Self-Optimizing a Clustering-based Tag Recommender for Social Bookmarking Systems

Malik Tahir Hassan, Asim Karim, Fahad Javed, and Naveed Arshad
Department of Computer Science
LUMS School of Science and Engineering
Lahore, Pakistan
{*mhassan,akarim,fahadjaved,naveedarshad*}*@lums.edu.pk*

***Abstract*—In this paper, we propose and evaluate a self-optimization strategy for a clustering-based tag recommendation system. For tag recommendation, we use an efficient discriminative clustering approach. To develop our self-optimization strategy for this tag recommendation approach, we empirically investigate when and how to update the tag recommender with minimum human intervention. We present a nonlinear optimization model whose solution yields the clustering parameters that maximize the recommendation accuracy within an administrator specified time window. Evaluation on "BibSonomy" data produces promising results. For example, by using our self-optimization strategy a 6% increase in average F1 score is achieved when the administrator allows *up to* 2% drop in average F1 score in the last one thousand recommendations.**

I. Introduction

Social bookmarking systems have become popular in recent years for organizing and sharing resources on the Web. Such systems allow users to build a database of resources, typically Web pages and publications, by adding basic information (like URLs and titles) about them and by assigning one or more keywords or tags describing them. The tags serve to organize the resources and help improve recall in searches. Individual users' databases are shared among all users of the system enabling the development of an information repository which is commonly referred to as a folksonomy. Tag recommendation for new posts by users is desirable for two reasons. First, it ensures uniformity of tagging enabling better searches, and second, it eases the task of users in selecting the most descriptive keywords for tagging the resource.

Automatic tag recommendation systems are typically built once and then used for a long period of time. However, the recommendation performance of such systems degrade with time as the social environment evolves but the tag recommendation system does not. It is desirable for a tag recommendation system to be self-optimizing whereby it remains updated with fresh knowledge and is capable of accurate recommendations over time. Based on monitoring of recommendation performance, and given the administrator's specification of average update time, the system should adapt automatically using optimal parameters.

In this paper, we present and evaluate a self-optimizing strategy for a clustering based tag recommendation system for social bookmarking applications. We adopt the discriminative clustering based tag recommender presented by [1]. In this approach, the historical data of posted resources is clustered and a ranked list of discriminating tags for each cluster is developed. Given a new posting, based on its contents, the approach recommends the top 5 tags from the cluster that is most relevant to the post.

Our self-optimization strategy is empirical in nature. We analyze the performance of our tag recommendation system under different parameter settings. We observe that the performance of the system degrades with time, which helps us to decide *when* to update the system. We also develop a nonlinear optimization model for selecting the optimal parameters for maximum recommendation accuracy given constraint on the update time. The relationships in the optimization model are determined empirically by curve fitting. The solution of the optimization model tells us *how* best to update the recommendation system given administrator constraint on time. Our self-optimizing tag recommendation system is evaluated on real social bookmarking system data of Bibsonomy [2] provided by ECML PKDD Discovery Challenge 2009 [3]. Our experiments demonstrate that the self-optimizing strategy can improve (and in general maintain) the performance of the tag recommendation system with minimal intervention from the administrator.

II. Related Work

Tagging resources with one or more words or terms is a common way of organizing, sharing, and indexing information. Tagging has been popularized by Web applications like image (e.g. flickr), video (e.g. YouTube), bookmark (e.g. del.icio.us), and publication (e.g. BibSonomy) sharing/organizing systems. Automatic tag recommendation for these applications can improve the organization of the information through 'purposeful' tag recommendations. Moreover, automatic tag recommendations ease the task of users while posting new resources.

