Recommendation by Example in Social Annotation Systems

Jonathan Gemmell, Thomas Schimoler, Bamshad Mobasher, and Robin Burke

Center for Web Intelligence
School of Computing, DePaul University
Chicago, Illinois, USA
{jgemmell,tschimoler,mobasher,rburke}@cdm.depaul.edu

Abstract. Recommendation by example is common in contemporary Internet applications providing resources similar to a user-selected example. In this paper this task is considered as a function available within a social annotation system offering new ways to model both users and resources. Using three real-world datasets we motivate several conclusions. First, a personalized approach outperforms non-personalized approaches suggesting that users perceive the similarity between resources differently. Second, the manner in which users interact with social annotation systems vary producing datasets with variable characteristics and requiring different recommendation strategies to best satisfy their needs. Third, a hybrid recommender constructed from several component recommenders can produce superior results by exploiting multiple dimensions of the data. The hybrid remains powerful, flexible and extensible despite the underlying characteristics of the data.

Keywords: Social Annotation Systems, Resource Recommendation, Recommendation by Example, Personalization, Hybrid Recommendation.

1 Introduction

Recommendation by example is a ubiquitous function of modern Web applications. Users select a resource and the system provides an ordered list of similar resources. In the context of music a user may be listening to a song and ask the system to recommend related music. The characteristics of that resource may represent the user's intention better than textual keywords ever could. Selecting a song from the recommendation list will in turn produce a new recommendation. In this manner the user can seamlessly navigate through the resource space.

This type of recommendation is commonplace but typically it is not personalized. While recommendation by example must take into account the selected resource, we believe that the recommender engine must also take advantage of the user preferences. Two users, by way of example, may ask for more movies like the Maltese Falcon. One user may mean, "give me more hard-boiled detective movies" regardless of actor or year. Another may mean, "show me more Humphrey Bogart movies" and may not necessarily be focused on mysteries. A third may be satisfied with popular movies from the 1940s. Leveraging a user's profile the system can personalize the results in order to identify similar resources from the viewpoint of the user.

C. Huemer and T. Setzer (Eds.): EC-Web 2011, LNBIP 85, pp. 209–220, 2011.

For our experimental study we restrict our analysis to a particular type of application common in the Social Web, social annotation systems, in which users annotate online resources with arbitrary labels often called tags. The user's selection of which tags to apply to a resource provides insights about which characteristic of the resource are important to the user. Moreover if the user has applied identical tags to two different resources we can assume that the two resources are similar from the viewpoint of that user. Other users may describe those resources differently. Working under these assumptions, social annotations systems permit an experimental methodology for studying personalized recommendation by example.

In this paper several recommenders are evaluated. We use a recommender based on cosine similarity as a starting point, a simple algorithm that one might expect in a recommendation by example scenario. We further propose a linear weighted hybrid which leverages several component recommenders. The hybrid permits the flexible integration of other recommenders with the cosine similarity model. These personalized models based on collaborative filtering emphasize the user model rather than the model of the example resource.

Our results conducted on three real world datasets reveal that 1) personalization improves the performance of recommendation by example in social annotation systems, 2) the hybrid effectively exploits multiple components to produce superior results, 3) differences in the datasets require an emphasis on different components and 4) the flexibility of the proposed hybrid framework enables it to adapt to these differences.

The rest of the paper is organized as follows. In Section 2 we position our contribution to the field with regard to similar efforts. In Section 3 we formalize the notion of personalized resource recommendation by example and describe our linear-weighted hybrid in addition to the components from which it is formed. Our experimental results and evaluation follow in Section 4. Finally, we conclude the paper with a discussion of our results.

2 Related Work

The recommendation by example paradigm has long been a core component of E-commerce recommender and information retrieval systems [26,31]. Early important approaches to the problem include association rule mining [1] and content-based classification [16]. Work by Salton and Buckley [23] demonstrated the importance of user feedback in the retrieval process. Content-based filtering has been combined with collaborative filtering in several ways [2,3,20] in order to improve prediction effectiveness for personalized retrieval. More generally, hybrid recommender systems [4] have been shown to be an effective method of drawing out the best performance among several independent component algorithms. Our work here draws from this prior work in applying a hybrid recommender to the domain of social annotation systems and specifically accommodating a recommendation by example query.

