Collaborative Tagging in Recommender Systems

Ae-Ttie Ji, Cheol Yeon, Heung-Nam Kim, and Geun-Sik Jo

Intelligent E-Commerce Systems Laboratory, Inha University, Korea
{aerry13, entireboy, nami}@eslab.inha.ac.kr, gsjo@inha.ac.kr

Abstract. This paper proposes a collaborative filtering method with user-created tags focusing on changes of web content and internet services. Collaborative tagging is employed as an approach in order to grasp and filter users' preferences for items. In addition, we explore several advantages of collaborative tagging for future searching and information sharing which is used for automatic analysis of user preference and recommendation. We present empirical experiments using real dataset from *del.icio.us* to demonstrate our algorithm and evaluate performance compared with existing works.

1 Introduction

The prevalence of digital devices and the development of Internet technologies and services enable end-users to be a producer as well as a consumer of media content. Even in a single day, an enormous amount of content including digital video, blogging, photography and wikis is generated on the web. It's getting more difficult to make a recommendation to a user about what he/she prefers among these items automatically because of not only their huge amount, but also the difficulty of automatically grasping of their meanings.

Collaborative filtering (CF) is an efficient approach to treat the above issues [3, 5, 6, 11, 13]. CF has an advantage over content-based filtering which is the ability to filter any type of items, e.g. text, music, videos and photos [6]. Because the filtering process is only based on historical information about whether or not a target user has preferred an item before, analysis of actual content itself is not necessarily required.

Collaborative tagging describes the process that allows many users to annotate content with descriptive keywords, i.e. tags [1, 12, 15]. Tagging is not new, but has recently become useful and popular as one effective way of classifying items for future search, sharing information and filtering [1, 15]. In terms of user-created tags, they imply users' preferences and opinions about items as well as metadata about them.

In our research, we propose a CF method with user-created tags focusing on changes of web content and internet services. Collaborative tagging is employed as an approach in order to grasp and filter users' preferences for items. The next section describes an overview of recent studies related to tagging. Our main contribution is a novel approach of recommendation systems with collaborative tagging. This approach is described in section 3. Section 4 presents the effectiveness of our approach through the experiments comparing our approach with existing works using *del.icio.us* (http://del.icio.us/) data. We conclude with a discussion and future directions.

M.A. Orgun and J. Thornton (Eds.): AI 2007, LNAI 4830, pp. 377–386, 2007.

2 Collaborative Tagging and Folksonomy

Collaborative tagging is the practice of allowing any user to freely annotate to content with any kind of tags [1, 15]. In some circumstances, such as the web, where there is no "librarian" to classify items or there are too many items to classify by a single authority, collaborative tagging is one of the most useful ways of categorizing or indexing content [1]. Moreover, tags are directly published and discussed on the web and may be applied to any kinds of items, even people [10]. Collaborative tagging can play a key role in sharing content in social networks [15].

Collaborative tagging is described as "folksonomy," in contrast with typical "taxonomy," even though there is some debate about the appropriateness of this term [4]. In contrast with taxonomy, tagging performs a horizontal and inclusive way for classification and therefore can have an advantage over hierarchical taxonomy in some cases. In taxonomy, a category with a more general concept includes more specific ones. Even though a hierarchical category assures a user that all the items exist in one corresponding stable place, the user cannot be sure that all relevant items are returned by a query. To avoid fruitless searching, the user needs to check multiple locations. Unlike a hierarchical search, in a collaborative tagging system such items can be annotated with a variety of terms simultaneously; general tags and specific ones. In addition, tags can filter out all relevant items and return only those items tagged with those tags. As users can provide tags without any intricate implementation, a tagging system can be an effective and easy way to help identify correct items and make search results more relevant. Golder and Huberman have discussed about such advantages over taxonomy as well as the other significant issues of tagging systems [1].