In recent years, several methods have been proposed for content-based tag recommendation in social bookmarking

systems. Lipczak's method extracts the terms in the title of a post, expands this set of terms by using a tag co-occurrence database, and then filters the result by the poster's tagging history [4]. He reports significant improvements in performance after each step of this three step process. In [5], Lipczak et. al. use resource IDs, resource contents, and user profiles to recommend tags. Symeonidis et al. [6] present a framework for tag recommendation based on semantic analysis. They represent the three entities in a social network (users, items, and tags) by 3-order tensors and apply Higher Order Singular Value Decomposition (HOSVD) to obtain a compact tagging model.

Document clustering has been used extensively for organizing and summarizing large document collections [7], [8]. A useful characteristic of clustering is that it can handle sparse document spaces by identifying cohesive groups. However, clustering is generally computationally expensive. Recently, Hassan et al. [1] present an efficient discriminative clustering based approach for tag recommendation. They recommend tags by combining the ranked lists obtained from content-based clustering, tag-based clustering, and user profiling. In this work, we use a simpler version of this approach using tag-based clustering only. The primary motivation of this work is optimization of a tag recommender rather than developing a new tag recommendation system.

Self-optimization is desirable in large scale systems and has been used with success in communication systems. In the domain of social networks, [9] and [10] present pheromone evaporation technique of ant colony and swarm intelligence for personalization in tagging systems. On the other hand, in this work, we propose a self-optimization strategy for tag recommendation using nonlinear programming to obtain optimal parameters.

III. Self-Optimizing a Clustering Based Approach for Tag Recommendation

We present our self-optimizing tag recommendation system by first describing the discriminative clustering method and its use for tag recommendation, followed by a discussion of our self-optimization strategy for this tag recommendation system.

A. Discriminative Clustering

In this work, we employ a discriminative clustering based tag recommendation system. A discriminative clustering method is used to cluster the historical data of posts based on the posts' tags. This method maximizes the sum of the discrimination information provided by posts and outputs a weighted list of discriminative tags for each cluster. Given a new post, and based on the post's contents, the top 5 tags of the most relevant cluster for the post are recommended. This is a simpler version of the discriminative clustering based tag recommendation system presented in [1]. We use this simpler version to highlight self-optimization in tag recommendation systems, although our self-optimization strategy can be extended to the original version as well.

Let $\{\mathbf{x}_i\}_{i=1}^N$ be the set of historical posts, where N is the total number of posts and the ith post is defined by the vector

$$\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{iT}]$$

where T is the size of the vocabulary of tags. Each $x_{ij} \geq 0$ is the weight of tag j in post i. A new post l is also defined in the same vector space model except that each $x_{lj} \geq 0$ is the weight of tag j occurring in the contents of the post. Let $K \ll N$ be the number of clusters desired.

Given the above definitions, the discriminative clustering method proceeds as follows. After doing a random initial clustering of the data, a discriminative tag weight w_j^k is computed for each tag j in the vocabulary and for each cluster k as [11]

$$w_j^k = \begin{cases} p(x_j|k)/p(x_j|\neg k) & \text{when } p(x_j|k) > p(x_j|\neg k) \\ p(x_j|\neg k)/p(x_j|k) & \text{otherwise} \end{cases}$$

where $p(x_j|k)$ and $p(x_j|\neg k)$ are the probabilities that tag j belongs to cluster k and the remaining clusters ($\neg k$), respectively. The discriminative tag weight quantifies the discrimination information that tag j provides for cluster k over the remaining clusters. They are also used to rank the most discriminating tags for each cluster.

Having computed the discriminative tag weights for the current clustering, two discrimination scores can be computed for each post i. One score, denoted as $Score^k(\mathbf{x}_i)$, expresses the discrimination information provided by post i for cluster k, whereas the other score, denoted as $Score^{\neg k}(\mathbf{x}_i)$, expresses the discrimination information provided by post i for clusters $\neg k$. These scores are computed by linearly pooling the discrimination information provided by each tag x_j in post i as [11]

$$Score^k(\mathbf{x}_i) = \frac{\sum_{j \in Z^k} x_j w_j^k}{\sum_j x_j} \text{ and}$$

$$Score^{\neg k}(\mathbf{x}_i) = \frac{\sum_{j \in Z^{\neg k}} x_j w_j^k}{\sum_j x_j}$$

