

Identifying Event Context using Anchor Information in Online Social Networks

Hansu Gu, Mike Gartrell, Liang Zhang, Qin Lv, Dirk Grunwald
Department of Computer Science, University of Colorado Boulder, Boulder, CO 80309-0430 USA
{hansu.gu, mike.gartrell, liang.zhang-2, qin.lv, dirk.grunwald}@colorado.edu

May 17th, 2013

ABSTRACT

Online social networks (OSNs) such as Twitter provide a good platform for event discussions. Recent research [26] [25] has shown that event discussions in OSNs are diverse and innovative and encourage public engagement in events. Although much research has been conducted in OSNs to track and detect events, there has been limited research on detecting or understanding the *event context*. Event context helps to better predict users' participation in events, identify relations among events, and recommend friends who share similar event context.

In this work, we have developed *AnchorMF*, a matrix factorization based technique that aims to identify event context by leveraging a prevalent feature in OSNs, the *anchor* information. Our AnchorMF work makes three key contributions: (1) a formal definition of the event context identification problem; (2) anchor selection and incorporation into the matrix factorization process for effective event context identification; and (3) demonstration of applying event context for user-event participation prediction, relevant events retrieval, and friendship recommendation. Evaluation based on 1.1 million Twitter users over a one-month data collection period shows that AnchorMF achieves a 20.0% improvement in terms of user-event participation prediction.

1. INTRODUCTION

With the rapid growth of online social networks (OSNs), more and more real-world events are being discussed on Web 2.0 platforms such as Facebook, Twitter, Tumblr, etc. Researchers have been using these platforms as social sensors to detect events, analyze event-related discussions, and predict event popularity. Despite much research on the aforementioned topics, there has been limited research that aims to detect or understand the *context* of events. The context for an event is essentially represented by the group of users who show inherent interests or willingness to participate in the event, such as people supporting their home football team, residents affected by a local fire or flooding, or people interested in Oscar nominations. The aggregated attributes of the group typically demonstrate commonalities in location, interests, age, gender, etc.

Event context identification is an important research problem and has many real-world applications. Successful event context identification will help to better predict the users who are going to participate in an event, thus creating value for enterprises and organizations for better marketing and event management. Event context also helps to identify relations among events if they share the same or similar context. Interesting patterns may be discovered even if events are not semantically related but otherwise share similar context. For example, as we will show in the experiments, event *Obama 2013 inauguration* is related to event *The International Consumer Electronics Show (CES) 2013* according to identified context. Another application is friendship recom-

mendation based on the event context for past event participation. As we will later show in the experiments described in Section 5, friendships are correlated with event context similarity among users.

Event context identification is a challenging problem for several reasons. First, it is difficult to define event context properly. Context is a subjective concept and the same group of users may be interpreted according to different common features. It is typically easier for a computer algorithm to discover a contextual pattern, rather than explain the cause for this pattern. Second, although other techniques may be applied to solve the event context identification problem, their performance is not good [24] [11]. Given historical event data, we can extract event contexts by characterizing events based on user participation, and at the same time characterizing users by their event participation. This process is very similar to the idea of *matrix factorization* [24]. Previous research mainly considered the original user-item rating matrix (i.e., the user-event matrix in our setting) and friendship information if available. However, as we show later in our experiments, friendship information does not show significant performance improvement. Finally, users interested in certain types of events tend to follow certain *anchor* accounts in OSNs. However, it is not clear how these anchor accounts can be selected (among massive following/follower relations), nor is it clear how to incorporate such anchor information into the overall event context identification process.

To address these challenges, we have developed *AnchorMF*, a unified solution for identifying event context by utilizing both user-event participation information and anchor information in OSNs. Given observations of user-event and user-follower matrices, a probabilistic model is built to consider users, events, and anchors as latent factors. An anchor selection algorithm is proposed to automatically identify informative anchors for the model. Finally, a Gibbs sampler and a maximum a posteriori (MAP) estimator are proposed to estimate the parameters of the model. AnchorMF is implemented and evaluated using a real-world Twitter data set which we have collected over one month and contains 1.1 million Twitter users. Evaluation results show that AnchorMF outperforms state-of-art techniques by 20.0% in terms of prediction accuracy. AnchorMF can identify relevant events using an information retrieval process. We also show that event contexts can be used for friendship recommendation.

To the best of our knowledge, this is the first work that aims to address the event context identification problem. This paper makes the following contributions: (1) a formal definition of the event context identification problem; (2) anchor selection and incorporation into the matrix factorization process for effective event context detection; and (3) application of event context to user-event participation prediction, relevant events retrieval, and friendship recommendation.

