# Tripartite Hidden Topic Models for Personalised Tag Suggestion

Morgan Harvey[1], Mark Baillie[1], Ian Ruthven[1], and Mark Carman[2]

[1] University of Strathclyde, CIS Department, Glasgow, UK
{morgan.mb,ir}@cis.strath.ac.uk
[2] University of Lugano, Faculty of Informatics, Lugano, Switzerland
mark.carman@usi.ch

**Abstract.** Social tagging systems provide methods for users to categorise resources using their own choice of keywords (or "tags") without being bound to a restrictive set of predefined terms. Such systems typically provide simple tag recommendations to increase the number of tags assigned to resources. In this paper we extend the latent Dirichlet allocation topic model to include user data and use the estimated probability distributions in order to provide personalised tag suggestions to users. We describe the resulting tripartite topic model in detail and show how it can be utilised to make personalised tag suggestions. Then, using data from a large-scale, real life tagging system, test our system against several baseline methods. Our experiments show a statistically significant increase in performance of our model over all key metrics, indicating that the model could be successfully used to provide further social tagging tools such as resource suggestion and collaborative filtering.

## 1 Introduction

Social tagging systems provide a new way for Internet users to organise and share their own digital content and content from other users. Users are able to annotate each resource with any number of free-form tags of their own choosing without having to adhere to an *a-priori* set of keywords. The result of which is a personalised categorisation system defined by its users that can assist in locating resources in the future. This freedom to categorise resources in any way a user chooses is seen as an important advantage for such systems, tags become more personally meaningful and the initial categorisation process is made easier.

Unfortunately this ease of use and freedom of word choice comes at a significant cost. If each user is free to choose whatever tags she wishes then it is unlikely that other users will choose exactly the same tags to describe the same resource or indeed to tag similar resources they have found. Many studies have shown that obtaining high consistency among different taggers is very difficult to achieve and can be affected by many factors including vocabulary use, personal understanding of the resource and language [14, 6]. These factors result in the categorisation scheme displaying a number of highly undesirable characteristics such as polysemous and synonymous terms which make searching or browsing

through the collection difficult and inaccurate.

This lack of a consistent and shared vocabulary also results in a large number of unique or "singleton" tags appearing in the folksonomy. Sigurbjörnsson and van Zwol investigated [10] the characteristics of a large sample of the Flickr database (which can be taken as a good reference point for most large-scale tagging systems) and found that the tag frequency closely follows a Zipfian distribution. This is where a small number of tags are used very frequently with tag use quickly tailing off leaving the so called "long tail" of infrequently used tags. Generally speaking, the tags at the extreme ends of the distribution are not particularly useful; the high-frequency tags are too generic and the singleton tags tend to be either compound phrases or misspellings and are likely to only be useful in very specific cases. The distribution of tags per resource was also found to follow a power law with a small number of resources being very thoroughly annotated and a large majority (64%) having only 1, 2 or 3 tags.

To assist the user when tagging new resources, most of these systems offer some form of tag recommendation to increase the chance that a given resource is tagged and also to increase the average number of tags assigned to each resource in the system. Despite their clear utility in improving social tagging system the literature on - particularly personalised - tag recommendation is still quite sparse. Existing approaches tend be based on a mixture of the most popular tags and tags which the user has used previously. Recently more sophisticated systems have been proposed, focussing on methods derived from collaborative filtering and simple co-occurence data or making use of information other than the tags provided by users (for example the HTML content of web pages) [9, 2].

In this paper we model the complete tripartite structure of a folksonomy by extending the latent Dirichlet allocation topic model and use this to provide personalised tag suggestions. Previous work by Wu et. al. [13] modelled broad folksonomic data using a simple Separable Mixture Model representation which reportedly worked well, however it makes the assumption that the probabilities of a user, a tag and a resource are all independent given a dimension $d_\alpha$. It is also not an entirely generative model and does not make use of a Bayesian hierarchical structure when inferring parameters, meaning that it could easily suffer from problems of over-fitting. A similar model proposed by Plangprasopchok and Lerman [8] was used to recommend resources to tagging system users. Our model improves on these by taking a fully Bayesian view of the problem at hand, thus providing a more statistically principled and scalable solution.