In these equations, $Z^k = \{j|p(x_j|k) > p(x_j|\neg k)\}$ and $Z^{\neg k} = \{j|p(x_j|\neg k) > p(x_j|k)\}$ are sets of tag indices that vouch for clusters k and $\neg k$, respectively. Each post, described by its tags $\mathbf{x}$, is then reassigned to the cluster k for which the cluster score $f^k = Score^k(\mathbf{x}) - Score^{\neg k}(\mathbf{x})$ is maximum. This is the cluster that makes each post most discriminating among all the clusters.

The overall clustering objective is to maximize the sum of discrimination information, or cluster scores, of all posts. Mathematically, this is written as

$$\text{Maximize } J = \sum_{i=1}^{N} \sum_{k=1}^{K} I^k(\mathbf{x}_i) \cdot f^k$$

where $I^k(\mathbf{x}_i) = 1$ if post i is assigned to cluster k and zero otherwise. Iterative reassignment is continued until the change in the clustering objective becomes less than a specified small value. Typically, the method converges satisfactorily in fewer than 15 iterations.

Given a new post $\mathbf{x}$, the top 5 tags of the kth cluster are recommended where k is such that $f^k(\mathbf{x})$, is a maximum.

1) Self-Optimization Strategy: The discriminative clustering based tag recommendation system described above will have to be updated from time to time to maintain its recommendation accuracy. This is because of changes in posting and tagging behaviors and additions to the tag vocabulary. Two questions arises while designing a self-optimizing system: (1) when should the recommendation system be updated? (2) how should it be updated?

The answer to the first question is easy. The recommendation system should be updated when its recommendation accuracy drops by more than a specified amount. The specified drop in accuracy can also be related to the number of recommendations after which an update is required, as demonstrated later in our experiments. In our context, update means re-building the clustering model again. The recommendation system can also be updated after a specified time interval. The administrator needs to specify the thresholds for accuracy drop and/or time interval.

The answer to the second question of how should the recommendation system be updated is more involved. First, we note that re-building the clustering model takes time. The computational complexity of the discriminative clustering method is $O(NKI)$, where N is the number of posts, K is the number of clusters, and I is the number of iterations. The clustering method converges satisfactorily in fewer than 15 iterations (see [1]) thus removing I from being a variable. This leaves N and K as the two key variables defining the re-building time. Second, as demonstrated in our experiments later, the recommendation accuracy of our discriminative clustering based approach depends nonlinearly upon both N and K.

Given the above observations, we define an optimization problem that is solved to determine the best values of N and K for re-building the cluster model. The optimization problem can be described qualitatively as

Maximize: *Accuracy*

subject to: *Time* $<$ t

where t is an administrator specified constraint on clustering time. This is in general a nonlinear optimization problem with both accuracy and time dependent on N and K.

We quantify the optimization problem empirically by learning the relationship of accuracy and time with their dependent variables (N and K). Once this is done, the optimization problem is solved to find the optimal values for N and K given the time constraint that maximizes the recommendation accuracy. The time constraint t is another parameter that is specified by the administrator.

IV. Experimental Setup

A. Data and their Characteristics

We evaluate our approach on data made available by the ECML PKDD Discovery Challenge 2009 [3]. The data are obtained from dumps of public bookmark and publication posts on BibSonomy [2]. The dumps are cleaned by removing spammers' posts and posts from the user dblp (a mirror of the DBLP Computer Science Bibliography), and by various text normalizations.

The post-core at level 2 data are obtained from the cleaned dump (until 31 December 2008) and contain all posts whose user, resource, and tags appear in at least one more post in the post-core data. The post-core at level 2 contain 64,120 posts (41,268 bookmarks and 22,852 publications), 1,185 distinct users, and 13,276 distinct tags.