2. RELATED WORK

There has been much event-related research in the literature. The field of *event detection and tracking* can be traced back to [32] [1] [31]. Kleinberg defined and extracted bursts of activity from emails using an infinite-state automaton [13]. Bursty events can also be detected from news texts by identifying bursty features with a binomial distribution model and threshold-based heuristics [4]. Ihler et al. focused on time-series data such as logs and proposed Markov-Poisson models to detect anomalous events [10]. A general probabilistic model was proposed to extract correlated bursty topic patterns in [28]. Chen et al. used user tag information to identify events which involve browsing and searching photos on Flickr [2]. Lappas et al. explored how bursty terms help enhance the search process [16]. These works show the importance and effectiveness of event analysis using Web data.

More research has been conducted on Twitter recently. Different crisis events have been analyzed to identify generative and innovative properties of discussion on Twitter [15] [26] [25]. Sakaki et al. developed an earthquake alarm system by extracting real-time earthquake events on Twitter [22]. Petrović et al. presented a locality sensitive hashing approach to efficiently detect events that have not been seen before based on tweets [21]. Weng et al. proposed wavelet-based signal clustering on Twitter text stream data to detect events [30]. Lin et al. leveraged interests of users and social relations to track the evolution of popular events [17]. Event popularity can be predicted by considering a variety of social features [7]. Such event detection techniques support algorithmic discovery of events on OSNs, and help to build the foundation of event-related research. However, they do not solve the event context identification problem directly.

This work also builds upon existing matrix factorization techniques. Salakhutdinov et al. proposed a probabilistic matrix factorization model, which factorizes the explicit user-item matrix to a user latent trait matrix and a item latent trait matrix [24]. A full Bayesian version of the model was also proposed to provide generalized parameter tuning and avoid overfitting [23]. For implicit datasets, Hu et al. adopted more features from the original user-item matrix and proposed an improved gradient descent method to solve the problem more efficiently [8]. More recent research considers friendship information as a useful feature to incorporate into the current framework. Trust based approaches consider friendship as trust to influence users' latent factors. In [18], friendship information was modeled as a linear combination of the basic model. Another approach was also proposed to model friends as a separate latent matrix and used friendship as observations [19]. SoicalMF was proposed to incorporate friendship into the same latent space as users and a user's latent factor is represented as the average of all friends' [11] latent factors. Gartrell et al. proposed to consider only close friends when combining friends' latent factor and used a Markov random field to aggregate latent factors [5]. Influence based models consider users' interests to be influenced by their friends. Huang et al. considered receiver interests, item qualities and interpersonal influences for final recommendation [9]. Jiang et al. incorporated interpersonal influences into the existing PMF model and showed significant performance improvement [12]. The assumption of influence-based models is that items must be coming from their friends, which is not always the case. Our work sep-

arates the anchor latent factor space from the user latent factor space with a feature selection process, which shows better performance than existing solutions.

3. PROBLEM FORMULATION AND SYSTEM OVERVIEW

3.1 Definitions and Problem Formulation

Each **event** $e = \{m_1, m_2, \dots\}$ is represented by a set of messages obtained by searching for specific keywords $W = \{w_1, w_2, \dots\}$ in an OSN (e.g., Twitter) and corresponds to a real-world event. Each message $m_i = \langle u_i, t_i \rangle$, meaning that the message was posted by user u_i at time t_i . u_i is considered a "participant" of event e in the cyber world.

The **context** of a given event is defined as a group of users who participate in the event because of some inherent reasons, i.e., common attributes of the participants or latent event/user factors. For instance, both location and interest are important attributes to represent event context: the context of a local basketball game could be the group of local people who like their basketball team. Therefore, the context of event e_j can be jointly characterized by the event latent factor E_j and the set of user latent factors U_{e_j} of all the participants of e_j .

Anchors are popular users or public pages in OSNs, and their followers tend to participate in certain types of events, e.g., the Twitter account of a local news venue or a user posting actively on a specific topic. Usually, anchors are not directly identified by OSNs, and any user who has followers can be an anchor candidate. Selecting anchors for effective event context identification is the key. Let \mathbb{U}_a be the set of followers of anchor candidate a , we select a as an anchor based on the following two factors:

1. $|\mathbb{U}_a| \geq \text{threshold}$, i.e., the anchor must have at least *threshold* followers. *threshold* is set to 269 based on our modeling analysis shown in Section 4.
2. The probability of a being an anchor depends on a 's concentration of events \mathbb{E} , i.e., \mathbb{U}_a participate in similar events. This probability is used as a weight in the model to reflect the impact of this anchor candidate.

