Probabilistic Generative Models of the Social Annotation Process

Said Kashoob, James Caverlee, Elham Khabiri
Department of Computer Science and Engineering
Texas A&M University
College Station, TX 77843
Email: {kashoob,caverlee,khabiri}@cse.tamu.edu

***Abstract*—With the growth in the past few years of social tagging services like Delicious and CiteULike, there is growing interest in modeling and mining these social systems for deriving implicit social collective intelligence. In this paper, we propose and explore two probabilistic generative models of the social annotation (or tagging) process with an emphasis on user participation. These models leverage the inherent social communities implicit in these tagging services. We compare the proposed models to two prominent probabilistic topic models (Latent Dirichlet Allocation and Pachinko Allocation) via an experimental study of the popular Delicious tagging service. We find that the proposed community-based annotation models identify more coherent implicit structures than the alternatives and are better suited to handle unseen social annotation data.**

I. Introduction

The past few years have seen the rapid proliferation of Web-based social systems – including online social networks like Facebook, user-contributed content sites like Flickr and YouTube, social tagging services like Delicious, among many others. These social systems have infused the traditional view of the web with a new social perspective by building on the unprecedented access to the interests and perspectives of millions of users. With this new social view of the web has come a commensurate interest in modeling and mining the wealth of new social information inherent in Web-based social systems, e.g., [1], [2], [3], [4].

One encouraging line of research is focused on the collective social intelligence embedded in the socially-generated metadata on social tagging services like Delicious and CiteULike. These tagging (or annotation) systems aggregate thousands of user's perspectives on web content via simple keywords or phrases that are used to annotate (or "tag") web pages, images, videos, and other web media. Although tags are applied by a large and heterogeneous tagger population, previous research has identified clear patterns in these systems, including the stabilization of tags over time [5] and a power-law distribution of tags applied to web pages [6]. Inspired by these and similar results (e.g., [7], [8], [9], [10]), we are interested to study more closely the underlying process that governs how a user and a tagging community "generate" tags.

Understanding the social annotation process is essential to modeling the collective semantics centered around large-scale social annotations, which is the first step towards potential improvements in information discovery and knowledge sharing. In one direction, the wealth of socially-generated metadata on social tagging services has spurred new social approaches for augmenting traditional web search and browsing, e.g., [11], [12], [13],[14], [15], [16],[17].

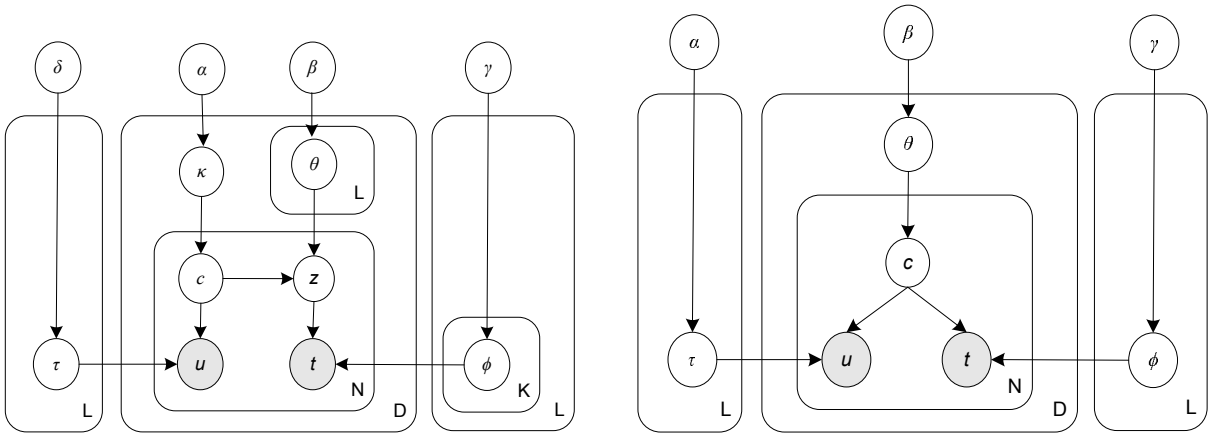
Concretely, we focus in this paper on developing probabilistic generative models for describing and modeling the social annotation process. Fundamental to our study is the notion of community. One of the hallmarks of social systems is this notion of community – be it friendships on Facebook, groups of users who tag a particular subset of web documents, users who share common interests, and so on. Since social systems are inherently community-based, it may be advantageous to explicitly model community in the tag generation process.

Based on this observation, we propose and evaluate two community-based probabilistic social annotation models for modeling the user and community perspective in social tagging. These generative models identify implicit interest-based communities of users that provide a deeper understanding of the social annotation process. We compare the proposed models with two probabilistic topic models: Latent Dirichlet Allocation (LDA) [18] and Pachinko Allocation (PAM) [19]. We find that the proposed models improve the empirical likelihood of held-out test data and that they discover more coherent latent communities.

II. Overview

The underlying structures formed by resources, tags, and users of tagging systems are a valuable source of information. We view social tagging as a combination of processes that generate the social web object by creating content (text, photos, videos), creating labels for the content, and creating users that annotate the content. In the abstract, we assume these processes to operate on separate vocabulary sets (content, tags, and users) from which a rich social web object is generated. From this perspective, the basic unit of the social tagging system is the social web object. Every such object is uniquely identified by its content, its tags, and its annotators. The object content is sequence of words, frames, etc. assigned to it by its author(s), the object's labels is sequence of tags given to it by various users, and the objects annotators is the sequence of users that choose to annotate the object.

Web objects carry different meanings that allow for a spectrum of interpretations. We assume these interpretations



(a) PSA (b) PSA (simplified version)

Fig. 1. Probabilistic Social Annotation models

to be be apparent in the tags assigned to the object by the viewers. We account for these meanings and interpretations in our models by letting communities have categories from which they select terms that fit their interpretation of the object, that is, the community interprets an object by assigning it proportions over the community's set of categories. This collective interpretation has been shown to emerge in social annotations. Naturally, Web objects do not get uniform attention and interest from all users and communities. Consequently, in our models we account for community interest per object.