Marlow et al. define several dimensions of tagging system design according to their possible implications [2]. We will review two of them related to our work briefly. From the user's right of tagging behavior, a tagging system can be classified into self-tagging, permission-based and free-for-all. Self-tagging, where users only tag the content they created for future personal retrieval, is provided by *Technorati* (http://www.technorati.com/) and *YouTube* (http://www.youtube.com/). Like in *Flickr* (http://www.flickr.com/), permission-based tagging is provided as specifying different levels of permission to tag. These two forms of tagging are also mentioned as narrow folksonomies [12, 14], and strictly speaking, they partially or do not support *collaborative* tagging [1]. *Del.icio.us* and *Yahoo! MyWeb* (http://myweb.yahoo.com/), which provide free-to-all tagging, allow any user to tag any items. Free-to-all tagging is also known as a broad folksonomy [12, 14]. According to the aggregation of tags, a tagging system is divided into a bag-model and a set-model. A set-model does not allow any repetition of tags, and so the system shows users only the "set" of tags attached for the item (e.g., *Flickr*, *YouTube* and *Technorati*). In contrast with set-model, a bag-model system allows duplicated tags for the same item from different users (e.g., *del.icio.us*, *Yahoo! MyWeb*). Based on the statistical information of tag frequencies, the system is able to present the item with the collective opinions of the taggers.

Especially in broad folksonomy, a tag frequency of an item tends to lead a long-tail curve (i.e., power law curve or power curve) [14], as the majority of the tags attached to the item are popular or common tags of the item. These popular tags with high frequencies can represent the content of the item.

3 Recommender System with Collaborative Tagging

Fig. 1 illustrates our method with two phases: Candidate Tag Set (*CTS*) generation and probabilistic recommendation. *CTS,* which implies the latent preference of a target user, includes the tags filtered with the CF scheme. Based on *CTS*, a *Naïve Bayes Classifier* is applied to decide which items to recommend stochastically.

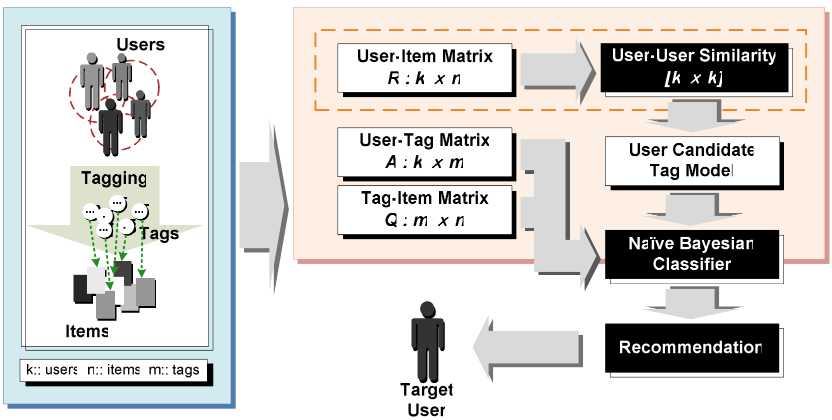

Fig. 1. Architecture of a Tag-based Collaborative Filtering System

Three matrices including preference history of users are used throughout this article.

User-item binary matrix, *R*. A list of k users $U = \{u_1, u_2, \ldots, u_k\}$ and a list of n items $I = \{i_1, i_2, \ldots, i_n\}$ can be represented as a *User-item binary matrix*, $R\ (k \times n)$. Each $R_{u,i}$ has 1 if a user u has selected (or tagged) an item i or 0 otherwise.

User-tag matrix, *A*. For a set of m tags $T = \{t_1, t_2, \ldots, t_m\}$, tag usages of k users can be represented as a *User-tag matrix*, $A\ (k \times m)$. Each $A_{u,t}$ represents the frequency of meaning how many times a user u has been tagging with a tag t.

Tag-item matrix, *Q*. *Tag-item matrix,* $Q\ (m \times n\ )$ includes tag frequency for n items. Each $Q_{t,i}$ implies the number of tag t tagged for item i by users.

