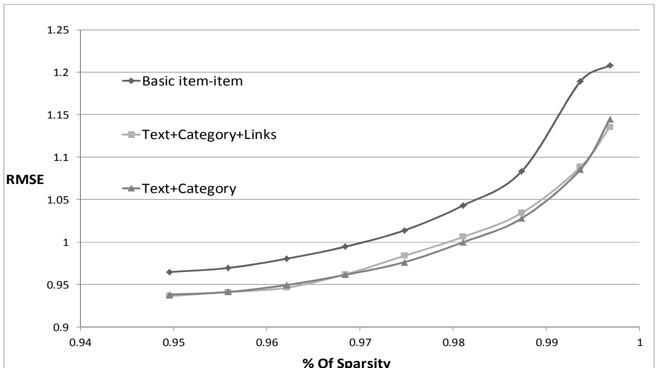

Figure 3: Comparison of combined pseudo rating methods

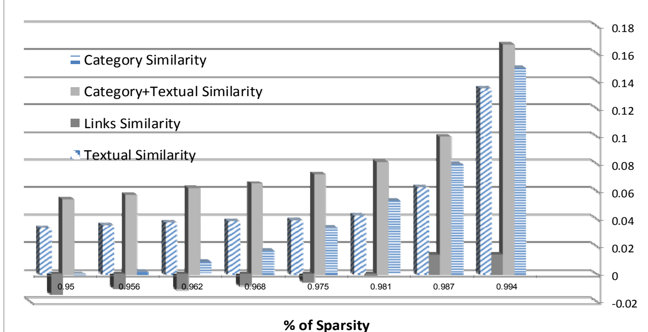

Figure 4: relative improvement over basic item-item similarity

4. The combination of several similarity methods yields better results than any method alone. We verified this conclusion by using paired-t tests with a confidence level of 95%. The combination of the category and text similarities seems to be slightly better than a combination of all three similarities, but we were not able to prove this hypothesis with sufficient confidence.
5. The information contained in IMDB is noticeably *inferior* to that found in Wikipedia (and using it in low sparsity may actually *harm* the results)

5. Conclusion

In this paper we examined the benefits of using Wikipedia as an external source of information for RS. We examined text, categories and links as means to augment the available data, and proposed a method for combining the ratings. Preliminary experiments have shown that text and the categories (which are tags assigned by readers) provide the greatest improvement.

In addition, we showed that the information stored in Wikipedia is more valuable in our approach than that of IMDB. We find it noteworthy that user generated content is more valuable for the purposes of collaborative filtering than information created by "experts". Furthermore, the information in Wikipedia is much more versatile and contains much more background information and elaboration on various aspects of the item (aspects that people find interest in, as it was added in the first place).

We believe that future work in this area consists of two possible directions: the first is the utilization of additional sources of information in Wikipedia (such as info boxes); the second is a more advanced use of the information sources presented in this paper. For example, a machine learning model may be able to infer which categories have greater impact than others on the similarity of items. Another option is to learn the cases in which collaborative filtering technique can perform better having artificial ratings.

6. REFERENCES

[1] Maltz, D. and K. Ehrlich, 1995. *Pointing the way: active collaborative filtering*, in Proceedings of the SIGCHI conference on Human factors in computing systems, ACM Press/Addison-Wesley Publishing Co.: Denver, Colorado, United States. p. 202-209.

[2] Adomavicius, G. and A. Tuzhilin, 2005. *Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions.* IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, 17(6).

[3] Melville, P., R.J. Mooney, and R. Nagarajan. 2002. *Content-Boosted Collaborative Filtering for Improved Recommendations*. AAAI-02 Proceedings,

[4] MIDDLETON, S.E., et al. 2002. *Exploiting Synergy Between Ontologies and Recommender Systems*. International Workshop on the Semantic Web, Proceedings of the 11th International World Wide Web Conference WWW-2002, Hawaii, USA.

[5] Semeraro, G., et al. 2009. *Knowledge infusion into content-based recommender systems.* in Proceedings of the third ACM conference on Recommender systems, ACM: New York, New York, USA. p. 301-304.

[6] Loizou, A. and S. Dasmahapatra. 2010 . *Using Wikipedia to alleviate data sparsity issues in Recommender Systems.* Semantic Media Adaptation and Personalization (SMAP), 2010. 5th International Workshop on p. 104 - 111.

[7] Lee, J.w., Lee S.g., and Kim, H.j.. 2011. *A probabilistic approach to semantic collaborative filtering using world knowledge.* Journal of Information Science. 37(1): p. 49-66.

[8] Salton, G. 1983. *Introduction to Modern Information Retrieval.* New York: McGraw Hill, p. 448.

[9] Sarwar, B., et al. 2001. *Item-based collaborative filtering recommendation algorithms*, in *Proceedings of the 10th international conference on World Wide Web*, ACM: Hong Kong, Hong Kong. p. 285-295