We go on to propose how the model can be used to improve on the performance of existing tag suggestion algorithms and describe appropriate experiments to test our hypothesis. We evaluate the precision and success of our models based purely on actual data from a live system, rather than via a user study. We then present and describe in detail the results from the experiments and discuss various advantages of our complete tagging model approach over function-specific algorithms such as those previously proposed. Finally we conclude with discussion of the results and explore directions for future work based on this model.

## 2 Hidden Topic Models and Modelling Folksonomies

Topic models attempt to probabilistically uncover the underlying semantic structure of a collection of resources based on analysis of only the vocabulary words present in each resource, this latent structure is modelled over a number of topics which are assumed to be present in the collection.

### 2.1 Latent Dirichlet Allocation

Latent Dirichlet allocation (LDA) [1] is a generative topic model which has attracted a lot of interest from both the machine learning and language processing community. LDA represents documents as random mixtures over latent topics which are random mixtures over observed words in the vocabulary. The model possesses a number of advantageous attributes; it is fully generative meaning that it is easy to make inferences on new documents or terms and overcomes the overfitting problem present in models such as Probabilistic Latent Semantic Indexing (pLSI) [5]. Also since in LDA each document is a mixture over latent topics it is far more flexible than models that assume each document is only drawn from a single topic.

In LDA each individual word token $w_n$ in the corpus $\mathbf{w}$ is assumed to have been derived from a single latent topic $z_n$, drawn from a distribution over topics for its parent document $d_n$. The probability of a word $w$ in the vocabulary given a topic $t$ is denoted by $\phi_{w|t} = P(w_n = w|z_n = t)$ and the probability of a topic given a document is denoted $\theta_{t|d} = P(z_n = t|d_n = d)$. Thus probability of a corpus $\mathbf{w}$ given (the matrices of) all term and topic probabilities $\Phi$ and $\Theta$ is:

$$P(\mathbf{w}|\Phi, \Theta) = \prod_{n=1}^{N} \sum_{t=1}^{T} \phi_{w_n|t} \theta_{t|d_n}$$

where $N$ is the length (in words) of the corpus $\mathbf{w}$ and $T$ is the number of latent topics. In order to make the model fully Bayesian, symmetric Dirichlet priors with hyperparameters $\alpha$ and $\beta$ are placed over the distributions $\theta_d$ and $\phi_t$.

Exact inference of the LDA model is intractable, however a number of methods of approximating the posterior distribution have been proposed including mean field variational inference [1] and Gibbs sampling [3]. Gibbs sampling is a Markov chain Monte Carlo method where a Markov chain is constructed that slowly converges to the target distribution of interest over a number of iterations. Each state of the Markov chain is (in this case) an assignment of a discrete topic (from 1 to $T$) to each $z_n$, i.e. to each observed word in the corpus. In Gibbs sampling the next state in the chain is reached by sampling all variables from their distribution when conditioned on the current values of all the other variables. Therefore for LDA each Gibbs sample is obtained by the following:

$$P(z_n = t|\mathbf{z}_{-n}, \mathbf{w}) \propto \frac{N_{-n,t}^{(w_n)} + \beta}{N_{-n,t}^{(.)} + V\beta} \frac{N_{-n,t}^{(d_n)} + \alpha}{N_{-n}^{(d_n)} + T\alpha}$$

where $\mathbf{z}_{-n}$ denotes the assignment of topics to all word positions (except the current topic $z_n$). $V$ is the vocabulary size. $N_{-n,t}^{(w_n)}$ is the number of times word $w_n$ is assigned to topic $t$ and $N_{-n,t}^{(.)}$ is the total number of words assigned to topic $t$ (both excluding $z_n$). $N_{-n,t}^{(d_n)}$ is the number of times topic $t$ occurs in document $d_n$ (excluding $z_n$) and $N_{-n}^{(d_n)}$ is the total number of words in document $d_n$ (less 1). After the sampling algorithm has been run over each word position in the corpus an appropriate number of times (i.e. until the chain has converged to a stationary distribution) we sample from the distribution to obtain estimates for our parameters $\Phi$ and $\Theta$ via the following equations:

