Predicting Latent User Attributes on Twitter

Yotam Hechtlinger
Department of Statistics
hechtlinger@gmail.com

Natalie Klein
Machine Learning Department
Department of Statistics
neklein@andrew.cmu.edu

Hyun Ah Song
Machine Learning Department
hi.hyunah@gmail.com

Abstract

Online social networks contain a wealth of information about users that can be harnessed to provide users with personalized content. While most websites now personalize content, it is still not uncommon to have them characterize us incorrectly. We seek to extend previous work that focuses on better predicting latent user attributes for users in the Twitter social network with a flexible graphical model proposed by El-Arini et al that allows a user to understand why the system has characterized them in a particular way. We seek to add information about which 'elite users' someone follows into this model because we hypothesize that this information helps reveal a user's long-term preferences better than more transient actions. To investigate this claim, we collected data via the Twitter API and implemented a Markov Chain Monte Carlo (MCMC) sampler to infer user attributes. Through experiments on both simulated and real data, we assess the model's ability to correctly assign user attributes and to relate actions to particular user attributes. We report successful associations of actions to attributes, with some ability to predict attributes, and see that both of these tasks are enriched by including 'follower' information. In future work, we seek to scale our implementation to larger data sets and to make further modifications to the underlying graphical model structure to seek more accurate user characterization.

1 Introduction

Inferring user attributes from online social network activity remains as an important and challenging problem, particularly in networks such as Twitter, where users do not provide much personal information in a user profile. Richer and more accurate systems to determine user attributes could improve recommendation systems, promotional advertising, and personalization of the online experience for each user. Many previous social network analysis studies have focused on Twitter due to an easy-to-use API for accessing the abundant publicly available information. In particular, previous work has focused on inferring user attributes from public user behavior on Twitter, using features such as words used in tweets, tweet length and frequency, retweets, hashtags, friend connections, user profile information, profile picture, and more (see project proposal for citations).

Many previous studies used simple supervised learning techniques to infer user attributes. However, one of the key challenges when attempting to train a supervised classifier is the collection of labeled data sets to use for training. Some studies looked for user-provided information (which is sparse on Twitter), while others leveraged information from linked profiles on other social network sites, and a few even resorted to hand-labeling data. Even so, labeled data sets are typically small and quite possibly biased due to collection techniques. As an alternative, some authors have used unsupervised

techniques such as topic modeling to group and classify users, assigning labels based on the content of tweets, retweets, or hashtags. While topic models have produced some interesting results in the area, recent work by El-Arini et al. [EAPH12] argues that there are some advantages to more general, customizable graphical models over topic models.

In this project our focus will be in augmenting the graphical model presented by El-Arini et al. [EAPH12] (hereafter referred to as the 'badge model') which allows for multiple labels (called 'badges') for each user which are learned from user profile text combined with user behavior. We find this approach promising because: (1) it outperforms other approaches in user attribute assignment, (2) it is more flexible than either supervised classifiers or LDA models, and (3) there are many opportunities to experiment with the model and potentially improve upon its performance with modifications. Our goal is to improve performance by incorporating network topology information ('following' actions) into the model.

1.1 Survey of Work

The work by El-Arini et al [EAPH12] seeks to assign user attributes on Twitter by predefining a list of user attributes, such as 'Apple fanboy', 'vegetarian' or 'hipster', which are called badges. In general a particular user's badges are latent, although some users are assumed to voluntarily associate themselves with a certain badge by explicitly mentioning it in their Twitter profile. This association has enabled the authors to produce a graphical model in which the user's actions are observed, where the authors defined actions as retweets and hashtags. Using self-assigned badges (from profile information), the model learns relationships between user behavior and badges, and can then suggest new badges for users based on their behavior. Users can have many different badges, and the model is able to specifically associate each badge with the behavior that produced it, offering a new level of transparency to the user. The authors extended the model into an application [EAXFG13] which builds on the previous paper by attempting to describe documents (in this case, news articles) by the badges of users who share the articles on Twitter.