For this work, we use the *tas* table of the post-core at level 2. This table contains the tag assignment (who attached which tag to which resource). Key fields of *tas* include: user ID, tag, content ID, and date. We process the posts in the order in which they are posted.

B. Evaluation Criteria

The performance of tag recommendation systems is typically evaluated using precision, recall, and F1 score (or $FScore$), where the F1 score is a single value (harmonic mean) obtained by combining both precision and recall. We report the precision, recall, and F1 score averaged over all the posts in the testing set.

V. Experimental Results

In this section, we present an analysis of our self-optimizing discriminative clustering based tag recommendation system. We present results demonstrating the effectiveness of discriminative clustering based tag recommendation. We evaluate its characteristics and discuss the implementation and evaluation of our self-optimization strategy.

A. Discriminative Clustering based Tag Recommendation

The performance of the discriminative clustering method is evaluated on the post-core at level 2 data. We cluster the posts in the training set based on the tags assigned to them. After clustering and ranking of tags for each cluster, we recommend the top 5 tags from the ranked list of the assigned cluster for each post in the test set. We report the average precision, recall, and FScore (used interchangeably with accuracy) values over the test set for the recommendation system. Unless stated otherwise, the training set contains the first 30,000 posts and the test set contains the remaining 34,120 posts.

Table I shows the top ranked tags for selected clusters. It is seen that the discriminative clustering method is capable of grouping posts and identifying descriptive tags for each group of posts. Noisy tags are not ranked high in the lists. The recommendation performance of the discriminative

Table I
TOP TAGS FOR SELECTED CLUSTERS (K = 200)[1]

No.	Top Discriminating Tags
1	svm, ki2007webmining, mining, kernels, textmining, dm, textclassification
2	windows, freeware, utility, download, utilities, win, shareware
3	fun, flash, games, game, microfiction, flashfiction, sudden
4	tag, cloud, tagcloud, tags, folksonomia, tagging, vortragmnchen2008
5	library, books, archive, bibliothek, catalog, digital, opac
6	voip, mobile, skype, phone, im, messaging, hones
7	rss, feeds, aggregator, feed, atom, syndication, opml
8	bookmarks, bookmark, tags, bookmarking, delicious, diigo, socialbookmarking

clustering approach is discussed in the subsequent sections. This discussion leads to the formulation and evaluation of the self-optimization strategy for the approach.

B. Relationship of N and K with Time and FScore

We study the recommendation performance of the discriminative clustering approach by varying the number of posts N in the training set and the number of clusters K. We find an almost linear relationship between clustering time and N, and clustering time and K. However, the dependence of clustering time on both N and K together is nonlinear. This is seen from the 3D surface plot of clustering time with N and K (Figure 1). We also verify the nonlinearity of this relationship by fitting first, second, and third degree polynomials to the data, finding that first and second degree polynomials produce large sum of squared errors (SSE) as compared to that produced by the third degree polynomial.

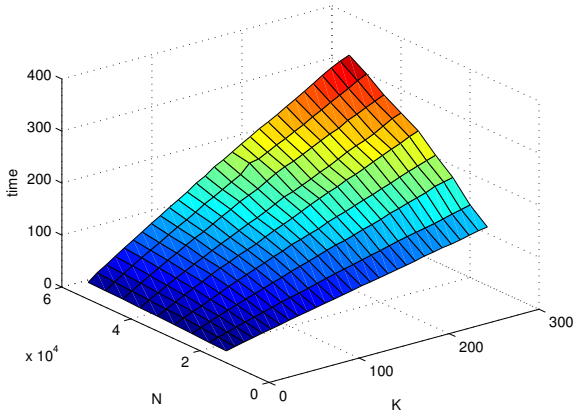

Figure 1. Clustering time versus both N and K

Nonlinear relationships exist between N and FScore and between K and FScore. More importantly, these nonlinear relationships are not always monotonic. Similarly, the relationship of FScore with both N and K is nonlinear in nature as demonstrated by the 3D surface plot in Figure 2. For this relationship also, we verify its nonlinearity by fitting first, second, and third degree polynomials, finding that a third degree polynomial fit produces a lower SSE.