There has been considerable work on the general recommendation problem in social annotation systems. Generalizable latent-variable retrieval model for annotation systems [30] can be used to determine resource relevance for queries of several forms. Tagging data was combined with classic collaborative filtering in order to further filter a user's domain of interest [27]. More recently, several techniques [12,13,17] have built upon and refined this earlier work. None of these approaches, however, deal with the possibility of resources themselves as queries.

Some work has been done in regards to resource-to-resource comparison in social annotation, although little in the way of direct recommendation. Some have considered the problem of measuring the similarity of resources (as well as tags) in a social annotation system by various means of aggregation [18]. An author-topic latent variable model has been used in order to determine web resources with identical functionality [21]. They do not, however, specifically seek to recommend resources to a particular user, but rather simply enable resource discovery utilizing the annotation data.

Our own previous work regarding annotation systems has focused on the use of tag clusters for personalized recommendation [11,29] and hybrid recommenders for both tag [6] and resource [7,8,10] recommendation. Here we extend our examination of a linear-weighted hybrid of simple algorithms for the specific problem of recommendation by example. This work further demonstrates the versatility and effective performance of this framework.

3 Recommendation by Example in Social Annotation Systems

We define resource recommendation as the production of an ordered list of resources likely to be of interest to a particular user. A special case of resource recommendation is one in which the user supplies a resource as an example. The system is required to recommend resources similar to the example. Taken as a sequence of recommendations the user can navigate from resource to resource exploring the resource space. This type of browsing is commonplace in applications recommending music, journal articles or consumer products just to name a few.

A key conjecture in this work is that personalization can be used to improve the user experience. Two users may perceive the similarity between resources differently. They may, for example, like a particular song but for different reasons. One enjoys the guitar solos. The other is influenced by the vocals. If these two users were to ask for similar songs, the recommendation engine must accommodate the differences in their taste.

In this work we limit our investigation of recommendation by example to social annotation systems, which enable users to annotate resources with tags. The collection of users, resources and tags provide a rich environment for users to explore. We call this space URT and view it as a three dimensional matrix in which an entry is 1 if u tagged r with t and is 0 otherwise.

A recommendation by example algorithm in this domain takes the form $\phi(u, r_q, r)$ where u is the user, r is potential recommendation and r_q is used by the recommender engine as an example. This function assigns a real-valued score to each potential recommendation describing its relevance to the user and the query resource. A system computing such a function can iterate over all possible resources and recommend the resources with the highest scores. The final result relieves users from the burden of information overload by providing a personalized view of the information space.

To tackle this problem we propose a linear-weighted hybrid algorithm constructed from simple components. The components vary in the information they capture. The models based on cosine similarity, for example, ignore the user profile and focuses on the example resource. The collaborative filtering algorithms focus more on the user profile. The hybrid is able to aggregate these simply models into a cohesive whole.

In general terms the hybrid is composed of recommendation components κ_1 through κ_k, whose output is combined by computing a weighted sum [4]. We assume that each component makes its own computation of the function $\phi_i(u, r_q, r)$. The output is normalized to be in the range $[0..1]$. Each component also has a weight α_i in the same range and we require that these values sum to 1. The hybrid is therefore defined as:

$$\phi(u, r_q, r) = \sum_{i=1}^{k} \alpha_i \phi_i(u, r_q, r) \tag{1}$$

To ascertain the correct α_i for each component we use a hill climbing technique, which is both simple and efficient. A subset of the data is selected as a holdout set for learning the algorithm parameters, including the α values. The α vector is initialized with random positive numbers. The recommender then operates over the holdout set, using the remaining data as training data. The accuracy of recommendations is calculated as described in Section 4.2. The vector is then randomly modified and tested again. If the accuracy is improved, the change is accepted; otherwise it is most often rejected. Occasionally a change to the α vector is accepted even when it does not improve the results in order to more fully explore the α space. Modifications continue until the vector stabilizes. Then the α vector is randomly reset and learning proceeds again.