Closely related to the badge model are topic models, which have been applied in the past for user personalization. The field of topic models emerged with a classical paper by Blei et al. [BNJ03], presenting the Latent Dirichlet Allocation (LDA) model which uses the distribution of words within each document as a sample in the vocabulary space in order to describe the document. In recent years, there have been many important additions to the original LDA model. Blei and Lafferty [BL06] and Li and McMallum [LM06] relaxed the original limitation that the topic proportions in a document are uncorrelated and suggested, respectively, the Correlated Topic Model and Pachinko Allocation. Arora et al. [AGHM12] assigned each topic a unique anchor word unique to only that topic, and presented an algorithm that guarantees the convergence of the topic model parameters in a practical, scalable polynomial-time. Ramage et al. [RH09] proposed a supervised variant of LDA algorithm called Labeled LDA (or L-LDA) for credit attribution in multi-label corpora. Unlike LDA, L-LDA requires the topics to relate to explicit, observed tags by learning correspondence between tag-topic relations. By learning topics that are restricted to the ones in the label set only, L-LDA guarantees learning of features of word distributions relevant to each topic, rather than learning random groupings of words that are sometimes hard to interpret, which is sometimes a problem with LDA. In a following paper, Ramage et al. [RDL10] apply L-LDA over Twitter by mapping the Twitter feed into other dimensions using a scalable L-LDA which was then used to characterize users by the type of content they typically shared.

Important information that is generally publicly available about Twitter users includes who 'follows' the user, and who the user 'follows'. While friendship/neighbor relationships can provide some information on local structure of network, much work has focused on studying 'influential users' in the network and the impact they have on a person who follows them. For example, Cha et al. [CHBG10] quantify user influence on Twitter. Watts et al. [WHMW11] further classified users into regular and 'elite' users and looked at flow of information among the types of users. Others leveraged network topology information in making predictions, such as Zheleva et al. [ZGS10] who used Markov Random Fields to infer hidden attributes by considering both friendships and group memberships. Kwak et al. [KLPM10] discusses how a large proportion of tweets involve current news and investigate the follower-following topology with regards to information diffusion across the network.

Variable	Distribution	Meaning
$\lambda_i^{(u)}$	Bernoulli($b_i^{(u)}\gamma_i^T + (1-b_i^{(u)})\gamma_i^F$)	Observed labels in user profile
$a_j^{(u)}$	Bernoulli($1-(1-\phi_{bg,j})\prod_{i:b_i^{(u)}=1}(1-\phi_{ij}s_{ij})$)	Observed actions of user
$b_i^{(u)}$	Bernoulli(ω_i)	The badge assignment for the user
ϕ_{ij}	Beta(α_ϕ, β_ϕ)	Rate of action j given badge i
ϕ_{bg}	Beta(α_ϕ, β_ϕ)	Rate of action j for other reasons
s_{ij}	Bernoulli(η_i)	Sparsity mask for ϕ_{ij}
η_i	Beta(α_η, β_η)	Controls sparsity of s_{ij}
ω_i	Beta($\alpha_\omega, \beta_\omega$)	Prior on $b_i^{(u)}$
γ_i^T	Beta(α_T, β_T)	Prior (true positive rate for $\lambda_i^{(u)}$)
γ_i^F	Beta(α_F, β_F)	Prior (false positive rate for $\lambda_i^{(u)}$)

Table 1: Meaning of variables in the model

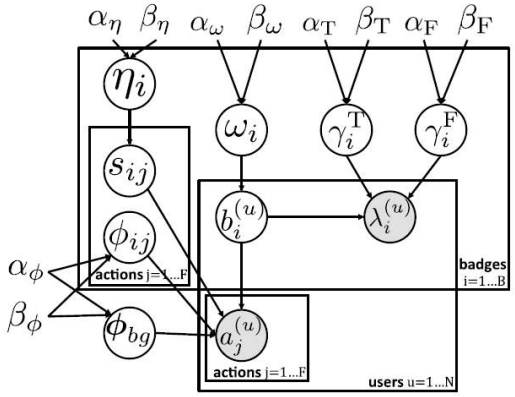


Figure 1: Graphical model plate diagram. $b_i^{(u)}$ is the latent badge, $\lambda_i^{(u)} = 1$ indicates whether the user's profile has a given label, and $a_j^{(u)} = 1$ indicates that the user performed the action j.