$$\phi_{w|t} = \frac{N_t^{(w)} + \beta}{N_t^{(.)} + V\beta}$$

$$\theta_{t|d} = \frac{N_t^{(d)} + \alpha}{N^{(d)} + T\alpha}$$

The priors $\alpha$ and $\beta$ essentially act as a pseudo count indicating a relation to smoothing in language models. This allows the model to fall back on the priors in the event of sparse data. We can now use our estimated parameters $\Phi$ and $\Theta$ to compute a variety of useful distributions such as which documents are similar to each other, which words are similar to each other and by sampling over new data we can easily incorporate new documents into our model without having to re-run the entire algorithm.

This model is not, however, particularly suited to tagging data as it is missing some potentially useful information: the identity of the user who made each annotation. We therefore present a new model, influenced by LDA, which will include this useful information therefore improving the accuracy of tag predictions and allowing for other useful tasks such as determining which users are similar and providing personalised search.

### 2.2 Tripartite Topic Model (TTM)

Annotations in social tagging systems typically consist of 3 parts: the resource being tagged, the user who tagged the resource and the tag itself. In [7] this is modelled as a tripartite graph with 3 disjoint sets of nodes: resources $\mathcal{D} = \{\mathtt{d}_1, \ldots, \mathtt{d}_D\}$, users $\mathcal{U} = \{\mathtt{u}_1, \ldots, \mathtt{u}_U\}$ and tags $\mathcal{W} = \{\mathtt{w}_1, \ldots, \mathtt{w}_V\}$ with the edges between these nodes representing the individual annotations. Each assignment of a tag to a resource by a user is denoted as the relation $\mathcal{Y}$ and is typically called a tag assignment. Therefore the complete folksonomy is a quadruple $\mathcal{F} := (\mathcal{U}, \mathcal{W}, \mathcal{D}, \mathcal{Y})$. The resources are typically identifiers linking each unique resource id to a single web resource such as an image - as on Flickr - or a bookmark - as on social bookmarking sites such as del.icio.us.

In [13] it is noted that tags are usually semantically related if they are used to describe the same resources many times. Correspondingly, resources are similar if they are annotated with the same tags and users share similar interests if

their annotations share many related tags. These relationships can be mapped onto a conceptual space of $T$ dimensions (or in the topic modelling case, topics), that represent categories of knowledge, where each entity's component on a given dimension measures how similar it is to that category. This provides a framework for discovery of meaningful relationships between entities. In order to include information about which user was responsible for each annotation (word position in the corpus) we change the $\theta_{t|d}$ distribution to be the probability of a topic $t$ given a resource $d$ and a user $u \in \mathcal{U}$, denoted $\theta_{t|d,u}$. The matrix $\Theta$ becomes a tensor and the probability of the corpus $\mathbf{w}$ is simply:

$$P(\mathbf{w}|\Phi,\Theta) = \prod_{n=1}^{N} \sum_{t=1}^{T} \phi_{w_n|t}\theta_{t|d_n,u_n}$$

where $u_n$ is the user who submitted the tag at position $n$ in the corpus $\mathbf{w}$ and $d_n$ was the resource being tagged.