A. Reference Model

Formally, we consider a universe of discourse $\mathcal{U}$ consisting of D socially annotated objects, U users, and a vocabulary of V tags. For each of the D objects $\{O_1, O_2, ..., O_D\}$, each socially annotation object O_i is modeled by both its intrinsic content C_i and the social annotations S_i attached to it by the community of users. Hence, each object is a tuple $O_i = \langle C_i, S_i \rangle$ where the content and the social annotations are modeled separately.

We call the social annotations S_i applied to an object its *social annotation document*. For example, the object corresponding to a web page annotated in the Delicious community would consist of the HTML contents of the web page as well as the *social annotation document* generated by the members of the Delicious community.

[Definition] Social Annotation Document: *For an object $O \in \mathcal{U}$, we refer to the collection of tags assigned to the object as the object's social annotation document S, where S is modeled by the set of users and the tags they assigned to the object: $S = \{\langle user_j, tag_j \rangle\}$.*

In contrast to web pages and text documents that are typically written by a single author or a team working together, a social annotation document is "written" by contributors that are largely unaware of each other and the tagging decisions made by others. We hypothesize that these contributors belong to implicit *communities* of interest.

B. Modeling Community

The collaborative tagging environment allows an object to be tagged by users with various interests, expertise, and in various languages. For example, an image of a Tyrannosaurus rex may be annotated by a scientist e.g., with tags like `cretaceous` and `theropod`), by an elementary school student (e.g., with tags like `meat-eater` and `t-rex`) and by a French-speaking tagger (e.g., with tags like `carnivore` and `lézard-tyran`).

We view the underlying groups that form around these interests, expertise, and languages as distinct *communities*. Concretely, we assume the existence of L distinct communities that are implicit in the universe of discourse $\mathcal{U}$, where each community is a mixture of users that view the world. Since community membership is not explicit in the social tagging world, we model it as a probability distribution, where each user has some probability of belonging to any community:

[Definition] Social Annotation Community: *A social annotation community c is a probability distribution over users in U such that $\sum_{u \in U} p(u|c) = 1$, where $p(u|c)$ indicates membership strength for each user u in community c.*

For each community, there may be some number of underlying *categories* that inform how each community views the world. Continuing our example, the scientist community may have underlying categories centered around Astronomy, Biology, Paleontology, and so on. For each object, the community selects tags from the appropriate underlying category or mixture of categories (e.g., for tagging the dinosaur, the tags may be drawn from both Biology and Paleontology). The fraction of community members that annotate a given object is indicative of the community interest in that object. Hence, we model for each community a set of K_l hidden categories, where each category is a mixture of tags.

[Definition] Social Annotation Category: *A social annotation category z is a probability distribution over tags in the vocabulary V such that $\sum_{t \in V} p(t|z) = 1$, where $p(t|z)$ indicates membership strength for each tag t in category z.*

In practice, communities and categories are *hidden* from us; all we may observe is the social annotation document that is a result of these communities and the categories they have selected. Inspired by recent work on LDA and other text-based topic models, we propose in the next section a generative probabilistic model for discovering communities of users.

III. Probabilistic Social Annotation Model

In this section we propose probabilistic generative models that aim to model the social annotation process by modeling the communities of interest that engage in social tagging and the implicit categories that each community considers.

A. Preliminaries

The probabilistic social annotation model is inspired by related work in text-based topic modeling. A topic model typically views the words in a text document as belonging to hidden (or "latent") conceptual topics. Prominent examples of latent topic models include Latent Semantic Analysis (LSA) [20], Probabilistic Latent Semantic Analysis (pLSA) [21], and Latent Dirichlet Allocation (LDA) [18].

In our case, an LDA-based model can be easily adapted to social annotations by considering the document unit to be a social annotation document and the underlying topics to be social annotation categories. Since LDA is typically used in document-based modeling and not tag-based modeling, we shall refer to the adapted version as TagLDA for clarity.

TagLDA: TagLDA views a tag document as a mixture of latent categories (or topics), where each category is a multinomial over tags in the vocabulary space. Formally, let Φ be a $K \times V$ matrix representing categories, where each ϕ_k is a distribution over tags for category k, K is the number of categories, and V is the size of tag vocabulary. Similarly, object are represented by $D \times K$ matrix Θ, where each θ_S is a distribution over categories for object S.

The TagLDA generative process is as follows:

1) for each category $z = 1, ..., K$
   - select V dimensional $\phi_z \sim$ Dirichlet(β)
2) for each object S_i, $i = 1, ..., D$
   - select K dimensional $\theta_i \sim$ Dirichlet(β)
   - For each tag t_j, $j = 1, ..., N_i$
     - Select a category $z_j \sim$ multinomial(θ_i)
     - Select a tag $t_j \sim$ multinomial(ϕ_{z_j})

Based on this generation process, a number of standard procedures (e.g., [18], expectation propagation [22], or Gibbs sampling [23]) can be used to infer the distribution of tags ϕ_k over each category k.

B. The PSA Model

TagLDA provides a foundation for discovering communities in social tags. Fundamentally, however, a social annotation document is a collaborative effort among many taggers, whereas TagLDA is a topic model with no notion of authorship or community. In essence TagLDA can be used to discover social annotation categories over tags, but not social annotation communities over users, since users are not explicitly modeled in the generation process. Recent work on author-topic models [24] has added the concept of "author" to the LDA model, but fundamentally these models are designed to model text documents that have a single (or a few) authors. In contrast, a social annotation document is the product of (potentially) hundreds of authors. These observations suggest a new approach.