Now we turn to the components that make up our hybrid. Many of these components rely on two-dimensional projections of the three dimensional annotation data [19]. Such projections reduce the dimensionality of the data, but sacrifice some of its informational content. For example, the relation between resources and tags can be defined as $RT(r, t)$, the number of users that have applied t to r.

$$RT(r, t) = \sum_{\forall u \in U} URT(u, r, t) \tag{2}$$

This notion strongly resembles the "bag-of-words" vector space model [24]. Similarly, we can produce a projection UT in which a user is modeled as a vector over the set of tags, where each weight, $UT(u, t)$, measures how often a user applied a particular tag across all resources. In all, there are six possible two-dimensional projections: UR, UT, RU, RT, TU, TR. In the case of UR, we have not found it useful to weight resources by the number of tags a user applies, as this is not always indicative of the user interest. Rather we define UR to be binary, indicating whether or not the user has annotated the resource.

CS_{rt} , CS_{ru} : Because users apply tags to resources, we can model resources as a vector of tags as taken from RT. This allows us to measure the cosine similarity between query resource and a potential recommendation. We call this technique CS_{rt}. However this approach is not personalized. Noticing that resources can also be described as a vector of users described by RU we can again use cosine similarity to judge the relevance of a resource to the example given by the user. We call this technique CS_{ru}.

KNN_{ur} , KNN_{ut} : These algorithms operate like the well-known user-based collaborative filtering algorithm [15,28]. We rely on a matrix of user profiles gathered from UR or UT. Depending on which projection is used we describe the component as either KNN_{ur} or KNN_{ut}. To make recommendations, we filter the potential neighbors to only those who have used the example resource r_q. We perform cosine similarity to find the k nearest neighbors and use these neighbors to recommend resources using a weighted sum based on user-user similarity. Filtering users by the query resource focuses the algorithm on the user's query but still leaves a great deal of room for resources dissimilar to the example. These approaches however are strongly personalized.

KNN_{ru} , KNN_{rt} : These algorithms are analogous to item-based collaborative filtering [5,25], which relies on discovering similarities among resources rather than among users. The projections RU (resources as vectors of users) and RT (resources as vectors of tags) are employed. This procedure ignores the query resource entirely, instead focusing on the similarity of the potential recommendations to those the user has already annotated. Again a weighted-sum is used as in common in collaborative filtering.

Each of these algorithms exploits different dimensions of the data and each has their own benefits and drawbacks. CS_{rt} focuses on the similarity between two resources but ignores the user preferences. KNN_{ru} and KNN_{rt} disregard the query resource and concentrates on the user history. Instead of attempting to integrate all dimensions of the data into a single cumbersome algorithm we rely on the hybrid to leverage the benefits of each of its component recommenders.

It should be noted that other integrative techniques have been proposed to leverage multiple dimensions of the data. In particular graph based approaches such as Adapted PageRank [14] and tensor factorization algorithms such as Pairwise Interaction Tensor Factorization [22] ($PITF$) have meet with great success in tag recommendation. However the computational requirements of Adapted Pagerank make it ill-suited for large scale deployment; a Pagerank vector must be calculated for each recommendation.

$PITF$ on the other hand offers a far better running time. Nevertheless adapting the tag recommendation algorithm to resource recommendation is not straightforward. First, $PITF$ prioritizes tags from both a user and resource model in order to make recommendations thereby reusing tags. In resource recommendation the algorithm cannot promote resources from the user profile as these are already known to the user. This requirement conflicts with the assumptions of the prioritization model; all possible recommendation are in effect treated as negative examples.

Second, tensor factorization methods, $PITF$ included, normally require an element from two of the data spaces in order to produce elements from the third. For example a user and resource can be used to produce tags. In recommendation by example the input is a user and a resource while the expected output also comes from the resource space. Furthermore in our investigation into tag-based resource recommendation [9], we found that collaborative filtering algorithms often outperform $PITF$. A fundamental advantage of the proposed linear weighted hybrid framework in comparison to other integrative models is that it can be adapted to wide variety of recommendation tasks.