3.1 Candidate Tag Set Generation

Candidate Tag Set, *CTS*. The concept of *CTS* starts from assuming that a target user is likely to prefer the items tagged with the tags, that is, the tags that have been used by similar users or by a target user before. *CTS* includes a set of tags which implied a target user's latent preference and is generated using CF scheme based on *user-tag matrix*, *A*. $CTS_w(u) = \{t_x | \ x=1,2, \ldots,w,\ t_x \in T\}$ indicates the *CTS* of user u included in the set of all tags, T and w means the number of candidate tags.

User-user Similarity. In order to find k nearest neighbor (*KNN*), cosine similarities between a target user and each users with tag frequencies of corresponding user in *user-tag matrix, A*. *KNN* includes users who have higher similarity score than the other users and means a set of users who prefer more similar tags with a target user. The similarity relationship between user *u* and *v*, *sim*(*u,v*) is defined as:

$$sim(u, v) = \cos(\vec{u}, \vec{v}) = \frac{\sum_{t \in T} A_{u,t} \cdot A_{v,t}}{\sqrt{\sum_{t \in T} (A_{u,t})^2} \sqrt{\sum_{t \in T} (A_{v,t})^2}} \tag{1}$$

$A_{u,t}$ and $A_{v,t}$ denotes the tag frequencies of user *u* and *v* each in *user-tag matrix, A*. The similarity score between two users takes a real number between 0 and 1 and the higher score a user has, the more similar he is to a target user.

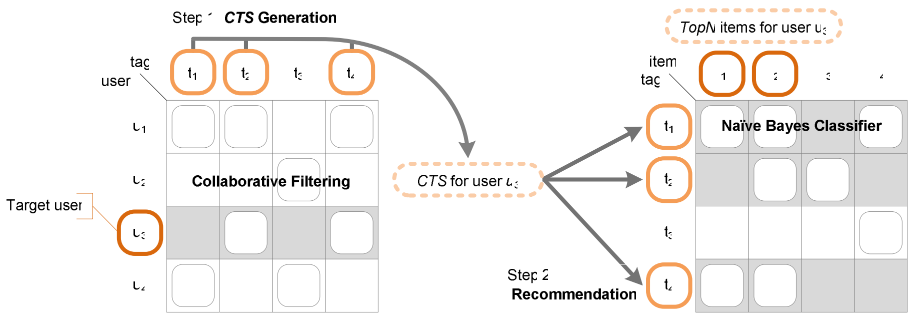

Fig. 2. Candidate Tag Set Generation via Collaborative Filtering

Tag Preference. The measurement of how much a user prefers a tag is given by [5]:

$$S_{u,t} = \sum_{o \in KNN(u)} (A_{o,t}) \cdot sim(u, o) \tag{2}$$

$S_{u,t}$ denotes the prediction value of user *u*'s preference for tag *t*. For each user *o* included in *KNN*(*u*), which is a set of *k* neighbors of user *u*, a sum of tag frequency $A_{u,t}$ weighted by *sim*(*u,o*), which is a similarity score between user *u* and *o*. The weighted sum leads to a higher prediction value for a more similar user to a target user. According to the highest-order of prediction value $S_{u,t}$, *w* tags are selected for generating $CTS_w(u)$. The algorithm for computing user-tag preferences is shown in Algotithm 1.