Rather than modeling the tag generation process as if tags are generated regardless of user, the Probabilistic Social Annotation (PSA) model combines the (user,tag) generation process; a natural consequence of such an approach is the discovery of user communities in addition to tag categories. Formally, the PSA model assumes a corpus of D social annotation documents drawn from a vocabulary of V tags and U users, where each social annotation document S_i is of variable length N_i. The model assumes that the $\langle user, tag \rangle$ pairs in a social annotation document are generated from a mixture of L distinct communities, where each community is a mixture of users that view the world based on a set of K_l hidden categories, and where each category is a mixture of tags. Therefore, the tagging process involves two steps: 1) the selection of a community from which to draw users and 2) the selection of the categories that influence the user's view or preference over tags based on the object's content, and the tagger's perception of the content. Let $\mathbf{S}_i, \mathbf{z}$, and $\mathbf{c}$ be vectors of length N_i representing $\langle user, tag \rangle$ pair, category, and community assignments, respectively, in a social annotation document. The PSA model generation process is illustrated in Figure 1(a) and described here:

1) for each community $c = 1, ..., L$
   - Select U dimensional $\tau_c \sim$ Dirichlet(δ)
   - for each category $z = 1, ..., K_c$
     - select V_c dimensional $\phi_z \sim$ Dirichlet(γ)
2) for each object $\mathbf{S}_i$, $i = 1, ..., D$
   - Select L dimensional $\kappa \sim$ Dirichlet(α)
   - for each community $c = 1, ..., L$
     - select K_c dimensional $\theta_c \sim$ Dirichlet(β)
   - For each position $S_{i,j}$, $j = 1, ..., N_i$
     - Select a community $\mathbf{c}_{i,j} \sim$ multinomial(κ_i)
     - Select a user $\mathbf{S}^u_{i,j} \sim$ multinomial($\tau_{\mathbf{c}_{i,j}}$)
     - Select a category $\mathbf{z}_{i,j} \sim$ multinomial($\theta_i^{\mathbf{c}_{i,j}}$)
     - Select a tag $\mathbf{S}^t_{i,j} \sim$ multinomial($\phi^{\mathbf{z}_{i,j}}_{\mathbf{c}_{i,j}}$)

A social annotation document's community distribution $\kappa_i = \{\kappa_{i,j}\}_{j=1}^{L}$ (representing the communities interest in the object) is sampled from a Dirichlet distribution with parameter $\alpha = \{\alpha_i\}_{i=1}^{L}$. A per object community's category distribution $\theta_i^c = \{\theta_{i,j}^c\}_{j=1}^{K}$ (representing the community interpretation of the object) is sampled from a Dirichlet distribution with parameter $\beta = \{\beta_i\}_{i=1}^{K}$. A category's tag distribution $\phi_z = \{\phi_{z,i}\}_{i=1}^{|V|}$ (representing a topic of interest) is sampled from a Dirichlet distribution with parameter $\gamma = \{\gamma_i\}_{i=1}^{|V|}$. Finally, A community's user distribution $\tau_c = \{\tau_{c,i}\}_{i=1}^{|U|}$ (representing a group of users with common interests) is sampled from a Dirichlet distribution with parameter $\delta = \{\delta_i\}_{i=1}^{|U|}$. The

generative process creates a social annotation document by sampling for each position $\mathbf{S}_{i,j}$ a community $\mathbf{c}_{i,j}$ from a multinomial distribution with parameter κ_i, a category $\mathbf{z}_{i,j}$ from a multinomial distribution with parameter $\theta_i^{\mathbf{c}_{i,j}}$. A user is then sampled for that position form a multinomial distribution with parameter $\tau_{\mathbf{c}_{i,j}}$. Similarly a tag is sampled for that position from a multinomial distribution with parameter $\phi_{\mathbf{c}_{i,j}}^{\mathbf{z}_{i,j}}$.

Based on the model we can write the likelihood that a position $\mathbf{S}_{i,j}$ is assigned a specific $\langle user, tag\rangle$ pair $\{u,t\}$ as:

$$p(\mathbf{S}_{i,j}=\{u,t\}|\kappa_i,\Theta,\Phi,\tau)=\sum_{l=1}^{L}p(\mathbf{S}_{i,j}^u=u|\tau_l)p(\mathbf{c}_{i,j}=l|\kappa_i)\left(\sum_{k=1}^{K_l}p(\mathbf{S}_{i,j}^t=t|\phi_l^k)p(\mathbf{z}_{i,j}=k|\theta_i^l)\right)$$

Furthermore, the likelihood of the complete social annotation document $\mathbf{S}_i$ is the joint distribution of all its variables (observed and hidden):

$$p(\mathbf{S}_i,\mathbf{z}_i,\mathbf{c}_i,\kappa_i,\Theta,\Phi,\tau|\alpha,\beta,\gamma,\delta)=\prod_{j=1}^{N_i}p(\mathbf{S}_{i,j}^t|\phi_{\mathbf{c}_{i,j}}^{\mathbf{z}_{i,j}})p(\mathbf{S}_{i,j}^u|\tau_{\mathbf{c}_{i,j}})p(\mathbf{z}_{i,j}|\theta_i^{\mathbf{c}_{i,j}})p(\mathbf{c}_{i,j}|\kappa_i)$$

Integrating out the distributions κ_i, Θ, τ and Φ and summing over $\mathbf{c}_i$ and $\mathbf{z}_i$ gives the marginal distribution of $\mathbf{S}_i$ given the priors:

$$p(\mathbf{S}_i|\alpha,\beta,\gamma,\delta)=\int\int\int\int p(\kappa_i|\alpha)p(\Theta|\beta)p(\Phi|\gamma)p(\tau|\delta)\prod_{j=1}^{N_i}\sum_{\mathbf{c}_{i,j}}p(\mathbf{S}_{i,j}^u|\tau_{\mathbf{c}_{i,j}})p(\mathbf{c}_{i,j}|\kappa_i)\left(\sum_{\mathbf{z}_{i,j}}p(\mathbf{S}_{i,j}^t|\phi_{\mathbf{c}_{i,j}}^{\mathbf{z}_{i,j}})p(\mathbf{z}_{i,j}|\theta_i^{\mathbf{c}_{i,j}})\right)d\Phi d\Theta d\tau d\kappa_i$$