4 Experimental Evaluation

In this section we describe the methods used to gather and pre-process our datasets. Our evaluation metrics and methodology are described. We then examine the results for each dataset, and finally draw some general conclusions.

4.1 Datasets

Our experiments were conducted using data from three large real-world social annotation systems. On all datasets we generate p-cores [14]. When possible we constructed 20-cores from the datasets. If the dataset was not large enough to render a 20-core, we instead constructed a 5-core.

Citeulike is a popular online tool used by researchers to manage and catalog journal articles. The site owners make their dataset freely available to download. Once a 5-core was computed, the remaining dataset contains 2,051 users, 5,376 resources, 3,343 tags and 105,873 annotations.

Amazon is America's largest online retailer. The site offers a myriad of ways for users to express opinions of the products. Recently Amazon has added social annotations to this list. After taking a 20-core of the data, it contained 498,217 annotations with 8,802 users, 10,679 resource and 5,559 tags.

LastFM users upload their music profiles, create playlists and share their musical tastes online. Users have the option to tag songs, artists or albums. The tagging data here is limited to album annotations. A p-core of 20 was drawn from the data. It contains 2,368 users, 2,350 resources, 1,141 tags and 172,177 annotations.

4.2 Methodology

While recommendation by example is an important area of study there does not exist to our knowledge social annotations datasets in which a user has explicitly stated he believes two items are similar. However in these systems a user applies tags to resources, in effect describing it in a way that is important to the user. We work under the assumption that if a user annotates two resources in the same way then these two resources are from the viewpoint of the user similar. Segmenting the results into cases in which one, two, three, four or five tags are in agreement allow us to analyze the results when there is very high probability that two resources are similar (when a user applies several similar tags to the resources) or when the probability is lower (when only a single tag is applied to both resources).

For each data set, we evenly divide it into five equal partitions. Four partitions were used as training data and the fifth was used for the learning of the parameters including the number of neighbors for the collaborative filtering approaches and the α values for the linear-weighted hybrid. That partition was then discarded and four-fold cross-validation was performed using these remaining four partitions. One partition P_h was selected as a holdout set and the remaining partitions served as training data for the recommenders.

To evaluate the recommendation by example algorithms, we iterated over all annotations in P_h. Each annotation contains a user, a resource and a set of tags applied by

Table 1. The α values for the components of the linear-weighted hybrid

	CS_{rt}	CS_{ru}	KNN_{ur}	KNN_{ut}	KNN_{ru}	KNN_{rt}
Citeulike	0.332	0.014	0.145	0.037	0.046	0.426
LastFM	0.077	0.082	0.035	0.075	0.682	0.049
Amazon	0.129	0.004	0.402	0.085	0.088	0.291

the user to the resource. We compare these tags to the tags in the user's annotations from the training data. If there is a match we generate a test case consisting of the user, the resource from the training data as the example resources and the resource from the holdout data as the target resource. The example resource and target resource may have one tag in common or several. We evaluate these cases separately looking at as many as five matching tags.

We use recall to evaluate the recommenders. It measures the percentage of items in the holdout set that appear in the recommendation set. Recall is a measure of completeness and is defined as $|R_h \cap R_r|/|R_h|$ where R_h is a set containing the target resource and R_r is the set containing the recommendations. We measure recall in the top 10 recommended items. Since each test case has only one target resource this measure is also known as *hit ratio*. The results are averaged for each user, averaged over all users, and finally averaged over all four folds.

4.3 Experimental Results

Figures 1 through 3 shows the results of our experiments. As per Section 4.2 we identify cases in which the user has annotated two resources with the same tags. Table 1 presents the learned α values of the component recommenders. The sum of these values is 1 and represents the relative contribution on the components. In the remainder of this section we discuss each dataset individually before concluding our paper with the general findings.

Citeulike. Citeulike users annotate journal articles. Its members are therefore mostly comprised of researches using the system to organize their library of related work. They often use tags drawn from their field and focus on articles within their area of expertise. The result is relatively clean datasets with strong dimensions relating tags to users and resources.