Recommendation from tags can partially improve the sparsity problem, which is one of the limitations of CF, due to the filtering process depending on the co-occurrence of items. For example, let us assume that Alice bookmarked two web pages, "Subway Map" and "Bus Map", Bob bookmarked "Subway Map", "Bus Map" and "Public Transportation Fares" and Cathy bookmarked "Public Transportation Fares". If Alice is a target user who needs to get a recommendation, the system provide "Public Transportation Fares" page for her from Bob, because Alice and Bob have co-bookmarked items, "Subway Map" and "Bus Map". However, the system cannot provide any items for Cathy or from her because she has no co-bookmarked items with other users, and so any similarity relationship between her and other users cannot be derived. In contrast, providing that Alice tagged "Subway Map" and "Bus Map" with *route map, public transportation* and *traffic* and Cathy tagged "Public Transportation Fares" with *public transportation* and *fare*, the system can provide the tag *fare* for Alice from Cathy because of co-tagged *public transportation*. Now, Alice can get recommendations of items tagged with *fare* as well as her own tags: *route map, public transportation* and *traffic* including “Public Transportation Fares” tagged by Cathy.

In addition, tags can be used as a more efficient means for modeling user’s preferences. Tags are able to support the detailed opinion of a user for a particular item and are easy to be changed and updated regardless of implementation. If a user’s preference is changed as time goes by, he can easily express his own opinion about the item by adding or changing tags.

Algorithm 1. Tag Preference Prediction Algorithm

```
Input: total user list U; size of KNN k; user-tag matrix A;
       user-user similarity matrix D;
Output: user-tag preference matrix S

Procedure computingTagPreference(U, k, A, D, S)
set all elements in matrix S to 0
for each u ∈ U
    for i ← 1 to r // r is row count of matrix U
        add D_{u,i} to itemset KNN
    for each x ∈ KNN // get KNN of each user
        if x ≠ among the k largest values in KNN then
            remove x from KNN
    for each t ∈ T // compute the preference matrix S
        for each x ∈ KNN
            S_{u,t} ← S_{u,t} + (A_{x,t} × D_{u,x})
```

3.2 Item Recommendation Via Naïve Bayes Approach

Based on a *CTS* model for each user, *top-N* items are recommended stochastically with a *Naïve Bayes Classifier* [9]. *Top-N recommendation* is one of the recommendation schemes offering a target user u the ordered set of items $TopN_u$ such that $|TopN_u| \leq N$ and $TopN_u \cap I_u = \varnothing$ [7]. I_u denotes a set of items that have been selected by a user u. Given a tag instance t_j in $CTS_w(u)$ as a set of feature variables, a Bayes classifier allows us to compute the *posterior* probability $P(I = i_y \mid t_j)$ which tag t_j was tagged for item i_y for each possible item i_y in a set of all items $I = \{i_1, i_2, \ldots, i_n\}$. A *priori* probability $P(I = i_y)$ and a item-conditional tag distribution $P(t_j \mid I = i_y)$ are computed as:

$$P(I = i_y) = \frac{\sum_{u=1}^{k} R_{u,y}}{\sum_{y=1}^{n}\sum_{u=1}^{k} R_{u,y}} \quad , \quad P(t_j \mid I = i_y) = \frac{1 + Q_{j,y}}{m + \sum_{t=1}^{m} Q_{t,y}} \tag{3}$$

$R_{u,y}$ and $Q_{j,y}$ denotes the binary value of u-th user for y-th item in user-item matrix, R and the tag frequency of j-th tag for y-th item in tag-item matrix, Q respectively. Avoiding that $Q_{j,y}$ turn out to be zero, we use *Laplace correction* in Equation 3 [8].

Let us assume that each feature t_j is conditionally independent of every other feature, the *posterior* probability as a preference probability $P_{u,y}$ of user u with $CTS_w(u)$ for an item i_y is given by:

$$P_{u,y} = P(I = i_y)\prod_{j=1}^{w} P(t_j | I = i_y) \quad (4)$$

Finally, the ordered set of items *TopN*$_u$ with the highest $P_{u,y}$ is recommended for a target user *u* [7]. The entire recommendation process proposed is described in Algotithm 2 with *TopN*$_u$ generation.