Finally our universe of discourse $\mathcal{U}$ consisting of all D social annotation documents occurs with likelihood:

$$p(\mathcal{U}|\alpha,\beta,\gamma,\delta)=\prod_{i=1}^{D}p(\mathbf{S}_i|\alpha,\beta,\gamma,\delta)$$

C. Parameter estimation and inference

The PSA model provides a generative approach for describing how social annotation documents are produced. Our goal it to recover the structures that produced these social annotation document – taking a set of social annotation documents and inferring the underlying model (including the hidden community and category distributions). This entails learning model parameters κ, τ, Θ, and Φ (the distributions over communities, users, categories, and tags, respectively).

Previous work that aimed at recovering similar hidden structure from joint posterior distributions has shown that exact computation of these parameters is intractable. There exists, however, several approximation methods in the literature for solving similar parameter estimation problems (like in LDA), including expectation maximization [18], expectation propagation [22], and Gibbs sampling. In this paper, we adopt Gibbs Sampling (see [23] for a thorough treatment) which is a special case of Markov-chain Monte Carlo methods that estimates a posterior distribution of a high-dimensional probability distribution. The sampler draws from a joint distribution $p(x_1,x_2,...,x_n)$ assuming the conditionals $p(x_i|x_{-i})$ are known, where $x_{-i}=(x_1,...,x_{i-1},x_{i+1},...,x_n)$.

Let $\mathbf{S},\mathbf{z}$, and $\mathbf{c}$ be vectors of length $\sum_i^D N_i$ representing $\langle user, tag\rangle$ pair, category, and community assignments, respectively, for the entire corpus. Also let u and t be user and tag variables. Following the approach used in [23] the joint probability distribution of the PSA model can be factored as:

$$p(\mathbf{S}^u,\mathbf{S}^t,\mathbf{z},\mathbf{c}|\alpha,\beta,\gamma,\delta) = p(\mathbf{S}^u|\mathbf{c},\delta)p(\mathbf{c}|\alpha)p(\mathbf{S}^t|\mathbf{z},\mathbf{c},\gamma)p(\mathbf{z}|\mathbf{c},\beta).$$

We derive the Gibbs sampler's update equation (details not shown for space considerations) for the hidden variables (community, and category) from the joint distribution and arrive at:

$$p(\mathbf{z}_i=k,\mathbf{c}_i=l|\mathbf{z}_{\neg i},\mathbf{c}_{\neg i},\mathbf{S}^t,\mathbf{S}^u)\propto \frac{n_{l,\neg i}^u+\delta_u}{\sum_{u=1}^U n_{l,\neg i}^u+\delta_u}\times\frac{n_{lk,\neg i}^t+\gamma_t}{\sum_{t=1}^V n_{lk,\neg i}^t+\gamma_t}\times\frac{n_{S,\neg i}^{lk}+\beta_{lk}}{\left(\sum_{k=1}^{K_l}n_S^{lk}+\beta_{lk}\right)-1}\times\frac{n_{S,\neg i}^{l}+\alpha_{l}}{\left(\sum_{l=1}^{L}n_S^{l}+\alpha_{l}\right)-1} \quad (1)$$

where $n_{(\cdot),\neg i}^{(\cdot)}$ is a count excluding the current position assignments of $\mathbf{z}_i$ and $\mathbf{c}_i$ (e.g., $n_{lk,\neg i}^t$ is the count of tag t generated by the k-th category of the l-th community excluding the current position).

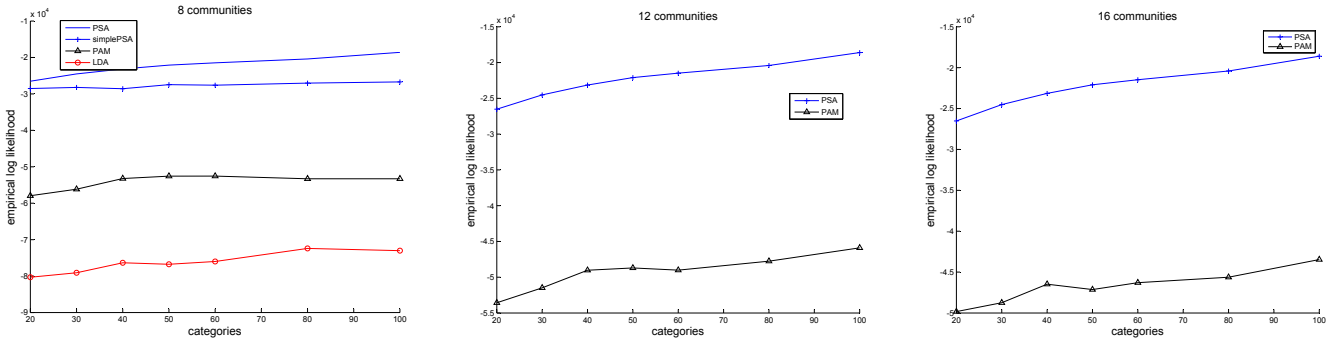
For the purpose of inference of new unseen web objects based on a model $\mathcal{M}$, the update equation for the Gibbs sampler is the following:

$$p(\tilde{\mathbf{z}}_i=k,\tilde{\mathbf{c}}_i=l|\tilde{\mathbf{c}}_{\neg i},\tilde{\mathbf{z}}_{\neg i},\tilde{S}^t,\tilde{S}^u,\mathcal{M})\propto \frac{\tilde{n}_{l,\neg i}^u+n_l^u+\delta_u}{\sum_{u=1}^U\tilde{n}_{l,\neg i}^u+n_l^u+\delta_u}\times\frac{\tilde{n}_{lk,\neg i}^t+n_{lk}^t+\gamma_t}{\sum_{t=1}^V\tilde{n}_{lk,\neg i}^t+n_{lk}^t+\gamma_t}\times\frac{n_{\tilde{S},\neg i}^{lk}+\beta_{lk}}{\left(\sum_{k=1}^{K_l}n_{\tilde{S}}^{lk}+\beta_{lk}\right)-1}\times\frac{n_{\tilde{S},\neg i}^{l}+\alpha_{l}}{\left(\sum_{l=1}^{L}n_{\tilde{S}}^{l}+\alpha_{l}\right)-1} \quad (2)$$

where where $n_{(\cdot),\neg i}^{(\cdot)}$ are counts from the given model $\mathcal{M}$, $\tilde{n}_{(\cdot),\neg i}^{(\cdot)}$ are counts from the new objects, and $\tilde{S}$ is a new unseen object.

D. PSA: Simplified Version

Alternatively, we can simplify the Probabilistic Social Annotation model by assuming that each community of users agrees on a single category view of the world. We can then combine the *community* variable c and *category* variable z into a single hidden variable. As a result, the model finds a distribution over tags as well as a distribution over users, capturing both the users' similarities/interests and their world view simultaneously. The generation process is illustrated in Figure 1(b) and works as follows:



(a) 8 communities (b) 12 communities (c) 16 communities

Fig. 2. Empirical likelihood results

1) for each community $c = 1, ..., L$
   - Select U dimensional $\tau_c \sim \text{Dirichlet}(\alpha)$
   - select V dimensional $\phi_c \sim \text{Dirichlet}(\gamma)$
2) for each object $\mathbf{S}_i$, $i = 1, ..., D$
   - Select L dimensional $\theta_i \sim \text{Dirichlet}(\beta)$
   - For each position $\mathbf{S}_{i,j}$, $j = 1, ..., N_i$
     - Select a community $\mathbf{c}_{i,j} \sim \text{multinomial}(\theta_i)$
     - Select a user $\mathbf{S}^u_{i,j} \sim \text{multinomial}(\tau_{\mathbf{c}_{i,j}})$
     - Select a tag $\mathbf{S}^t_{i,j} \sim \text{multinomial}(\phi_{\mathbf{c}_{i,j}})$

An advantage of simplifying the model is a lower computational complexity. A drawback, however, is you restrict members of a community to a single world view; which slightly reduces model generalization to unseen data as seen in our results.

The update equation for the Gibbs sampler (1) reduces to:

$$p(\mathbf{c}_i = k|\mathbf{c}_{\neg i}, \mathbf{S}^t, \mathbf{S}^t) \propto \frac{n^u_{k,\neg i} + \alpha_u}{\sum_{u=1}^{U} n^u_{k,\neg i} + \alpha_u} \times \frac{n^t_{k,\neg i} + \gamma_t}{\sum_{t=1}^{V} n^t_{k,\neg i} + \gamma_t} \times \frac{n^k_{S,\neg i} + \beta_k}{\left(\sum_{k=1}^{K} n^k_S + \beta_k\right) - 1} \quad (3)$$

and the Gibbs sampler predictive update equation (2) becomes:

$$p(\tilde{\mathbf{c}}_i = k|\tilde{\mathbf{c}}_{\neg i}, \tilde{S}^t, \tilde{S}^u, \mathcal{M}) \propto \frac{\tilde{n}^u_{k,\neg i} + n^u_k + \alpha_u}{\sum_{u=1}^{U} \tilde{n}^u_{k,\neg i} + n^u_k + \alpha_u} \times \frac{\tilde{n}^t_{k,\neg i} + n^t_k + \gamma_t}{\sum_{t=1}^{V} \tilde{n}^t_{k,\neg i} + n^t_k + \gamma_t} \times \frac{n^k_{\tilde{S},\neg i} + \beta_k}{\left(\sum_{k=1}^{K} n^k_{\tilde{S}} + \beta_k\right) - 1} \quad (4)$$

IV. Experiments

In this section we evaluate the quality of the PSA models over the prominent social tagging service Delicious. Our goal is to evaluate how well the model predicts previously unseen data and the quality of the discovered latent structures.

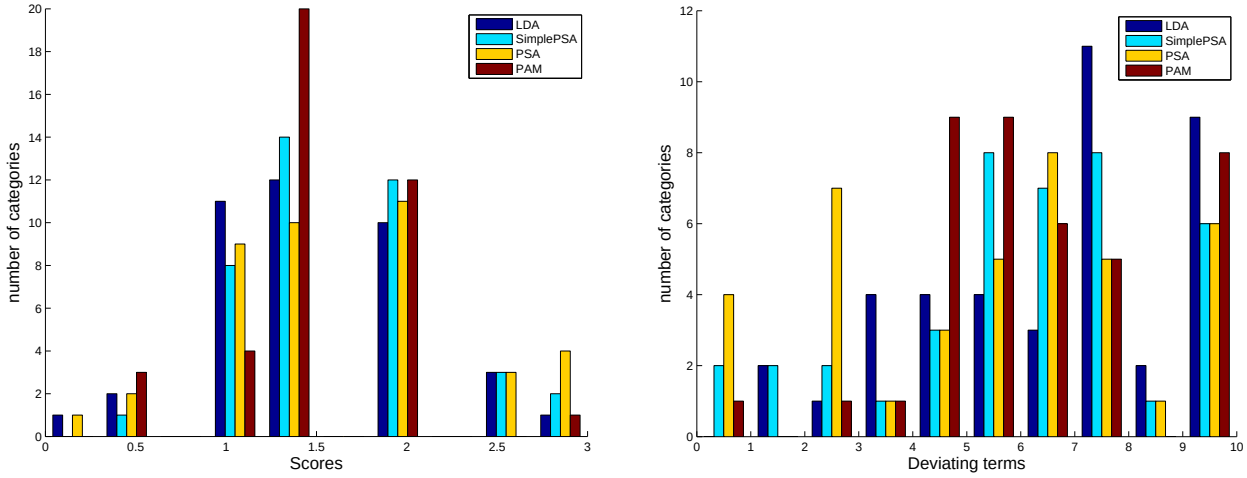
Dataset: The Delicious crawler starts with a set of popular tags. Our crawler has discovered 607,904 unique tags, 266,585 unique Web pages annotated by Delicious, and 1,068,198 unique users. Of the 266,585 total Web pages, we have retrieved the full HTML for 47,852 pages. After removing rare tags and users and normalizing/stemming the tags, we filter the set to keep only pages in English with a minimum length of 20 words, leaving us with 27,572 Web pages with 16,216 unique annotations and 150,264 unique users. We use $20,000$ of the objects to train the models and the remaining $7,572$ are used for testing.