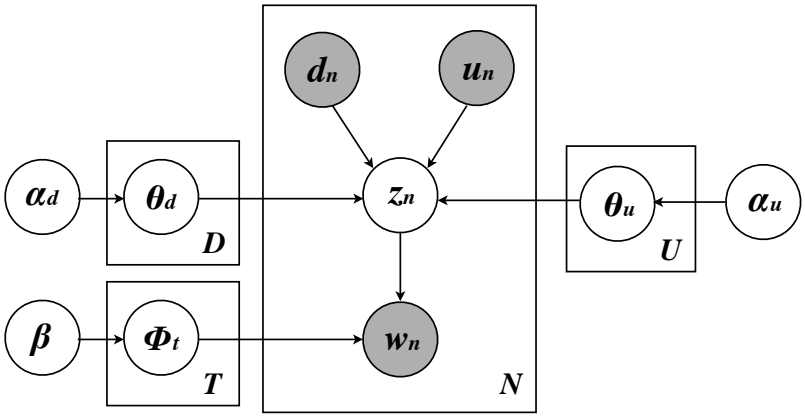


**Fig. 1.** Plate model graphical representation of Tripartite Topic Model. $T$ is the number of latent topics; $D$ the number of resources and $N$ is the number of word tokens and $U$ is the number of users.

The representation of users and resources over topics can be seen as a large, extremely sparse, 3D tensor $\in \mathbb{N}^{\mathcal{D}\times\mathcal{U}\times\mathcal{T}}$. Due to the size and sparsity of this tensor it would take an enormous amount of time to fully sample a conditional distribution and therefore we have two options. We can either sample from the distribution via a method such as Gibbs sampling or we can split the tensor $\Theta$ into two matrices $\Theta_d$ and $\Theta_u$ and make the Naive Bayes assumption that the distributions of documents and users are conditionally independent given the topic assignments $\mathbf{z}$. That is for each position in the corpus the probability of a topic given the resource the tag is assigned to is independent of the probability of the topic given the user who assigned the tag. Therefore the probability of a given topic assignment given $\theta_d$ and $\theta_u$ is:

$$p(z|\theta_d,\theta_u) = \frac{p(z)p(\theta_d,\theta_u|z)}{p(\theta_d,\theta_u)} = \frac{p(z)p(\theta_d|z)p(\theta_u|z)}{p(\theta_d,\theta_u)}$$
$$= \frac{p(z)[\frac{p(\theta_d)p(z|\theta_d)}{p(z)}][\frac{p(\theta_u)p(z|\theta_u)}{p(z)}]}{p(\theta_d,\theta_u)} \propto \frac{p(z|\theta_d)p(z|\theta_u)}{p(z)}$$

We place a Dirichlet distribution as a prior on $\theta_u$, so now we have hyper-parameters $\beta$, $\alpha_d$ and $\alpha_u$ on $\phi_t$, $\theta_d$ and $\theta_u$ respectively. Figure 1 shows a Bayesian network plate diagram of the complete model.[3] Now that we have an appropriate representation for the probability of a topic given a user and the probability of a topic given a resource a Gibbs sample can be obtained as follows:

$$P(z_n = t|\mathbf{z}_{-n},\mathbf{w}) \propto \frac{N_{-n,t}^{(w_n)} + \beta}{N_{-n,t}^{(.)} + V\beta}\left(\frac{N_{-n,t}^{(d_n)} + \alpha_d}{N_{-n}^{(d_n)} + T\alpha_d}\frac{N_{-n,t}^{(u_n)} + \alpha_u}{N_{-n}^{(u_n)} + T\alpha_u}\right)\left(\frac{N_{-n,t}}{N_{-n}}\right)^{-1}$$

Where for each tag annotation for each resource $u_n$ is the known user who annotated the resource with that tag. Estimates for the $\phi$ and $\theta_d$ matrices are the same as for LDA and the new $\theta_u$ matrix can be calculated from samples of the Gibbs sampling Markov chain in a similar manner. The resulting model of the complete folksonomy can then be used to uncover relationships between users, tags and resources and therefore make useful inferences about new data. In this case we use the model to offer users intelligent tag predictions based on their own prior tagging behaviour as well as the behaviour of the community.