Algorithm 2. Recommender System with Collaborative Tagging

```
Input: total user list U; user-item matrix R; user-tag matrix
       A; tag-item matrix Q; size of CTS w; size of Top-N N;
       size of KNN k; items not rated by user u L_u

Method:
for each u ∈ U // generate user-user similarity matrix D
    for each v ∈ U
        if v ≠ u then
            D_{u,v} ← sim(u, v)
// generate user-tag preference matrix S
call generatingTagPreference (U, k, A, D)
for each u ∈ U { // recommending items to each user
    TopN_u ← generatingTopNItem (u, w, N, L_u, S)
    Recommending Top-N items to user u
}

Procedure generatingTopNItem(u, w, N, L_u, S)
// get CTS of user u from user-tag preference matrix S
for i ← 1 to m // m is column count of matrix S
    add S_{u,i} to itemset CTS_w(u)
for each x ∈ CTS_w(u)
    if x ≠ among the w largest values in CTS_w(u) then
       remove x from CTS_w(u)
for each i_y ∈ L_u
    add NaiveBayesClassifier(u, CTS_w(u), i_y, Q) to TopN_u
for each z ∈ TopN_u // recommend Top-N items to user u
    if P_{u,z} = 0 ∨ P_{u,z} ≠ the N largest values in TopN_u then
       remove z from TopN_u
return TopN_u
```

4 Experimental Results

In this section, we empirically evaluate the recommendation algorithm via collaborative tagging and compare its performance against the performances of user-based CF [11] and item-based CF [7]. The system prototype was implemented using *JDK* 5.0 and *MySQL* 5.0 and experiments were performed on *Dual Xeon* 3.0 GHz, 2.5GB RAM computers.

4.1 Dataset and Evaluation Metrics

Del.icio.us is a well-known social bookmark service supporting collaborative tagging. We collected our dataset by examining the *del.icio.us* site and constructed a 1544 × 17,390 user-item binary matrix, *R,* a 1544 × 10,077 user-tag matrix, *A* and a 10,077 × 17,390 tag-item matrix, *Q*. Sparsity level, which is defined as 1-(nonzero elements / total elements) [3], of the collected dataset is as follows; user-item matrix is 0.9989 and user-tag matrix is 0.9971. The dataset was divided into two parts; a training set containing 21,653 bookmarks (80%) and a test set containg 5,413 bookmarks (20%).

Table 1. The Dataset from *del.icio.us*

users	items	tags	book marking	Tagging
1,544	17,390	10,077	27,066	44,681

The performance was measured by looking at the number of items in the test set that were also included in $TopN_u$ recommended for a target user *u* by a particular scheme, which is also called *recall* [7, 11]. Hit-ratio for each target user *u* is given by:

$$hit - ratio(u) = \frac{|Test_u \cap TopN_u|}{|Test_u|} \tag{5}$$

$Test_u$ denotes a set of items tagged by a target user u. Average *recall* of all the *k* users in test set is given by:

$$recall = \frac{\sum_{u=1}^{k} hit - ratio(u)}{k} \times 100 \tag{6}$$

4.2 Performance Evaluation of Benchmark Algorithms

The neighborhood size has significant impact on the quality of results from collaborative filtering [4]. We evaluated the quality of user-based CF [11] and item-based CF [7] based on a user-item matrix, *R*, where we varied the size of the *k*-nearest neighbor, *KNN*, from 10 to 100. We set the number of returned items *N* to 10 for each user in the test set and computed *recall* for each algorithm [3].

Fig. 3 illustrates the variation of *recall* for each algorithm. It can be observed that the recommendation quality improves as the number of *k* is increased. After 50, the increase rate of a user-based scheme diminishes. Generally, an item-based scheme performed better than a user-based one. This is caused by the sparsity of the dataset is too high to compute user-user similarities [13]. With a relatively small neighborhood size, an item-based scheme outperformed a user-based one. However, because the number of items is far lager than the number of users, computation of item-item similarities took much longer than user-user similarities.