A. Training the models

We compared PSA and the simplified version of PSA (simplePSA) against TagLDA and a tag-based version of Pachinko Allocation (PAM) [19]. PAM is an LDA extension that aims to capture correlations among latent topics using a directed acyclic graph. We focus here on the four level PAM where the internal nodes in the tree represent super-topic/subtopic distributions and leaf nodes are distributions over the vocabulary space. For our purposes, we refer to the super-topics as communities, subtopics as categories, and our documents are collections of tags assigned by various users.

We use the public implementations of LDA and PAM distributed in the Mallet toolkit [25]. Hyperparameters for both models are set to toolkit standard: for LDA $(\alpha = 50/K, \beta = 0.01)$ and for PAM $(\alpha = 50/K, \beta = 0.001)$ with optimization enabled for both models. For the PSA models we experimented with several combinations of hyperparameters. We also estimate hyperparameters using the fixed-point iteration method in [26]. The results we compare with other models are run with hyperparameters $(\alpha = 0.1, \beta = 1, \gamma = 0.1, \delta = 0.1)$ and optimization enabled.

For all models, a Gibbs sampler starts with randomly assigned communities/categories, runs for 2000 iterations with optimization every 50 iterations and an initial burnin-period of 250 iterations. For TagPAM we experiment with three communities sizes $(8, 12, 16)$ each with 20 to 100 categories. For PSA we experiment with community category combinations that result in total number of categories from 20 to 100. simplePSA and TagLDA – which have no community/category hierarchy – are run from 20 to 100 categories (although recall that simplePSA models user communities and tag categories simultaneously)



(a) scores (b) deviating terms

Fig. 3. User study results: (a) shows the count of categories form each model and their respective score (0 to 3), with 0 representing no coherence and 3 representing excellent coherence (b) shows the count of categories from each model and their respective number of deviating terms

TABLE I
USER STUDY RESULTS

Average	LDA	PAM	PSA	simplePSA
score	1.51	1.49	1.60	1.59
number of deviating terms	5.72	5.7	5.34	5.35

B. Evaluation

We compare all the models using two metrics: 1) ability to predict previously unseen data and 2) quality of discovered latent structures.

We evaluate each model's generalization to unseen data using the empirical likelihood method [19]. To compute empirical likelihood we generate 1000 documents based on the models generative process. We then build a multinomial over the vocabulary space from these samples. Finally, we compute the empirical likelihood of a held out testing set using the obtained multinomial over the vocabulary space.

For quality evaluation we solicit human judgments on the coherence of discovered categories. We conduct a user study to judge the coherence of the categories uncovered by the models. Categories from each model were anonymized and put in random order. Each user is asked to judge the category coherence by trying to detect a theme from the category's top 10 terms. Coherence is graded on a $0-3$ scale with 0 being poor coherence and 3 excellent coherence. The users are also asked to report the number of terms that deviate form the theme they thought the category represented.

We use testing sets of size 1%, and 10% of the size of the training set for testing all models. The empirical likelihood results are consistent for the two sets, therefore we report results from the smaller set.

We plot the empirical likelihood results in Figure 2. The y-axes show the empirical log likelihood and the x-axes show the number of categories. Focusing on Figure 2(a), the PSA model performs the best, followed by simplePSA, TagPAM and TagLDA respectively. PSA and simplePSA performance improves with increasing number of categories, with PSA spiking at 35 categories then slowly continuing to improve. simplePSA behaves similarly with its initial spike at 50 categories. TagPAM's performance improves initially, peaks around $40-50$ categories, then decreases slightly and stabilizes. Likewise, the performance of TagLDA peaks around 40 categories, decreases slightly, then peaks again at 80 categories. The results shown Figure 2(b and c) show similar results with improved performance for TagPAM when number of communities is increased. Still the PSA model performs better than TagPAM.

Based on the above results, 40 categories lead to good performance in all models. For our user study we present the discovered 40 categories from each model for coherence evaluations. A sample of these categories is shown in Table II.

A group of four evaluators judged the categories' coherence and noted the deviating terms. The average user studies results are as shown in Table 1 . Evaluating coherence, we can see from the Table that on average, PSA and simplePSA perform the best followed by TagPAM and TagLDA respectively. PSA shows on average a 6% improvement over TagLDA and a 7% improvement over TagPAM. We also look at the number of deviating terms from the perceived theme and observe similar improvements.