### 2.3 Suggesting New Tags

Given some initial tags provided by the user for a given resource and the output from our topic model, we want to predict which tags the user will enter next and offer them as suggestions. To do this we can estimate a distribution over the latent topics for the pseudo-document $q$ comprising the tags supplied by the user. This can be calculated as a point estimate:

$$P(t|q) = \frac{N_t^{(q)} + P(t)\alpha_d}{N^{(q)} + \alpha_d}$$

To calculate a value for the topic distribution $P(t|q)$, we need to estimate a value for $N_t^{(q)}$, the count for the topic $t$ in the pseudo-document $q$. We can calculate the expected value for $N_t^{(q)}$ by summing over all terms in the query as follows:

$$E[N_t^{(q)}] = \sum_{w \in q} P(t|w)N_w^{(q)}$$

[3] We note that the conditional independence assumption between $d_n$ and $u_n$ given $z_n$ implies that one of the arrows joining the three variables should be reversed. We have, however, left both arrows pointing to $z_n$ in order to simplify the description of the generative process.

Where $P(t|w)$ can be calculated using the $\phi_t$ distribution from our model via Bayes' rule, and $N_w^{(q)}$ is the number of times tag $w$ appears in the query $q$. Now that we have a distribution for the pseudo-document over the latent topics we can estimate the probability of observing a new term:

$$P(w|q) = \sum_{t}^{T} P(w|t)P(t|q)$$

This gives us the probability of a term in the corpus given the pseudo-document, so if we calculate this for $\forall w \in \mathcal{W}$ and then order these by probability in descending order we can choose the top $n$ terms in this ranked list as tag suggestions. In our tripartite model we also want to include the user's personal preferences in these suggestions. Based on the matrix $\Theta_u$ from our model, the personalised distribution over terms can be calculated thus:

$$P(w|q,u) = \sum_{t}^{T} P(w|t)\frac{P(t|q)\theta_{t|u}}{P(t)}$$

Where $u$ is the user who generated the tags for the pseudo-document. This final distribution over terms indicates each term's probability given the previously observed terms (the terms of the psuedo-document) and the topical interests of the user. These probabilities can then be used as a multiplier on traditional tag suggestion methods (such as the one outlined in the description of baseline method 3 below) and provides a smoothed, personalised weighting for each term.

## 3 Experimental Set-up

In our experiment we compare our method and LDA with 3 "baseline" methods through empirical evaluation based on held-out data from a real-life data set obtained from a large online social tagging system. In this section we outline our experimental set-up in more detail, explain the various methods for tag suggestion and briefly describe the data set and the settings of parameters for our tripartite model.

### 3.1 Evaluation Method

In order to evaluate the accuracy of the tags suggested by our model we need some form of relevancy judgement, for example a list of all accurate and useful tags for each resource. One method for doing this which has been utilised previously is a user study where users are asked if they think that tags suggested for each resource are relevant or not. We have chosen not to use this method as we are interested in personalised results, therefore only the user(s) who originally tagged the resource can really say whether a tag is relevant or not. In this case a user study would likely provide an over-estimate of the quality of the results and therefore we choose to evaluate our system based only on the tags provided by

the user on the live system. Given a set of $l$ tags for a given resource we choose $m$ tags as input for our suggestion algorithm and use the remaining $l$-$m$ as the set of relevant suggestions. These resource are chosen from a set of held-out resources (i.e. resources that have not been used to train the model) and will give an estimate of the quality of the suggested tags which we believe will more accurately reflect the performance of a live system.

Since we are interested in the ability of our system to return a good ranked list of suggested items we use the following evaluation metrics:

**P@k - "precision at rank k"** the ratio of suggested tags that are relevant, averaged over test resources. We report P@k for k=1, k=5 and k=20.

**S@k - "success at rank k"** the ratio of times where there was at least 1 relevant tag in the first k returned. We report S@k for k=1, k=5 and k=20. S@1 and P@1 are the same and are therefore not reported separately.

**MRR = "mean reciprocal rank"** the multiplicative inverse of the rank of the first relevant suggested tag, averaged over test resources.

Note that we have chosen to evaluate the precision and success metrics for k values up to 20 as this is the number of tags usually suggested on social tagging web sites. k values of 1 and 5 are the most commonly reported in other literature and people tend to pay more attention to the first few results in a ranked list. When training the systems we hold out 20% of resources chosen at random and then "bin" these resources into two sets; one (hereafter referred to as set1) containing documents with between 4 and 8 annotations and the other (set2) with 9 or more annotations.