2 Model

2.1 Model description

The plate diagram in Figure 1 shows the structure of the graphical model. Notice that the observed variables are $\lambda_i^{(u)}$ and $a_j^{(u)}$, while the rest are latent. Table 1 gives a brief summary of the variables in the model, their distributions, and the meaning of the variable/reason for inclusion in the model. While we originally planned to modify this model further, we succeeded only in implementing the basic badge model without extensive modification. While the reader can refer to the original paper [EAPH12] for more details on the model we implemented, the plate diagram and summary of variables presented here along with the description of the model in section 1.1 should give the reader a general idea. This model is flexible and provides identifiability that would not be attained by other unsupervised approaches such as topic models (see El-Arini et al for further explanation and comparisons of this model to L-LDA).

2.2 Inference

In order to perform inference on this graphical model, we followed the suggestions in El-Arini et al to use a collapsed Gibbs sampler, interleaving Metropolis-Hastings steps for variables that are more difficult to sample directly. Collapsed Gibbs sampling marginalizes out many of the variables,

reducing the total number of variables to be sampled, so we are left having to sample only four variables using expressions that depend only on the hyperparameters but not on the values of the prior variables themselves. As we worked through the derivations presented in the appendix to the El-Arini paper, we found the motivation behind using Metropolis-Hastings to sample three of the variables due to the complicated and potentially numerically unstable calculations needed to sample directly. Therefore, we sample $b_i^{(u)}$ as a regular Gibbs sampling, but sample the three other variables ($\lambda_i^{(u)}$, ϕ_{ij}, and ϕ_{bg}) using Metropolis-Hastings steps. The predictions of badges and associations of badges with specific actions are determined by examining sample averages that represent estimates of the expected value of posterior distributions.

3 Implementation

3.1 Data

Twitter offers data for non-commercial purposes mainly through two channels: The Stream API and the REST API. We used the Python package Twython to connect to both of the APIs and collect data. First, we connected to the Stream API, which simply gives a stream of current tweets, limited to 1% of the Twitter volume, that we treat as a random sample of live Twitter activity. We limited this stream to English language tweets, but otherwise set no restrictions. We did not actually gather the full text of each tweet, but instead gathered the user name and id, the user profile text, and any hashtags or retweets the user put in the tweet. After collecting $200,000$ random tweets, we analyzed the text in the user profiles and selected badges out of the most commonly used words that we also felt would be unique (we avoided words like 'enthusiast', which occurred frequently, but do not uniquely characterize users). The badges selected for our experiments were: *'rock'*, *'conservative'*, *'texas'*, *'pop'*, *'feminist'*, *'photographer'*, *'beer'* and *'and'*. The badge *'and'* was selected so that we could see what happened when the badge really was not meaningful.

Once we had the list of badges, we chose $18,425$ unique users who carry the badges in their profile description and queried the Stream API, asking specifically for those users' tweets. We followed this set of users for 6 days, while sending queries to the REST API in order to get more specific user information such as 'Who does this user follow?'. Since the REST API requests are rate-limited, we needed several days to collect the follower information for all of the users. The process provided us with $1,669,470$ unique tweets and $241,311$ unique actions. For experiments, we narrowed this down to actions which are not too sparse, and only included the users who performed actions during collection. We stored the data in text files and processed it using Python. To scale our work to larger data sets, we would move to a database format instead, and find an alternative way to extract the popular celebrities each user follows, as the REST API was a considerable bottleneck.