CS_{rt} does very well as one might expect. The tags applied to resources are indicative of their characteristics and using this data allows the algorithm to discover similar resources. KNN_{rt} is the second best performing component. This result is perhaps more surprising since it completely ignores the example resource. However this result is explainable by the nature of the user interactions found in Citeulike; users are focused on their area of expertise. Resources and tags in the user profile are strongly indicative of the user's interest making the user profile quite focused.

When performing item-based collaborative filtering in Citeulike it appears better to model resources as tags rather than users as shown by the relative performance of KNN_{ru} and KNN_{rt}. Likewise the cosine similarity model which describes resources

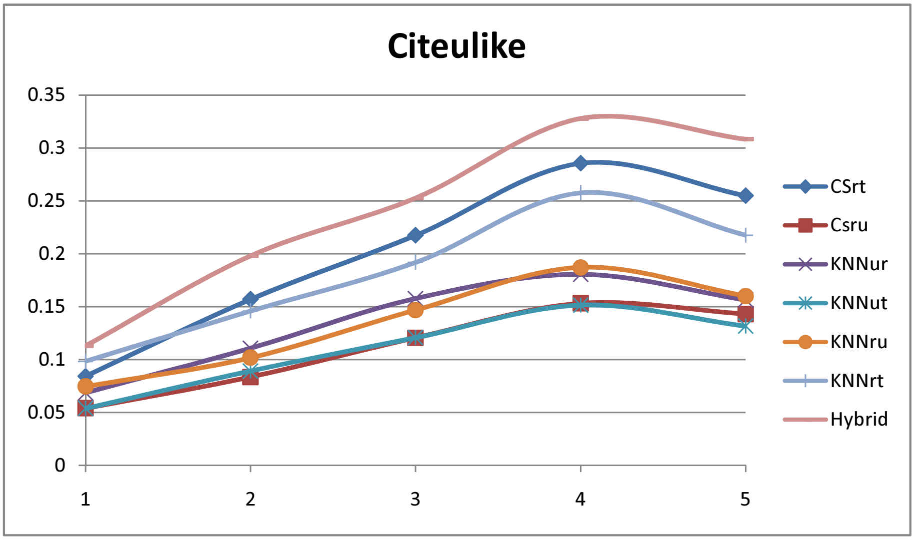

Fig. 1. Citeulike: The hit-ratio for six recommendation by examples algorithms for cases where there is an agreement of one through five tags. The hybrid is composed of all six techniques.

as tags outperform the method which models them as users. These results underscore the care which users exhibit when assigning keywords to resources, likely because they employ Citeulike to organize resources for latter retrieval.

The hybrid outperforms its constituent parts by as much as 5 percent. The α values shown in Table 1 reveal that the hybrid relies most strongly on CS_{rt} and KNN_{rt} the two strongest individual components. Yet KNN_{ur} also makes a strong showing accounting for almost 15% of the hybrid. This result shows that even though a technique may perform poorly alone, it may contribute unique information to a hybrid.

LastFM. LastFM users share their musical tastes and discover new music online. The site has evolved considerable overtime, but still allows its users to tag music (songs, artists or albums). As opposed to Citeulike its users take considerably less care when applying tags to resources. Generic tags such as 'rock' or non-descriptive tags such as 'album_i_own' are common. More often however the users interact with one another explicitly forming friendships, joining groups, comparing music tastes or browsing each other's profiles. This observation is confirmed in the relative performance of CS_{rt} and CS_{ru} as well as KNN_{ru} and KNN_{rt}. The user space is far more developed and modeling resources as users produces better results than modeling them as tags.

The collaborative approaches KNN_{ru} and KNN_{ur} which largely ignores the example resource outperforms the cosine similarity models which focuses entirely on the example. This result implies that in the music domain a user's profile is more important than the example he provides in a recommendation by example scenario.