### 3.2 Baseline Systems

So that we compare the results of our algorithms to the algorithms already used in social tagging systems we run the above tests on 3 "baseline" methods, LDA as well as on our Tripartite Topic Model (TTM). The first 2 of these methods simulate the tags that would be suggested on sites such as Flickr and Delicious and the final baseline method represents a slightly more sophisticated algorithm that has been proposed in previous literature [10, 9].

**TopSys** the simplest set of suggestions; the top k tags in the system by frequency of use.

**TopUser** the most frequently used tags by the user who tagged the resource, if more than 1 user has tagged the resource the union of all users' tags is used.

**CoTag** tag co-occurence using asymmetric normalisation, as used in previous research to find like terms in a folksonomy [9]. Where $Sim(i,j) = \frac{|i \cap j|}{|i|}$

### 3.3 Data Set

We conducted our tests on data provided by Bibsonomy[1] - a social bookmark and publication sharing system and a good example of a large, broad folksonomy

[1] Knowledge and Data Engineering Group, University of Kassel: Benchmark Folksonomy Data from BibSonomy, version of June 30th, 2007. http://bibsonomy.org/

and as such is ideal for our research aims. The Bibsonomy data set shares similar characteristics with other large folksonomic data sets noted in previous research, most notably the tag use frequency follows a power law, as does the number of annotations per resource.

To filter out noise and to provide useful data for our evaluation methods we discard any resources that have less than 4 annotations and remove any tags that are used to annotate less than 5 resources. This results in a data set of 36167 resources from 992 users with a total vocabulary of 5116 terms, 28143 (77.8%) of the resources fit into set1, the remaining 8024 (22.2%) fit into set2. To select test data we use stratified random sampling resulting in a total of 7235 (20%) held-out resources with 5630 (79.4%) from set1 and 1605 (20.6%) from set2. In order to ensure that the results returned are not simply due to the held-out resources chosen we perform all tests over 10 different folds. The unfiltered data set displays similar characteristics to those of other folksonomies analysed in related literature; the mean number of tags per resource is 3.27 (median 2), 68.6% of all resources have less than 3 tags. Our filtered data therefore represents only 31.4% of the total Bibsonomy data set, highlighting the wide applicability of a good tag suggestion system.

### 3.4 Choosing $T$ and Determining Convergence

One very influential parameter that must be set in any latent topic model is the value of $T$; the number of latent topics in the model. While there has been some work published on algorithms which attempt to estimate this value automatically using Dirichlet Processes [12], it is generally acceptable to use empirical methods to determine the optimal value to use. In our analysis we run our tag suggestion algorithm and compare the precision and success values for each model, we are looking for vales of $T$ where the delta improvement in the precision is small as this is where the optimal value of $T$ lies. When run on the Bibsonomy data set we find a correlation between both metrics for the values over $T$, with both indicating that around 200 latent topics provides the most optimal fit for the data using our model. We therefore use this parameter setting for our experiments.

We use the Rao-Blackwellised Gibbs sampling method proposed by Styvers and Griffths [4, 3]. It is important when using methods such as Gibbs sampling to estimate a posterior distribution that the Markov chain is given enough time to "burn-in", i.e. when it begins to approach a stationary distribution [11]. To determine when our chain is beginning to approach a stationary distribution we can calculate the log likelihood of the model given the currently sampled estimate every $n$ iterations. If the chain is converging correctly we should find that these values should initially decrease quite rapidly then as the chain approaches convergence the change (delta) in log likelihoods should become smaller until the deltas become negligible. For all of our topic model estimations we discarded the first 500 iterations of the chain and then averaged over samples for every 25 iterations of the chain thereafter until reaching 2000 iterations.