3.2 Code

In implementing the sampler, we performed several steps to ensure the numerical accuracy of our results. For the sampling of $b_i^{(u)}$, we needed to normalize the values calculated before performing the actual sampling, so we used logarithms of the probabilities to avoid underflow, and subtracted the maximum log probability from both values before normalizing. Similarly, we used logarithms whenever possible in the Metropolis-Hastings steps to avoid underflow problems and to make the calculation of beta and gamma functions more stable.

We originally implemented the sampler in the R language. Unfortunately, once the code was completed, we realized that it was very slow, and further research confirmed that for sequential tasks such as MCMC samplers, R is notoriously slow. Due to lack of time to rewrite the entire sampler in another language, we attempted to optimize the R code in several ways. First, we used R's sparse matrix package to handle the action matrix, which is a large matrix of binary values and is indeed very sparse. Second, we used R's vectorization capabilities to eliminate loops and use vector or matrix operations whenever possible. Third, we used parallel processing when possible, though there were very few steps where we could do this due to the nature of Gibbs sampling. Finally, we rewrote two of the most time-expensive functions in C++ and called these functions from within the R code. While these optimizations did speed up the code considerably, we were still unable to

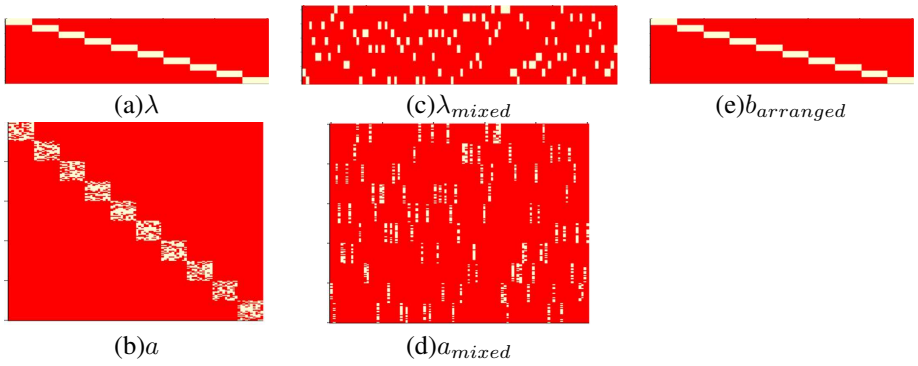


Figure 2: Artificial data input and output

run the sampler on our full set of collected data in time for this paper, and instead experimented on smaller subsets of our collected data.

4 Experimental Results

We performed several experiments to determine whether the sampler is working properly and to determine whether the results are meaningful in context. While we were restricted to smaller data sets than originally intended due to implementation problems, we still found some interesting results that we think would scale nicely to the larger data set. In all experiments, the number of badges/labels is 8, but the number of users and actions varies depending on the experiment (all on the order of 500).

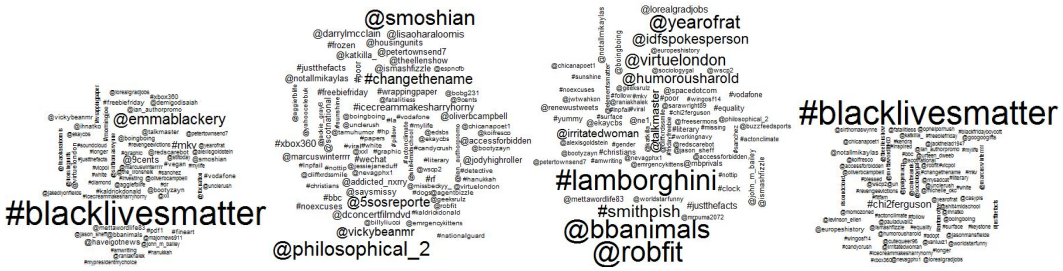
4.1 Artificial data model validation

In order to confirm that we implemented the sampler correctly, we applied it to an artificial data set that has 10 badges, 100 users, 200 actions, and designed λ, and a matrices as in Figure 2 (a) and (b). Then we mixed column of each matrix (as shown in (c) and (d)) and ran our algorithm. In order to interpret the results easily, we re-arranged the columns of the b matrix according to the order of how we mixed λ to get $b_{arranged}$, which is the result of badge assignment (shown in Figure 2 (e)). It clearly assigned badges that have same pattern as ground-truth λ matrix. From this artificial experiment, we were able to verify that in the simple scenario simulated the sampler is performing as intended.