In Figure 3 we report the detailed user study scores and and deviating terms for all categories from all four models. Figure 3(a) shows the number of categories from each model and the coherence scores they received. Notice that PSA and simplePSA have higher number of categories receiving a score of 2 or higher compared to TagPAM and TagLDA . We can also see that PSA and simplePSA have lower number of categories receiving a score of 1 or lower compared to

TagPAM and TagLDA. Figure 3(b) shows the number of categories from each model versus the number of deviating terms. Again PSA and simplePSA have a higher number of categories containing small number of deviating terms compared to TagPAM and TagLDA and they have a lower number of categories containing large number of deviating terms compared to TagPAM and TagLDA.

C. Computational complexity

As we have seen above, learning model parameters via Gibbs sampling involves iterations over the entire corpus that sample conditional probabilities for communities and categories at each position. let N be number of iterations, L be number of communities, K be number of categories, D be number of documents, and S be average document length. The computational complexity of PSA using Gibbs sampling is $\mathcal{O}(NLKDS)$. That is a factor of L higher than LDA and simplePSA. The complexity of PAM is $\mathcal{O}(N(L+K+1)DS)$.

V. The Role of Users

The improvements achieved by or models are due primarily to the inclusion of the user as a generated variable. Smaller improvement comes from the hierarchical structure of communities and categories that we introduce. This is clearly evident in Figure 2(a). Notice the performance of PSA compared to that of simplePSA. In this section, we show how the introduction of the user as a generated variable in the social annotation process impacts the Gibbs sampler.

As an example, suppose we have a corpus with a tag vocabulary of length 3, $V = \{w_1, w_2, w_3\}$, two users, $U = \{u_1, u_2\}$, and two communities $L = \{C_1, C_2\}$. Also suppose the corpus contains a single document of length 6, $\mathbf{S} =< u_1, w_1; u_1, w_2; u_1, w_2; u_1, w_1; u_2, w_2; u_2, w_3 >$. Assume the priors to be uniform over tags, users, and communities. Table III compares the impact of excluding/including users on the Gibbs sampler. Suppose the Gibbs sampler has completed $n-1$ iterations and the resulting community assignments are as shown in the 5_{th} row of the table. To illustrate this difference, we use Equation (3) to compute the probability of community assignment for the word at position 1 of the corpus. Remember that our Gibbs sampler excludes the current position, so at the beginning of iteration n two words of the corpus belong to community C_1 and the other three words belong to community C_2.

When calculating the update probabilities in the case that excludes users we ignore the first factor of Equation (3). Notice that w_1 which occurs at position 4 is assigned to community C_1. The Gibbs sampler gives almost equal chance for this word to belong to either community. This is because of two competing factors: (i) the majority of the words in this document already belong to community C_2, so one factor favors C_2; (ii) at the same time the word w_1 has already been assigned once to community C_1 which balances both outcomes.

Now let us consider the case in which users are included. Here the Gibbs sampler clearly prefers community C_1 over C_2 for this position. The reason being that the user associated with the word at position 1 which also happen to be the user most interested in this document had already been assigned twice to C_1. We can point to at least three advantages of including users in social annotation modeling 1) faster convergence; users co-occurrences and associations with tags resolve ties leading to faster consensus 2) better results in terms of quality of categories as shown in our user study 3) additional clustering of users that can be useful in numerous application.

VI. Related Work

The past few years have seen an increased interest in modeling social annotations. Several works that adapt topic-modeling based approaches for modeling social annotations include mapping tags, users, and content to a single underlying conceptual space [15], mapping combined content and tags to an underlying topic space [27], mapping content, tags and additional link information to multiple underlying topic spaces [28]. Additionally, In [29] and [30], the authors assume hidden structure of interests and topics that generate tags for resources. They then are able to discover related resources based on their relevance (distributions) to interests and topics.

In previous work [31] we developed a related generative model to the PSA model presented in this paper. The previous model also finds mixtures of tag categories, however, it does not model users at all in the annotation generation process. As a result, the user-based community aspect so important to the PSA model is omitted entirely.

The work presented here builds on these previous efforts in the following ways: (i) groups of users with a common understanding or similar interpretations of resources are represented as a community (ii) each community has a world view encoded by a set of categories (iii) we recognize that the annotation process involves a user choice to participate in annotating an object and a subsequent decision on the type of tags to be used and place that at the core of our models. With the maturing of this area, it would be interesting for a comprehensive cross-analysis of PSA and related annotation models.

VII. Conclusion

Understanding the social annotation process is essential to modeling the collective semantics centered around large-scale social annotations; which is the first step towards potential improvements in information discovery and knowledge sharing. In this work, we have introduced two novel probabilistic generative models of the social annotation process, emphasizing the user/community role as a major actor in this domain. We compare our models to two prominent topic models (PAM and LDA). Our experimental results show improvements in models generalizing to unseen social annotation documents as well as improvements in the quality of latent structures they discovered.

In our continuing work, we are considering applications based on the results of the social annotation models we have introduced here.

TABLE II
SAMPLE CATEGORIES UNCOVERED BY THE DIFFERENT MODELS

LDA	airlin flight seat hotel businesscard airfar airplan card vacat ticket firefox scalabl cluster amazon scale jobsearch ec blueprint tumblr mapreduc tumblelog screensav fileshar religion evolut bibl opensoci restaur christian foodblog lego steampunk vista
PAM	airlin flight seat hotel webcom airfar airplan vacat ticket xkcd cheap scalabl cluster speed scale deploy number shoe tune concurr mapreduc bandwidth religion cloud evolut bibl tagcloud oreilli christian comedi flex flashcard ibm
PSA	airlin flight seat hotel airfar airplan vacat ticket cheap trip holidai django cach scalabl cluster eclips amazon scale j2ee memcach trac s3 religion recycl bibl eco evolut christian lego consumer knot ecolog garden
simplePSA	airlin flight seat airfar airplan deal ticket coupon bargain cheap hotel authent scalabl cluster openid rest amazon webhost scale opensoci s3 ec2 religion evolut bibl christian sga lego mckaysheppard genet atheism dna church