## 4 Experimental Results

In this section we present the results from the series of experiments described in the previous section. First we look at overall performance of the 5 tag suggestion methods for a "typical" scenario of a user providing 2 tags for our methods to base their suggestions on, we highlight the difference in performance over the two resource "sets" and comment on how this is likely to relate to real performance. We then look at how varying the number of user tags provided affects the quality of tags suggested by our tripartite model.

### 4.1 Tag Suggestion Performance

The results of the tag suggestion tests using the 5 different methods on resources from set1 (sparsely annotated resources) are presented in Table 1. The results for TopSys over all metrics are extremely poor (as expected), the results for TopUser are slightly better but still well below those returned by the other, more sophisticated methods. Statistically significant improvements of the TTM method over both CoTag and basic LDA are observed for all metrics. These results show that our tripartite model is able to fit the available training data better than the other methods and therefore provides more useful and accurate suggestions. The larger improvements in precision and MRR indicate that the TTM method is suggesting fewer incorrect tags and is returning more relevant tags at a higher rank than the other methods.

**Table 1.** Results for sparsely annotated resources. P@20 has been excluded from this table as it is not relevant for resources from set1. Last column shows the percentage improvement of TTM over CoTag, * indicates a statistically significant result, 2-sample t at 95% confidence.

| | **TopSys** | **TopUser** | **CoTag** | **LDA** | **TTM** | **% change** |
|---|---|---|---|---|---|---|
| **S@1** | 0.0490 | 0.2269 | 0.3449 | 0.3197 | 0.3736 | +8.32* |
| **S@5** | 0.1540 | 0.4495 | 0.5648 | 0.5494 | 0.6270 | +11.01* |
| **P@5** | 0.0353 | 0.1329 | 0.1786 | 0.1705 | 0.2029 | +13.61* |
| **S@20** | 0.3552 | 0.6853 | 0.7637 | 0.7583 | 0.8238 | +7.87* |
| **MRR** | 0.1023 | 0.2718 | 0.3608 | 0.3574 | 0.4056 | +12.42* |

The results from resources from set2 (densely annotated resources), presented in Table 2, show that while the TTM still outperforms other methods over all metrics, the improvements are smaller. In this case the difference in performance between CoTag and TTM is statistically significant for all metrics except for S@20. In keeping with the results from set1 the greatest improvements are in precision and MRR, however all improvements over LDA and CoTag are smaller with the success metric being fairly similar for all 3 methods. This is likely because the small number of resources where the systems are unsuccessful are annotated with terms that have either not been used together before or do not

exist at all in the training set. In this case the scope for performance improvement over the CoTag method is very small.

**Table 2.** Results for densely annotated resources. Last column shows the percentage improvement of TTM over CoTag, * indicates a statistically significant result, 2-sample t at 95% confidence.

| | **TopSys** | **TopUser** | **CoTag** | **LDA** | **TTM** | **% change** |
|---|---|---|---|---|---|---|
| **S@1** | 0.1576 | 0.3499 | 0.6312 | 0.5879 | 0.6437 | +1.98* |
| **S@5** | 0.3829 | 0.5882 | 0.7811 | 0.7693 | 0.8132 | +4.11* |
| **P@5** | 0.1258 | 0.2436 | 0.4007 | 0.3796 | 0.4236 | +5.71* |
| **S@20** | 0.6593 | 0.8246 | 0.9376 | 0.9329 | 0.9516 | +1.49 |
| **P@20** | 0.0749 | 0.1391 | 0.2022 | 0.1972 | 0.2181 | +7.86* |
| **MRR** | 0.2244 | 0.2788 | 0.3857 | 0.3890 | 0.4125 | +6.95* |

### 4.2 Varying the Number of Input Tags

Selected results from CoTag and TTM are presented in Table 3 for varying numbers of input tags. They indicate that the performance of CoTag at 2 and 3 input tags is significantly better than with 1, however there is little difference between the performance with 2 or 3 tags. TTM performs well when only given a single input tag to infer suggestions from and its performance in terms of precision and MRR increases as the number of input tags increases. Success@k metrics are not significantly different over varying numbers of input tags.