4.2 Associating badges with actions

Here we analyze the output of the sampler for various conditions with a focus on determining which actions were associated with each badge. We looked at a matrix $Actionfreq_{ij} = \phi_{ij} \times s_{ij}$, which shows the probability of action j taken from badge i including the sparsity mask. We constructed distribution of actions using values of $Actionfreq_{i,:}$ for each i and generated word clouds for each badge. First, we used only hashtags (#) and retweets (@) as the observed actions. Selected word cloud examples for badge 'and', 'rock', 'conservative', and 'feminist' are shown in Figure 3. The word cloud for badge 'and' demonstrates actions that are common to randomly assigned users. While we gathered data, the most dominant action taken among a broad portion Twitter users was *#blacklivesmatter*, as we gathered data when the Ferguson grand jury was deliberating. For badge 'rock,' actions of hashtags and retweets showed noisy patterns, likely due to multiple interpretations of the word. For the 'conservative' badge, related actions were distinct compared to other badges. The profile of Twitter accounts *robfit*, and *idkspokesperson* are shown in Figure 4 and appear to have 'conservative' properties. For the 'feminist' badge, *#blacklivesmatter* was the one and only dominant action taken, perhaps because feminists are concerned with social justice.

In Figure 5 we add the 'following' action (indicated in $\{\cdot\}$). It appears the word clouds are nearly taken over by the 'following' actions, which supports our hypothesis that a 'following' action takes an important role in explaining each badge, perhaps mores than 'hashtag' or 'retweet' actions, al-



(a) 'and' (b) 'rock' (c) 'conservative' (d) 'feminist'

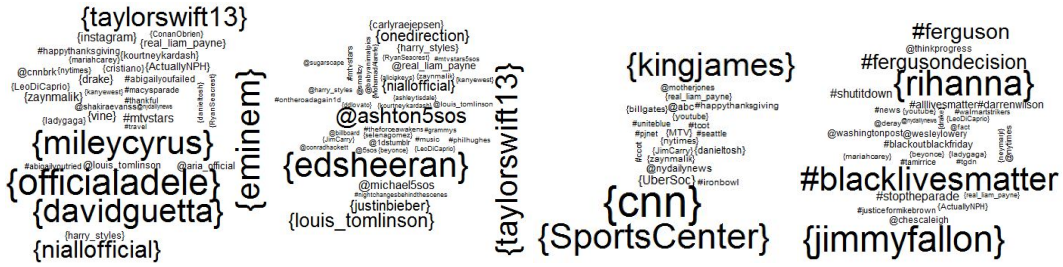
Figure 3: Wordclouds for selected bages. (Actions of hashtags and retweets)

(a) (b)

Figure 4: Profile of Twitter account (a) *robfit*, and (b) *@IDFSpokesperson*

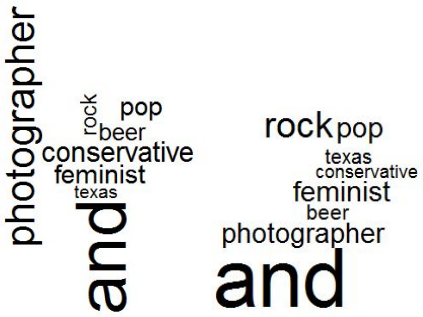
though this should be taken with a warning: the 'following' part of the matrix was less sparse than the 'hashtag' or 'retweet', as this is a simple action for users to share. When we look at each word cloud in detail, we can observe differences from Figure 3. For 'and,' we cannot relate the dominant action taken among Twitter users with the most prevalent current event of the time (Ferguson). This makes sense because while many tweeted about Ferguson, few changed their 'following' behavior based on it. Therefore we see the cloud dominated by some of the most popular twitter accounts. For badge 'rock,' we see more badge-related actions compared to Figure 3. Perhaps rock lovers are more likely to follow a rock star than mention the word 'rock' in a profile. For the 'conservative' badge, we see following *cnn* which makes sense for this group. It was interesting to see how following *SportsCenter* and *kingjames* (Twitter account for LeBron James, a basketball player) appeared together, though it is unclear why this is related to conservatism. For the 'feminist' badge, the word cloud is similar to Figure 3 (d) but arguably more informative. *#blacklivesmatter* was one of the dominant actions, but other actions such as following $\{jimmyfallon\}$ and $\{rihanna\}$ also shown to be popular actions taken among users with 'feminist' badge.