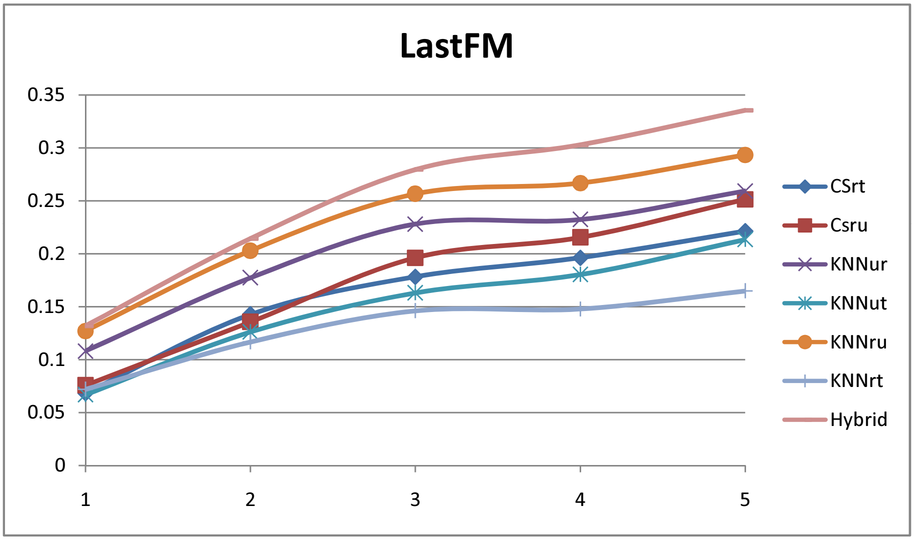

Fig. 2. LastFM: The hit-ratio for six recommendation by examples algorithms for cases where there is an agreement of one through five tags. The hybrid is composed of all six techniques.

The hybrid again outperforms its individual components again by as much as 5 percent. However, the contributions of the components differ. The hybrid is dominated by KNN_{ru} and the remaining components offer single digit contributions. In contrast to Citeulike which has several strong dimensions, LastFM's data is narrowed to the user-resource dimension. The remaining components play a small roll in the hybrid, but their complimentary information provide enough additional information to improve the performance of the hybrid.

Amazon. At the Amazon Web site customers are allowed to tag products. Often these tags are drawn from the product description such as 'HDTV'. Also the product space is easily separable – clothes and books, or mysteries and romance. Customers often focus on a few of these interests rather than annotating several disparate items. These characteristics make the Amazon data an easier target permitting as much as 70 percent hit ratio.

The simple components whether they draw on the relation between users and resources or resources and tags all perform equally well. Their equivalent performance might lead one to think that they are interchangeable. To the contrary when aggregated into a hybrid the proposed framework is able to leverage the benefits offered from each component. The individual components are exploiting different dimensions of the data.

While the hybrid outperforms its components once again we also see that it relies on different algorithms to do so. KNN_{ur} is the dominate recommender followed by KNN_{rt}. This is suggested by an understanding of how users interact with the Web site. They focus on particular domains forming a strong user-resource relation. Moreover