**Table 3.** Results from densely annotated resources from fold 10 for varying number of input tags. Number of input tags in square brackets.

| | **CoTag[1]** | **CoTag[2]** | **CoTag[3]** | **TTM[1]** | **TTM[2]** | **TTM[3]** |
|---|---|---|---|---|---|---|
| **S@1** | 0.6058 | 0.6398 | 0.6186 | 0.6464 | 0.6594 | 0.6492 |
| **S@20** | 0.8986 | 0.9322 | 0.9388 | 0.9366 | 0.9522 | 0.9520 |
| **P@20** | 0.1936 | 0.2022 | 0.1948 | 0.2214 | 0.2245 | 0.2302 |
| **MRR** | 0.3494 | 0.4032 | 0.4061 | 0.3966 | 0.4172 | 0.4243 |

## 5 Conclusions and Future Work

In this paper we have proposed a new probabilistic latent topic model to deal with data from social tagging systems. The model allows us to estimate topic distributions over users and documents and term distributions over topics. The model is applied to data from the broad-folksonomy social tagging system Bibsonomy and is used to suggest new tags to users based on a small number of tags that they have entered as well as their past annotations. We have shown that this model suggests more relevant tags than current systems by comparing

these to held-out tags from annotated resources. In terms of precision, the use of our model improves upon the suggestions provided by the CoTag method on sparsely annotated resources by between 7.87 and 13.6%, improves upon basic LDA by 11.4 to 19.1% and vastly outperforms the more common TopSys and TopUser methods. The results are particularly promising for sparsely annotated resources which are extremely common in tagging systems, indicating that the tripartite model's suggestions would work well in a live system.

TTM provides a complete model of the data collected from a folksonomy and therefore could easily be utilised in future work for other useful estimations and is not merely suited to tag suggestion. For example the model could be used to find similar user groups by clustering based on values from the user-topic matrix. The tag suggestion algorithm could be adapted to suggest new resources rather than tags and therefore provide a form of personalised collaborative filtering over resources in the tagging system. Since topic models do not require explicit co-occurence between terms (tags) in order for them to share semantic similarity, our model could be utilised to improve searching in folksonomic systems, which at the current time are heavily restricted by the vocabulary problem. Further research into these possibilities is left as future work.

## References


[1] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, (3):993–1022, 2003.

[2] N. Garg and I. Weber. Personalized tag suggestion for flickr. In *WWW*, 2008.

[3] T. Griffiths and M. Steyvers. Finding scientific topics. *PNAS*, 2004.

[4] G. Heinrich. Parameter estimation for text analysis. Technical report, Fraunhofer IGD, 2008.

[5] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1/2):177–196, 2001.

[6] R.S. Hooper. Indexer consistency tests—origin, measurements, results and utilization. Technical report, IBM, Bethesda, 1965.

[7] A. Hotho, R. Jaschke, C. Schmitz, and G. Stumme. Information retrieval in folksonomies: Search and ranking. *Lecture Notes in Information Science*, 4011:411–426, 2006.

[8] A. Plangprasopchok and K. Lerman. Exploiting social annotation for automatic resource discovery. AAAI07, 2007.

[9] P. Schmitz. Inducing ontology from flickr tags. In *WWW*, 2006.

[10] B. Sigurbjörnsson and R. van Zwol. Flickr tag recommendation based on collective knowledge. In *WWW*, 2008.

[11] A. F. M. Smith and G. O. Roberts. Bayesian computation via the gibbs sampler and related markov chain monte-carlo methods (with discussion). In *Journal of the Royal Statistical Society*, volume 55, pages 3–23, 1993.

[12] Y.W. Teh, M.I. Jordan, M.J. Beal, and D.M. Blei. Hierarchical dirichlet processes. In *JASA 101(476)*, pages 1566–1581, 2006.

[13] X. Wu, L. Zhang, and Y. Yu. Exploring social annotations of the semantic web. In *WWW*, 2006.

[14] P. Zunde and M. E. Dexter. Indexing consistency and quality. *American Documentation*, 20(3):259–267, April 1969.