Further, we wanted to determine the extent to which an action of following a certain person can explain the badge of that person by examining whether we can infer profile words by observing actions taken by the user. We chose actions $\{cnn\}$ and $\{taylorswift13\}$, and obtained a list of users who took each action. Then, we observed the distribution of these users' actual profile words or badges, shown in Figure 6. In both cases, the word 'and' dominates but is not meaningful in this context. For (a) $\{cnn\}$, after ignoring 'and', we see that badges 'conservative' and 'photographer' are the most frequently used profile words for users who follow $\{cnn\}$, so users taking action of $\{cnn\}$ are likely to have profile words 'photographer' or 'conservative.' For (b) $\{taylorswift13\}$, we can again ignore 'and', and see that 'rock' dominates the profiles of people



(a) 'and' (b) 'rock' (c) 'conservative' (d) 'feminist'

Figure 5: Wordclouds for selected badges with actions of hashtags, retweets, and following. The "follow" action is indicated by $\{\cdot\}$.



(a) $\{cnn\}$ (b) $\{taylorswift13\}$

Figure 6: Distribution of profile words of users who took a certain action

who follow $\{taylorswift13\}$. Comparing with the word cloud for 'rock', it appears the model is truly learning relationships between actions and badges successfully.

4.3 Prediction of badges

While one of the strengths of the model is to associate actions with badges, we were also interested in the predictive power of the model. We first analyzed the extent to which users are assigned badges that directly correspond to labels in the user's profile. By averaging the sampled $b_i^{(u)}$ values for user u and badge i over many trials, we obtain a proportion that shows how many times the badge was assigned to the user during sampling which we threshold to determine which badges we officially assign to a user based on the estimated mean of the posterior distribution. Table 2 shows the results for several different runs of the sampler, confirming that the model does generally correctly assign badges if the user has the word in the profile, though the sample size affects accuracy.

While the model can (and does) assign labels that are not explicitly in a user's profile, we have no way of assessing whether these badges are correct other than by inspection because there is no ground truth. We did inspect some of the results by hand and found that often an assigned (but non-

Iterations	Burn-in	Includes followers?	Number of users	Number of actions	p
30	20	No	295	189	93%
30	10	No	327	229	97%
40	20	No	295	189	98%
50	20	No	295	189	99%
100	10	Yes	400	336	99%
300	50	Yes	400	336	100%

Table 2: Percent of badges p predicted correctly (compared to observed labels) with threshold 0.9 for the posterior mean for the badges.

Iterations	Burn-in	Includes followers?	Number of users	Number of actions	Threshold	p
40	20	No	295	189	0.8	5%
40	20	No	295	189	0.5	27%
100	10	Yes	400	336	0.8	19%
100	10	Yes	400	336	0.5	88%
300	50	Yes	400	336	0.5	11%
300	50	Yes	400	336	0.8	84%

Table 3: Percent of badges p predicted correctly compared to held-out labels.

labeled) badge did appear appropriate for the user in question. To quantify this, we took the existing labels from user profiles and randomly deleted 1/20 of the labels for each badge, then determined how well the sampler could assign badges reflecting the deleted labels, given only the user's actions and the association of actions with the badge due to other users who had the label in the profile.