The problem of **event context identification** is then defined as follows. Given M events $\mathbb{E} = \{e_1, e_2, \dots, e_M\}$ participated by N users $\mathbb{U} = \{u_1, u_2, \dots, u_N\}$, the output of event context identification is $C = \{c_1, c_2, \dots, c_M\}$, where each c_i is the context of e_i . The event contexts capture the event latent factors and user latent factors, which in turn can identify the subset of users who are likely to participate in each event. The success of event context identification can be evaluated by comparing the user-event participation predicted by the event contexts and the actual user participation in events. Detailed evaluation results are presented in Section 5.

3.2 System Overview

Figure 1 illustrates the high-level process of *AnchorMF* for event context identification. Given a set of events, we first select anchors from the candidate users (Section 4.1), then incorporate the selected anchors into an extended probabilistic matrix factorization (PMF) model (Section 4.2), and finally through model inference (Section 4.3) we obtain the event contexts represented by event and user latent factors.

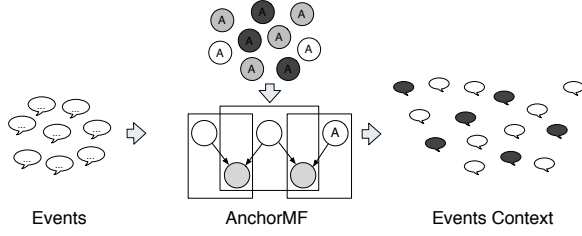


Figure 1: AnchorMF system overview.

3.3 Preliminary: PMF

Given a set of N users $\mathbb{U} = \{u_1, \dots, u_N\}$, a set of M events $\mathbb{E} = \{e_1, \dots, e_M\}$, and the binary matrix $R = [R_{ui}]_{N \times M}$ representing users' participation in events, the probabilistic matrix factorization (PMF) model factorizes R into two latent matrices $U \in \mathbb{R}^{K \times N}$ and $E \in \mathbb{R}^{K \times M}$, representing k -dimensional latent trait vectors for users and events. The graphical model is shown in Figure 2.

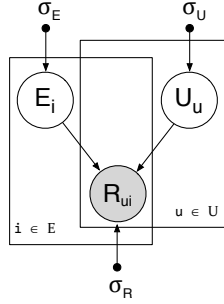


Figure 2: Probabilistic matrix factorization (PMF).

PMF defines the following distributions:

$$\begin{aligned}
 p(R|U, E, \sigma_R^2) &= \prod_{u=1}^N \prod_{i=1}^M \mathcal{N}(R_{u,i} | U_u^T E_i, \sigma_R^2) \\
 p(U | \sigma_U^2) &= \prod_{u=1}^N \mathcal{N}(U_u | 0, \sigma_U^2 \mathbf{I}) \\
 p(E | \sigma_E^2) &= \prod_{i=1}^M \mathcal{N}(E_i | 0, \sigma_E^2 \mathbf{I})
 \end{aligned} \tag{1}$$

4. THE ANCHORMF MODEL

4.1 Anchor Selection

As discussed in Section 3, anchors are any user accounts which have at least a certain number of followers and whose followers show a good concentration on similar events. Using a real-world Twitter dataset we have collected (Section 5.1), we start with anchor candidates with at least 1 follower, and the set of candidate anchors shrinks as the selection process progresses. For simplicity, we refer to the anchor candidates in each round as anchors.

4.1.1 Anchor and User Distribution

We first need to understand the relation between anchors and their followers. The problem can be decomposed into the distribution of followers given anchors and the distribution of anchors given followers.

The red curve in Figure 3 shows the complementary cumulative distribution function (CCDF) of the number of followers given anchors. The x-axis is the number of followers

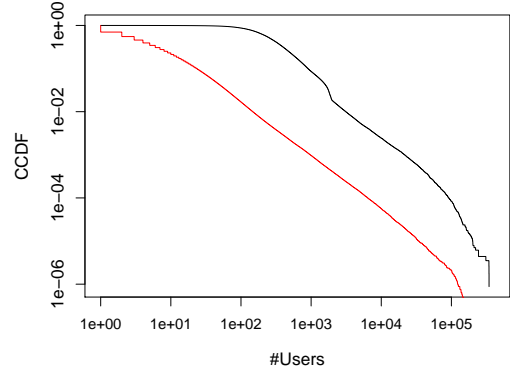


Figure 3: Anchor and user distribution.