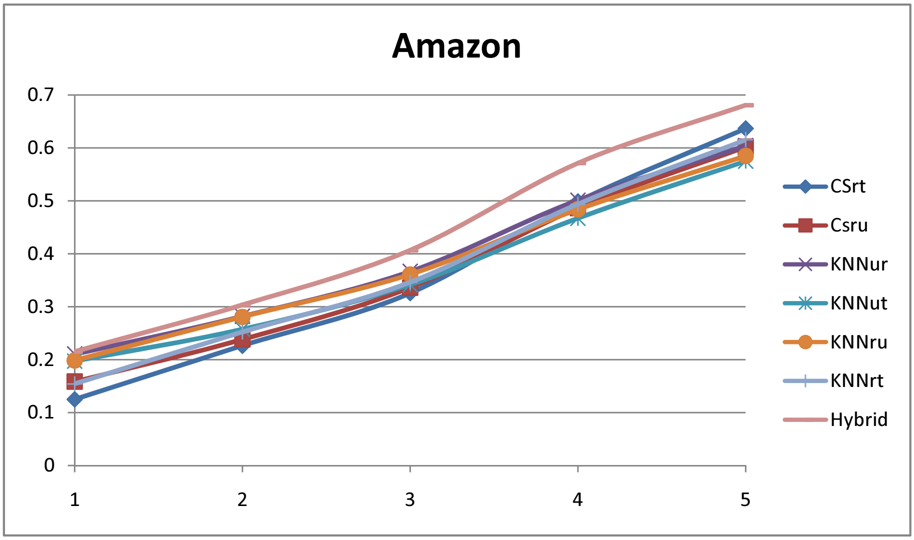

Fig. 3. Amazon: The hit-ratio for six recommendation by examples algorithms for cases where there is an agreement of one through five tags. The hybrid is composed of all six techniques.

they often use preconceived tags generated strong user-tag and resource-tag connections. These two component, leveraging different dimensions of the data, work together in order to offer meaningful advantages to the hybrids.

5 Conclusion

In this work we have investigated recommendation by example for use in social annotation systems. This type of user interaction offers a great deal of utility to users as they explore very large resource spaces. Our belief that personalization is important to satisfying the user's needs is confirmed with experimentation using three real world datasets. In order to blend the benefits of personalization with techniques focused on the example we proposed a linear-weighted hybrid. The hybrid was able to effectively exploit multiple components to produce superior results even though differences in the datasets required an emphasis on different components.

Our proposed linear-weighted hybrid offers additional advantages. It can exploit multiple dimensions of the data, while maintaining the speed and simplicity of if its parts. Second, it is extensible allowing additional components based on the underlying data. For example, systems that include ratings or allow users to generate friendship links can exploit this information by adding additional components to the hybrid. Finally, by analyzing the relative contributions of the components one can gain insights into how the components interact and reveal interesting patterns of user behavior.

Acknowledgment. This work was supported in part by a grant from the Department of Education, Graduate Assistance in the Area of National Need, P200A070536.

References

1. Agrawal, R., Imielinski, T., Swami, A.N.: Mining Association Rules between Sets of Items in Large Databases. In: Buneman, P., Jajodia, S. (eds.) Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, D.C, pp. 207–216 (1993)
2. Balabanović, M., Shoham, Y.: Fab: content-based, collaborative recommendation. Commun. ACM 40(3), 66–72 (1997)
3. Basu, C., Hirsh, H., Cohen, W.W.: Recommendation as Classification: Using Social and Content-Based Information in Recommendation. In: AAAI/IAAI, pp. 714–720 (1998)
4. Burke, R.: Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction 12(4), 331–370 (2002)
5. Deshpande, M., Karypis, G.: Item-Based Top-N Recommendation Algorithms. ACM Transactions on Information Systems 22(1), 143–177 (2004)
6. Gemmell, J., Ramezani, M., Schimoler, T., Christiansen, L., Mobasher, B.: A fast effective multi-channeled tag recommender. In: European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases Discovery Challenge, Bled, Slovenia (2009)
7. Gemmell, J., Schimoler, T., Mobasher, B., Burke, R.: Hybrid tag recommendation for social annotation systems. In: 19th ACM International Conference on Information and Knowledge Management, Toronto, Canada (2010)
8. Gemmell, J., Schimoler, T., Mobasher, B., Burke, R.: Resource Recommendation in Collaborative Tagging Applications. In: E-Commerce and Web Technologies, Bilbao, Spain (2010)
9. Gemmell, J., Schimoler, T., Mobasher, B., Burke, R.: Tag-based resource recommendation in social annotation applications. In: User Modeling, Adaptation and Personalization, Girona, Spain (2011)
10. Gemmell, J., Schimoler, T., Ramezani, M., Christiansen, L., Mobasher, B.: Resource Recommendation for Social Tagging: A Multi-Channel Hybrid Approach. In: Recommender Systems & the Social Web, Barcelona, Spain (2010)
11. Gemmell, J., Shepitsen, A., Mobasher, B., Burke, R.: Personalizing Navigation in Folksonomies Using Hierarchical Tag Clustering. In: 10th International Conference on Data Warehousing and Knowledge Discovery, Turin, Italy (2008)
12. Guan, Z., Wang, C., Bu, J., Chen, C., Yang, K., Cai, D., He, X.: Document recommendation in social tagging services. In: Proceedings of the 19th International Conference on World Wide Web, WWW 2010, pp. 391–400. ACM, New York (2010)
13. Guy, I., Zwerdling, N., Ronen, I., Carmel, D., Uziel, E.: Social media recommendation based on people and tags. In: Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, pp. 194–201. ACM, New York (2010)
14. Jäschke, R., Marinho, L., Hotho, A., Schmidt-Thieme, L., Stumme, G.: Tag recommendations in folksonomies. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) PKDD 2007. LNCS (LNAI), vol. 4702, pp. 506–513. Springer, Heidelberg (2007)
15. Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., Riedl, J.: GroupLens: Applying Collaborative Filtering to Usenet News. Communications of the ACM 40(3), 87 (1997)
16. Lewis, D.D., Schapire, R.E., Callan, J.P., Papka, R.: Training algorithms for linear text classifiers. In: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1996, pp. 298–306. ACM, New York (1996)
17. Liang, H., Xu, Y., Li, Y., Nayak, R., Tao, X.: Connecting users and items with weighted tags for personalized item recommendations. In: Proceedings of the 21st ACM conference on Hypertext and Hypermedia, HT 2010, pp. 51–60. ACM, New York (2010)