The results are summarized in Table 3. The percentages represent how often the badge corresponding to the removed label was assigned to the user. We determined badge assignment by thresholding the posterior mean at two different levels, 0.8 and 0.5. An alternative approach would be to rank the badges by posterior mean probability and select badges in order. While higher accuracy is desired, several possible explanations exist for this behavior. First, randomly deleting some labels could have complex impacts on the small data set, such as some users ending up with no labels or some actions ending up with critically low numbers of occurrences. Second, the model is designed to keep the association of actions with badges sparse, so we expect it to be fairly difficult for the model to assign a badge to a user based on actions alone (and in fact this is confirmed by inspection). Third, the badges we selected may not be distinctive enough to be selected consistently based only on actions, or some of the users may not be very active, making prediction for those users nearly impossible.

5 Discussion

Based on experiments, we see our implementation of the badge model performed as expected in relating actions to badges and performed similarly to the original badge model in predicting held-out badges. Our results suggest that running the sampler for longer periods of time on larger data sets would yield better results. In addition, we believe that collection of more data so that we could cull specifically for very active users would also help to improve the model's performance, particularly in prediction tasks.

While the relationships between actions and badges discovered by this model are intrinsically interesting, we are especially interested in the differences between the cases when including 'following' actions versus including only retweets and hashtags. Based on the experiments, it appears that 'following' actions certainly do provide more information that helps to explain badges. We posit that this is because some attributes are more easily expressed through short-term actions (such as retweets or hashtags) and some are expressed better through long-term actions (such as following an elite user). Based on our work, a combination of this information offers the most promise for making better predictions about user attributes in this type of model.

Some challenges that can be addressed in this area include expediting the collection of follower information, as the rate-limited API makes the data collection quite slow, and this may be one reason previous works have not generally used this kind of information. In addition, modifying the model to account for the difference between short-term and long-term actions (perhaps by adding additional nodes to the graph) could enhance the ability of the model to give appropriate weight to the different kinds of actions. Finally, a challenge that we considered but were unable to address is the same as presented by El-Arini et al, which is to find a better method for selecting badges so that they are meaningful, descriptive, and account for similar categories of users.

References

[AGHM12] Sanjeev Arora, R Ge, Y Halpern, and David Mimno. A practical algorithm for topic modeling with provable guarantees. *arXiv preprint arXiv: ...*, 28, 2012.

[BL06] D Blei and J Lafferty. Correlated topic models. *Advances in neural information processing systems*, 2006.

[BNJ03] DM Blei, AY Ng, and MI Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[CHBG10] Meeyoung Cha, H Haddadi, F Benevenuto, and PK Gummadi. Measuring User Influence in Twitter: The Million Follower Fallacy. *ICWSM*, 2010.

[EAPH12] K El-Arini, Ulrich Paquet, and Ralf Herbrich. Transparent user models for personalization. *Proceedings of the 18th . . .* , pages 6–14, 2012.

[EAXFG13] K El-Arini, Min Xu, EB Fox, and Carlos Guestrin. Representing documents through their readers. *Proceedings of the 19th ACM . . .* , 2013.

[KLPM10] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is Twitter, a social network or a news media? *. . . of the 19th international conference on . . .* , pages 591–600, 2010.

[LM06] W Li and A McCallum. Pachinko allocation: DAG-structured mixture models of topic correlations. *Proceedings of the 23rd international conference on . . .* , 2006.

[RDL10] Daniel Ramage, ST Dumais, and DJ Liebling. Characterizing Microblogs with Topic Models. *ICWSM*, 2010.

[RH09] Daniel Ramage and David Hall. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. *Proceedings of the 2009 . . .* , (August):248–256, 2009.

[WHMW11] Shaomei Wu, Jake M. Hofman, Winter a. Mason, and Duncan J. Watts. Who says what to whom on twitter. *Proceedings of the 20th international conference on World wide web - WWW '11*, page 705, 2011.

[ZGS10] Elena Zheleva, Lise Getoor, and S Sarawagi. Higher-order graphical models for classification in social and affiliation networks. *NIPS Workshop on Networks Across . . .* , pages 1–7, 2010.