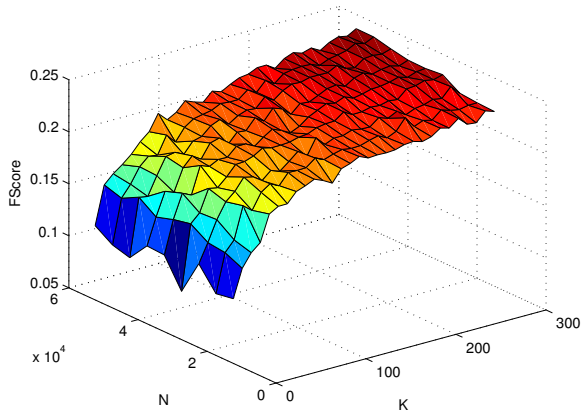

Figure 2. Fscore versus both N and K

C. Performance Variation With Number of Test Posts

After the tag recommendation system is built from training data it is used to suggest tags for new posts. The performance of the system on new posts is bound to change with time as posting behaviors change. We investigate this aspect of the system by evaluating its performance on test posts ranging in quantity from 1,000 to 34,000 (Figure 3). The variation in performance is clear from this figure. More interestingly, it is seen that FScore values drop sharply after reaching a peak at about 6,000 test posts. Analyzing the data we find the reason for this drop in performance: there are a couple of new users actively posting using a single tag (in German language). This observation highlights the need to monitor performance and to update the system when performance drops below a specified threshold.

D. When and How to Update the Recommendation System

Updating our tag recommendation system involves rebuilding the clustering model by running the discriminative clustering method. This will allow the recommendation system to adapt to changes and drifts in the data, and ensure continued high performance. Two questions arise in this respect: (1) when should we update our recommendation system? (2) how should we update it, i.e., what should be

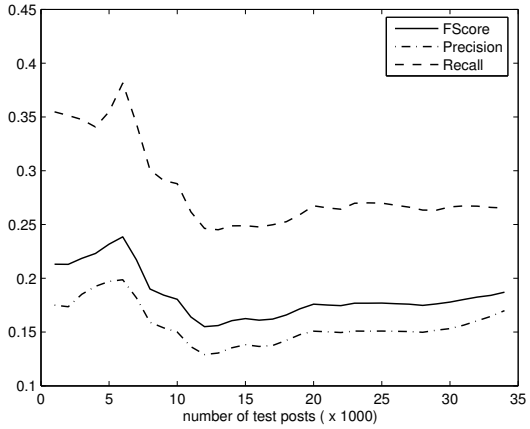

Figure 3. Performance variation with number of test posts (best performance at about 6,000)

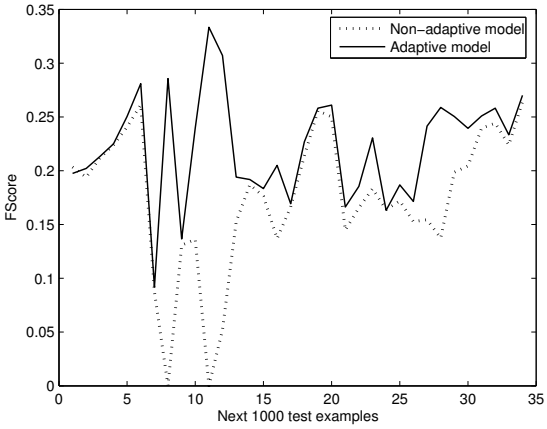

Figure 4. Comparison between original and re-clustering based recommendation systems

the values of the parameters of clustering (number of training posts (N) and number of clusters (K))?