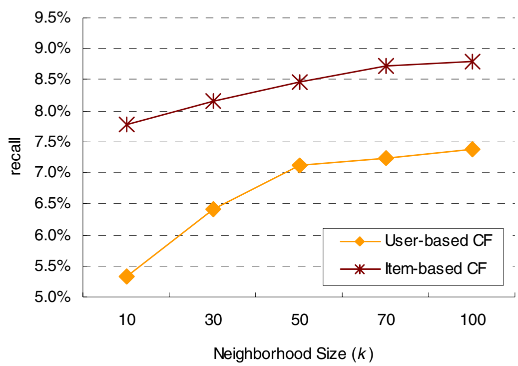

Fig. 3. The Variation of Recalls for User-based CF and Item-based CF

4.3 Experiments with Candidate Tag Set Size

We expected that the size of *CTS*, *w,* can be a significant factor affecting the quality of recommendation in our work, and so we evaluated our algorithm by measuring *recall* according to each size of *CTS* from 10 to 100. In order to obtain *CTS* based on user-tag matrix, *A*, user-based CF was used where *k* is set to 50. *N* is set to 10 through all evaluations.

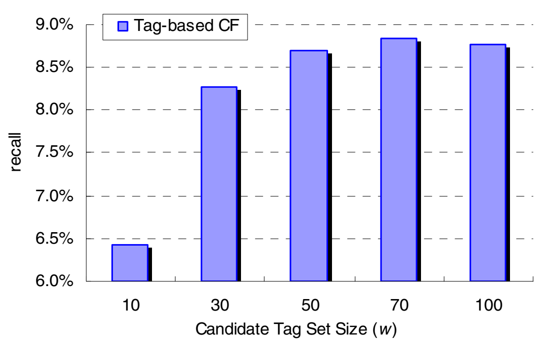

Fig. 4. The Variation of Recall of Tag-based CF

Fig. 4 shows that our algorithm tends to result in better quality as the size of *w* is increased. However, the quality of algorithm was rather worse when *w* was 100 (recall was 8.772%), whereas its recall was 8.839% which was the highest when *w* was 70. This result indicates that superfluous tags which do not represent user's preference can be included in *CTS*. That is, selecting too many numbers of tags can cause not only even bad impact on representing user's preference but also unnecessary cost for computation. For this reason, *w* should be selected within a reasonable level for experimentation, so we set it to 70, which obtained the best quality.

4.4 Comparisons of Performance

To experimentally compare the performance of our algorithm with those of user-based CF and item-based CF, we selectively varied the number of returned items *N* from 10 to 50 in an increment of 10. According to the results of the prior experiments in section 4.2 and 4.3, *k* and *w* were set to 50 and 70 respectively.

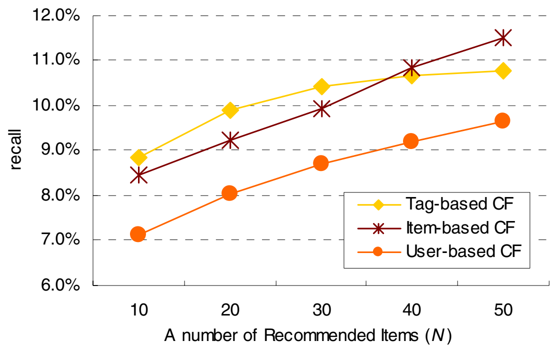

Fig. 5. Comparisons of Recall as the value of *N* increases

As we can see from these experiments, overall recall for all three algorithms was improved according to the increment of *N*. However, due to sparsity of the collected dataset from del.icio.us, all three methods did not perform well. These results were also affected by the number of items (17,390) being 10 times larger than the number of users (1,544). As shown in Fig. 5 nevertheless, tag-based CF outperformed user-based CF. In addition, the proposed algorithm performed better than an item-based one as *N* was increased from 10 to 30, even though after that the result became worse than item-based one. That is, when a relatively small number of items is recommended, the proposed method causes a more proper item to be at a higher rank in the returned item set, $TopN_u$, and so our algorithm can provide better items for a target user than the other algorithms.