TABLE III
GIBBS SAMPLING EXAMPLE WITH AND WITHOUT USERS

	Without Users							With Users					
corpus	w_1	w_2	w_2	w_1	w_2	w_3		u_1, w_1	u_1, w_2	u_1, w_2	u_1, w_1	u_2, w_2	u_2, w_3
Gibbs Sampling	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
Iteration $n-1$	C_1	C_1	C_2	C_1	C_2	C_2		C_1	C_1	C_2	C_1	C_2	C_2
Iteration n	$P(c_1 = C_1) = 0.48$ $P(c_1 = C_2) = 0.52$	C_1	C_2	C_1	C_2	C_2		$P(c_1 = C_1) = 0.83$ $P(c_1 = C_2) = 0.17$	C_1	C_2	C_1	C_2	C_2

VIII. ACKNOWLEDGMENTS

The first author is supported by a scholarship form the Ministry of Manpower, Oman. This work is partially supported by faculty startup funds from Texas A&M University and the Texas Engineering Experiment Station.

REFERENCES

[1] L. Kennedy and M. Naaman, "Less talk, more rock: automated organization of community-contributed collections of concert videos," in *WWW '09*, 2009, pp. 311–320.
[2] M. Goetz, J. Leskovec, M. Mcglohon, and C. Faloutsos, "Modeling blog dynamics," in *ICWSM '09*, May 2009.
[3] U. Brandes, P. Kenis, J. Lerner, and D. van Raaij, "Network analysis of collaboration structure in wikipedia," in *WWW '09*, 2009, pp. 731–740.
[4] D. J. Crandall, L. Backstrom, D. P. Huttenlocher, and J. M. Kleinberg, "Mapping the world's photos," in *WWW '09*, 2009, pp. 761–770.
[5] S. Golder and B. A. Huberman, "The structure of collaborative tagging systems," Aug 2005. [Online]. Available: http://arxiv.org/abs/cs/0508082
[6] H. Halpin, V. Robu, and H. Shepherd, "The complex dynamics of collaborative tagging," in *WWW '07*, 2007.
[7] C. Cattuto, V. Loreto, and L. Pietronero, "Collaborative tagging and semiotic dynamics," May 2006. [Online]. Available: http://arxiv.org/abs/cs.CY/0605015
[8] C. Cattuto, A. Baldassarri, V. D. P. Servedio, and V. Loreto, "Vocabulary growth in collaborative tagging systems," Apr 2007. [Online]. Available: http://arxiv.org/abs/0704.3316
[9] X. Li, L. Guo, and Y. E. Zhao, "Tag-based social interest discovery," in *WWW '08*, 2008, pp. 675–684.
[10] C. Veres, "The language of folksonomies: What tags reveal about user classification," *Natural Language Processing and Information Systems*, pp. 58–69, 2006.
[11] S. Bao, G. Xue, X. Wu, Y. Yu, B. Fei, and Z. Su, "Optimizing web search using social annotations," in *WWW '07*, 2007.
[12] C. H. Brooks and N. Montanez, "Improved annotation of the blogosphere via autotagging and hierarchical clustering," in *WWW'06*.
[13] P. Heymann, G. Koutrika, and H. Garcia-Molina, "Can social bookmarking improve web search?" in *WSDM '08*, 2008.
[14] R. Li, S. Bao, Y. Yu, B. Fei, and Z. Su, "Towards effective browsing of large scale social annotations," in *WWW '07*, 2007.
[15] X. Wu, L. Zhang, and Y. Yu, "Exploring social annotations for the semantic web," in *WWW '06*, 2006, pp. 417–426.
[16] S. Xu, S. Bao, B. Fei, Z. Su, and Y. Yu, "Exploring folksonomy for personalized search," in *SIGIR '08*, 2008, pp. 155–162.
[17] Y. Yanbe, A. Jatowt, S. Nakamura, and K. Tanaka, "Can social bookmarking enhance search in the web?" in *JCDL '07*, 2007, pp. 107–116.
[18] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," in *Journal of Machine Learning Research*, vol. 3, April 2003, pp. 993–1022.
[19] W. Li and A. Mccallum, "Pachinko allocation: Dag-structured mixture models of topic correlations," in *ICML '06*, 2006, pp. 577–584.
[20] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society of Information Science*, vol. 41, no. 6, pp. 391–407, 1990.
[21] T. Hofmann, "Probabilistic latent semantic indexing," in *SIGIR '99*, 1999, pp. 50–57.
[22] T. Minka and J. Lafferty, "Expectation-propagation for the generative aspect model," in *UAI '03*, 2003, pp. 352–359.
[23] G. Heinrich, "Parameter estimation for text analysis," Tech. Rep., 2004.
[24] A. Mccallum, X. Wang, and A. Corrada-Emmanuel, "Topic and role discovery in social networks," in *IJCAI*, 2005.
[25] A. K. McCallum, "Mallet: A machine learning for language toolkit," 2002, http://mallet.cs.umass.edu.
[26] T. P. Minka, "Estimating a dirichlet distribution," 2003. [Online]. Available: http://research.microsoft.com/~minka
[27] D. Zhou, J. Bian, S. Zheng, H. Zha, and C. L. Giles, "Exploring social annotations for information retrieval," in *WWW '08*, 2008, pp. 715–724.
[28] D. Ramage, P. Heymann, C. D. Manning, and H. G. Molina, "Clustering the tagged web," in *WSDM '09*, 2009, pp. 54–63.
[29] A. Plangrasopchok and K. Lerman, "Exploiting social annotation for automatic resource discovery," in *Proceedings of AAAI workshop on Information Integration*, 2007.
[30] A. Plangprasopchok and K. Lerman, "Modeling social annotation: a bayesian approach," 2008.
[31] S. Kashoob, J. Caverlee, and Y. Ding, "A categorical model for discovering latent structure in social annotations," in *ICWSM'09*, 2009.