Figure 3 helps us in answering the *when* part of the question. It is seen that the performance varies with time as new posts are seen. Thus, when the performance monitor reports an average FScore drop greater than the specified threshold, re-clustering is required. More specifically, the drop in performance should be monitored in the past b recommendations. For instance, if in the last 1,000 recommendations the FScore drops by more than 2%, then the recommendation system needs to be updated by re-building the clustering model. Figure 4 shows the performance of the recommendation system over the *next* 1,000 posts when the system is updated (using fixed non-optimal re-clustering) and when it is not. An update is only done when the performance drops by more than 2% in the last 1,000 posts. Figure 4 shows that the average FScore of the system that is updated is about 5% higher than that of the system that is not updated. Non-optimal re-clustering uses all available posts seen so far, and K is set to 200.

The second question of *how* to update the recommendation system requires deciding the parameters of clustering. That is, we have to choose the values of N and K. Figure 2 has demonstrated that the values of N and K can significantly impact the performance of the recommendation system. We need to maximize the performance by re-clustering, given limited time. The optimal values of N and K for re-clustering are found by solving a nonlinear optimization problem, as discussed in Section V-E.

E. Nonlinear Optimization Model

The relationship of clustering time and FScore with both N and K is shown in Figures 1 and 2, respectively. These relationships are modeled by a third degree polynomial. After determining the third degree polynomials for FScore and time, we formulate the optimization problem as a nonlinear constrained integer programming problem. For presentation convenience in the following expressions N is the number of training posts in *thousands* and FScore is given as a percentage.

Maximize $FScore =$
$3.26 \times 10^{-5}N^3 + 1.35 \times 10^{-6}K^3 + 1.84 \times 10^{-7}NK^2 + 4.43 \times 10^{-6}N^2K - 0.0049384N^2 - 0.00079702K^2 - 0.00023509NK + 0.22396N + 0.15754K + 8.4041$
such that
$0.00036217N^3 + 9.20 \times 10^{-7}K^3 - 8.03 \times 10^{-6}NK^2 - 0.00011386N^2K - 0.042938N^2 - 0.00041314K^2 + 0.027492NK + 1.6426N + 0.047557K - 12.314 <$ t **and**
$1 \leq N \leq N_m$
and $1 \leq K \leq K_m$
and N and K are integers.

In the above formulation, t is the maximum available time allowed for re-clustering, N_m is the maximum number of training posts in thousands, and K_m is the maximum number of possible clusters (which is set to 300 in our experiments).

The nonlinear curve fitting and optimization problems are solved efficiently by using the MATLAB software. As an example, setting t to be 200, N_m to be 55 (i.e. 55,000 training posts), and K_m to be 300, the optimal solution returned by MATLAB is $N = 55$ and $K = 175$.

F. Performance of Self-Optimizing Tag Recommendation System

We evaluate the performance of our self-optimizing tag recommendation system under several settings demonstrating its benefit. Figure 5 shows the average FScore of the system under five settings: (1) ORG: *Original* system without re-clustering, (2) FRC: *Fixed re-clustering* (whenever

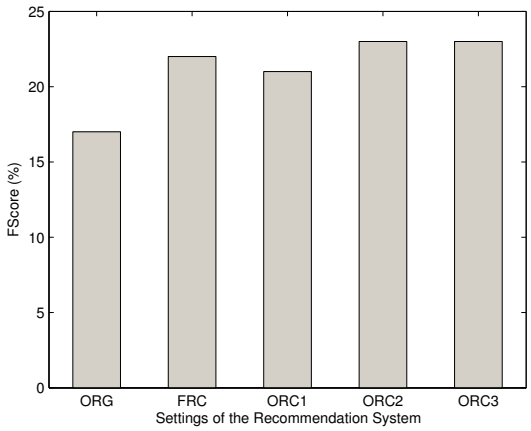

Figure 5. Performance of original, re-clustering, and self-optimizing tag recommendation systems

FScore drops by more than 2% in the last 1,000 posts) using all seen posts, $K = 200$, and t is unbounded, (3) ORC1: *Optimum re-clustering 1* as in (2) but with optimal values for N and K, and $t \leq 120$ seconds, (4) ORC2: *Optimum re-clustering 2* as in (3) but $t \leq 300$ seconds, and (5) ORC3: *Optimum re-clustering 3* as in (3) but t is unbounded.