# pairs (M)	16.32	7.86	4.64	3.04	3.69	2.81	4.25
#anchors	1	2	3	4	5.42	7.84	19.68
#events	2.47	2.68	2.82	2.92	3.02	3.13	3.35

Table 1: Number of user pairs and their average number of shared anchors and shared events.

and the y-axis is the percentage of anchors. This heavy-tailed distribution shows that many anchors are followed by few users and very few anchors are followed by many users. The black curve in Figure 3 shows the CCDF of the number of anchors that users follow. Here, the x-axis is the number of anchors and the y-axis is the percentage of users. This black curve is also a heavy-tailed distribution and shows that many users follow few anchors and very few users follow many anchors. We notice that the black curve has a flat beginning; this indicates that users tend to have a minimum numbers of anchors to follow, which is approximately 100 as we can see from the figure. We also notice that there is an anomaly near 2,000 followers for the black curve. This is likely due to the fact that Twitter's policy [27] allows each user to follow at most 2,000 anchors unless he/she is very active on Twitter. As we can see from the figure, less than 5% of the users in our dataset follow more than 2,000 anchors. The gap between the red and black curves is caused by the fact that when counting the number followers of anchors, we only consider the users in our event dataset, and not all followers of the anchors at Twitter.

We use the goodness-of-fit based method proposed in [3] to fit the two CCDFs shown in Figure 3. The power law model gives us two parameters α and x_{min} . α is the scaling parameter, which indicates how skewed the distribution is (the slope of the CCDF). As described in [3], a typical value of α is between 2 and 3. Our estimated α is 2.24 for the anchor distribution and 2.26 for the user distribution. These results match what we see in Figure 3 and the model shown in [14], which indicate that our dataset is representative. The second parameter, x_{min} , indicates the minimum x-axis value that fits the power law. The x_{min} of the anchor distribution is 269, and we use this number as the minimum frequency of an anchor candidate. Therefore, all the anchor candidates must have at least 269 followers. With this parameter setting, the number of anchor candidates is reduced to less than 1% of the original anchor candidate set size, which significantly reduces the amount of computation in our modeling process.

4.1.2 Relation Between Anchors and Events

Before considering the event concentration of anchors, we

#users	Pearson Correlation	Spearman Correlation
10,000	0.11	0.12

Table 2: Correlation between #anchors and #events per user.

first study the relation between anchors and events, specifically, if users who follow the same anchors tend to participate in the similar events. We randomly sampled 10,000 users from our dataset, and consider for each pair of users the number of shared anchors and number of shared events. We separate all the user pairs into different buckets based on quantile and ensure that all user pairs with the same number of shared anchors fall into the same bucket. Table 1 shows the aggregate results for each bucket, including the number of user pairs, average number of shared anchors, and average number of shared events. As shown in the table, when users share more anchors, the number of shared events also increases. Therefore, identifying the appropriate anchors can serve as good indicators for event participation and event context identification.

We further analyze for each user if the number of anchors he/she follows is correlated with the number of events he/she participates in. We use both Pearson’s correlation to check linear correlation and Spearman’s correlation to check non-linear correlation. The formulas are shown in Eq. 2.

$$r_{A,E} = \frac{E[(A - \mu_A)(E - \mu_E)]}{\sigma_A \sigma_E}, \rho_{A,E} = \frac{E[(a - \mu_a)(e - \mu_e)]}{\sigma_a \sigma_e} \quad (2)$$

As shown in Table 2, there is very little correlation between a user’s number of anchors and number of events. These results indicate that anchor information and event information are independent signals, and adding anchor information on top of user-event information can potentially boost the performance of event context identification.

4.1.3 Event Concentration and Anchor Weight

Based on the anchor-user distribution analysis, we prune anchor candidates with fewer than 269 followers. Next, we need to select candidates which show a good concentration of similar events. We solve this problem by first looking at the users who follow an anchor and the events that those users participate in. We compute an anchor-event matrix by multiplying the anchor-user and user-event matrices:

$$M_{ae} = M_{au} \times M_{ue} \quad (3)$$

Each element $N_{ki} = M_{ae}[k, i]$ is the number of anchor k ’s followers who participate in event i . We denote E' as the set of events participated by anchor k ’s followers, and each event i is duplicated N_{ki} times in the set, i.e., $E' = \{e_{1,1}, \dots, e_{1,N_{k1}}, \dots, e_{M,N_{kM}}\}$. We then need to consider whether the events in E' are similar to each other. The event concentration for each anchor k is defined as:

$$w_k = \sum_{i=1}^M \binom{N_{ki}}{2} \cdot 1 + \sum_{i=1, j \geq i+1}^M N_{ki} \cdot N_{kj} \cdot S(i, j) \quad (4)$$