18. Markines, B., Cattuto, C., Menczer, F., Benz, D., Hotho, A., Gerd, S.: Evaluating similarity measures for emergent semantics of social tagging. In: Proceedings of the 18th International Conference on World Wide Web, WWW 2009, pp. 641–650. ACM, New York (2009)
19. Mika, P.: Ontologies are us: A unified model of social networks and semantics. Web Semantics: Science, Services and Agents on the World Wide Web 5(1), 5–15 (2007)
20. Pennock, D.M., Lawrence, S., Popescul, R., Ungar, L.H.: Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments. In: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence, pp. 437–444 (2001)
21. Plangprasopchok, A., Lerman, K.: Exploiting Social Annotation for Automatic Resource Discovery. In: Proceedings of AAAI Workshop on Information Integration (April 2007)
22. Rendle, S., Schmidt-Thieme, L.: Pairwise Interaction Tensor Factorization for Personalized Tag Recommendation. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining, New York (2010)
23. Salton, G., Buckley, C.: Improving retrieval performance by relevance feedback, vol. 41, pp. 288–297. Wiley, San Francisco (1990)
24. Salton, G., Wong, A., Yang, C.: A Vector Space Model for Automatic Indexing. Communications of the ACM 18(11), 613–620 (1975)
25. Sarwar, B., Karypis, G., Konstan, J., Reidl, J.: Item-Based Collaborative Filtering Recommendation Algorithms. In: 10th International Conference on World Wide Web, Hong Kong, China (2001)
26. Schafer, J.B., Konstan, J.A., Riedl, J.: E-Commerce Recommendation Applications. Data mining and knowledge discovery 5(1), 115–153 (2001)
27. Sen, S., Vig, J., Riedl, J.: Tagommenders: connecting users to items through tags. In: WWW 2009: Proceedings of the 18th International Conference on World Wide Web, pp. 671–680. ACM, New York (2009)
28. Shardanand, U., Maes, P.: Social Information Filtering: Algorithms for Automating Word of Mouth. In: SIGCHI Conference on Human Factors in Computing Systems, Denver, Colorado (1995)
29. Shepitsen, A., Gemmell, J., Mobasher, B., Burke, R.: Personalized Recommendation in Social Tagging Systems using Hierarchical Clustering. In: ACM Conference on Recommender Systems, Lausanne, Switzerland (2008)
30. Wu, X., Zhang, L., Yu, Y.: Exploring social annotations for the semantic web. In: Proceedings of the 15th International Conference on World Wide Web, Edinburgh, Scotland (2006)
31. Yang, Y., Chute, C.G.: An example-based mapping method for text categorization and retrieval. ACM Trans. Inf. Syst. 12, 252–277 (1994)