These results demonstrate the benefit of our self-optimization strategy. The self-optimizing settings ORC2 and ORC3 significantly outperform the fixed re-clustering (FRC) and no update (ORG) settings. Their FScore value of 23% is 6% higher than that obtained by the original setting and 1% higher than that obtained by the fixed re-clustering setting. It is seen that for this relatively small data the clustering time does not play a significant role in the results.

VI. Conclusion and Future Work

In this paper, we propose and evaluate a self-optimizing discriminative clustering approach for tag recommendation in social bookmarking systems. The motivation behind making tag recommendation systems self-optimizing is to ensure continued high recommendation performance with minimal administrator input. We use an efficient discriminative clustering based approach for tag recommendation. This recommendation system is then made self-optimizing by developing an optimization formulation that seeks to maximize the recommendation accuracy by finding the optimal values for number of training posts and number of clusters to use. To develop this formulation, we study and analyze the performance of the recommendation system on real-world data. In particular, we study the effect of changing clustering parameters on clustering time and recommendation accuracy. The relationships between clustering time and recommendation accuracy with the clustering parameters are determined empirically. We find that our self-optimization strategy can increase recommendation accuracy significantly with minimal administrator input.

In the future, we would like to modify our clustering method and use incremental clustering to reduce the time required for adaptation. In addition, we plan to develop a window based model where less useful information will play a smaller role in recommendation.

References

[1] M. Hassan, A. Karim, S. Manandhar, and J. Cussens, "Discriminative clustering for content-based tag recommendation in social bookmarking systems," in *Proceedings of ECML PKDD Discovery Challenge Workshop*, 2009.

[2] BibSonomy, "Bibsonomy: A blue social bookmark and publication sharing system," http://www.bibsonomy.org/, 2009.

[3] F. Eisterlehner, A. Hotho, and R. Jäschke, "Ecml pkdd discovery challenge 2009," http://www.kde.cs.uni-kassel.de/ws/dc09, 2009.

[4] M. Lipczak, "Tag recommendation for folksonomies oriented towards individual users," in *Proceedings of ECML PKDD Discovery Challenge 2008*, 2008, pp. 84–95.

[5] M. Lipczak, Y. Hu, Y. Kollet, and E. Milios, "Tag sources for recommendation in collaborative tagging systems," in *Proceedings of ECML PKDD Discovery Challenge Workshop*, 2009.

[6] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos, "A unified framework for providing recommendations in social tagging systems based on ternary semantic analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 3, pp. 179–192, 2009.

[7] N. O. Andrews and E. A. Fox, "Recent developments in document clustering," Computer Science, Virginia Tech, Tech. Rep., 2007. [Online]. Available: http://eprints.cs.vt.edu/archive/00001000/

[8] J. Kogan, C. Nicholas, and M. Teboulle, *Grouping Multidimensional Data*. Springer, 2006, ch. A Survey of Clustering Data Mining Techniques, pp. 25–71.

[9] E. Michlmayr and S. Cayzer, "Learning user profiles from tagging data and leveraging them for personal (ized) information access," in *Proceedings of the Workshop on Tagging and Metadata for Social Information Organization*, 2007.

[10] R. Sharma and P. Bedi, "Personalized tag recommendations to enhance user's perception," in *Proceedings of International Conference on Advances in Recent Technologies in Communication and Computing*, 2009, pp. 944–947.

[11] K. Junejo and A. Karim, "A robust discriminative term weighting based linear discriminant method for text classification," in *Proceedings of Eighth IEEE International Conference on Data Mining (ICDM)*, 2008, pp. 323–332.