5 Conclusions and Future Works

As a part of Web 2.0, collaborative tagging is getting popular as an important tool to classify dynamic content for searching and sharing. We analyzed the potential of collaborative tagging systems, including personalized and biased user preference analysis, and specific and dynamic classification of content for applying to recommendations. Also proposed is a novel recommendation algorithm based on CTS selected from collaborative tags of users using a CF scheme.

As described in our experimental results, the proposed algorithm obtained better recommendation quality compared to a traditional user-based CF algorithm. Moreover, we also observed that our method can provide more suitable items for user preference even though the number of recommended items is small.

However, the empirical result showed that "noise" tags which have bad influence on analyzing user preference can be included in *CTS*. Such tags, due to the characteristics of tag, personalized and content-criticizable (e.g., *bad*, *my work* and *to read*), should be treated effectively for more valuable and personalized analysises. In addition, there remain common issues that have been mentioned in keyword-based analysis; polysemy, synonymy and basic level variation [1]. Semantic tagging is one of the interesting issues that we plan to consider for addressing these problems in the future.

References

1. Golder, S.A., Huberman, B.A.: Usage Patterns of Collaborative Tagging Systems. Journal of Information Science 32(2), 198–208 (2006)
2. Marlow, C., Naaman, M., Boyd, D., Davis, M.: HT06, Tagging Paper, Taxonomy, Flickr, Academic Article, To Read. In: Procs. of the17th Conf. on Hypertext and Hypermedia, pp. 31–40 (2006)
3. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based Collaborative Filtering Recommendation Algorithms. In: Procs. of the 10^{th} Int. Conf. on WWW, pp. 285–295 (2001)
4. Mathes, A.: Folksonomies-Coooperative Classification and Communication though Shared Metadata, http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html
5. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In: Procs of ACM 1994 Conf. on Computer Supported Cooperative Work, pp. 175–186 (1994)
6. Herlocker, J.L, Konstan, J.A., Riedl, J.: Explaining Collaborative Filtering Recommendations. In: Procs of ACM 2000 Conf. on Computer Supported Cooperative Work, pp. 241–250 (2000)
7. Deshpande, M., Karypis, G.: Item-based Top-N Recommendation Algorithms. ACM Transactions on Information Systems (TOIS) 22(1), 143–177 (2004)
8. Han, J., Kamber, M.: Data Mining Concepts and Techniques, 2nd edn. Morgan Kaufmann Publishers, San Francisco (2006)
9. Sahami, M., Dumais, S., Heckerman, D., Horvitz, E.: A Bayesian Approach to Filtering Junk E-Mail. In: Learning for Text Categorization: Papers from the 1998 Workshop (1998)
10. Muller, M.J., Ehrlich, K., Farrell, S.: Social Tagging and Self-Tagging for Impression Management. In: CSCW 2006. Procs of the ACM Conf. on Computer Supported Cooperative Work (2006)
11. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Analysis of Recommendation Algorithms for E-Commerce. In: Procs of the 2^{nd} ACM Conf. on E-Commerce, pp. 158–167 (2000)
12. Voss, J.: Collaborative Thesaurus Tagging the Wikipedia Way (2006), http://arxiv.org/abs/cs/0-604036
13. Miller, B.N., Konstan, J.A., Riedl, J.: PocketLens: Toward a Personal Recommender System. ACM Transactions on Information Systems (TOIS) 22(3), 437–476 (2004)
14. Wal, T.V.: Explaining and Showing Broad and Narrow Folksonomies (2005), http://www.perso-nalinfocloud.com/2005/02/explaining_and_.html
15. Kelkar, S., Ajita, J., Doree, S.: An Activity-based Perspective of Collaborative Tagging. In: ICWSM 2007. Procs of the Int. Conf. on Weblogs and Social Media (2007)