The formula above aims to compute the average pair-wise event similarity for events in E' , which is used to represent the anchor’s concentration over events. If a pair contains two of the same events, the similarity is 1, otherwise, the similarity is defined by $S(i, j)$:

$$S(i, j) = \int_{-\infty}^{i^T j} \mathcal{N}(i^T j; 0, 1) d(i^T j) \quad (5)$$

The similarity function is the cumulative normal distribution of the inner product space given the i and j event pairs [?]. The intuition is that the similarity should be defined in the inner product space and lie between 0 and 1. Based on the similarity function and each anchor’s event concentration, we can then select anchors and proceed with incorporating the anchor information in the AnchorMF model.

4.2 Incorporating Anchor Information into the PMF Framework

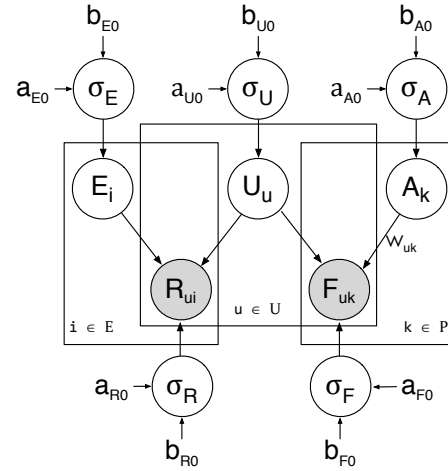


Figure 4: AnchorMF graphical model.

We incorporate anchor information into the PMF framework by factorizing the user-anchor matrix and also considering the importance of the anchors. As illustrated in Figure 4, the AnchorMF model considers a new observation F , which indicates what anchors each user follows. Correspondingly, we add a latent factor A to represent the anchors’ latent influence on users. According to Equation 4, each anchor k has a weight, and weight is the same for every user u , denoted as W_{uk} . Consistent with the PMF model, we also add priors and hyper-priors into the model. U and E are latent variables for users and events, respectively, and R is a binary observation matrix where each element indicates whether or not a user participates in an event. We derive Equation 6 directly from Figure 4.

$$\begin{aligned} & p(U, E, A | R, F, \sigma_R^2, \sigma_U^2, \sigma_E^2, \sigma_A^2, \sigma_F^2) \\ & \propto p(R|U, E) \times p(F|U, A) \times p(U) \times p(E) \times p(A) \\ & = \prod_{u=1}^N \prod_{i=1}^M \mathcal{N}(R_{u,i} | U_u^T E_i, \sigma_R^2) \\ & \times \prod_{u=1}^N \prod_{k=1}^P \left[\mathcal{N}(F_{u,k} | U_u^T A_k, \sigma_A^2) \right]^{W_{uk}} \\ & \times \prod_{u=1}^N \mathcal{N}(U_u | 0, \sigma_U^2 \mathbf{I}) \times \prod_{i=1}^M \mathcal{N}(E_i | 0, \sigma_E^2 \mathbf{I}) \\ & \times \prod_{k=1}^P \mathcal{N}(A_k | 0, \sigma_A^2 \mathbf{I}) \end{aligned} \quad (6)$$

To facilitate model inference, we also derive the log of the posterior probability as follows:

$$\begin{aligned}
& \ln p(U, E, A | R, F, \sigma_R^2, \sigma_U^2, \sigma_E^2, \sigma_A^2, \sigma_F^2) \\
&= -\frac{1}{2\sigma_R^2} \sum_{u=1}^N \sum_{i=1}^M (R_{u,i} - U_u^T E_i)^2 \\
&\quad -\frac{1}{2\sigma_R^2} \sum_{u=1}^N \sum_{k=1}^P w_{u,k} (F_{u,k} - U_u^T A_k)^2 \\
&\quad -\frac{1}{2\sigma_U^2} \sum_{u=1}^N U_u^T U_u - \frac{1}{2\sigma_E^2} \sum_{i=1}^M E_i^T E_i - \frac{1}{2\sigma_A^2} \sum_{k=1}^P A_k^T A_k + C
\end{aligned} \tag{7}$$

We denote $-\frac{1}{2\sigma_R^2}$ as λ_R , $-\frac{1}{2\sigma_U^2}$ as λ_U , $-\frac{1}{2\sigma_E^2}$ as λ_E , $-\frac{1}{2\sigma_A^2}$ as λ_A , and $-\frac{1}{2\sigma_F^2}$ as λ_F . We model $\{\lambda_R, \lambda_U, \lambda_E, \lambda_A, \lambda_F\}$ as conjugate Gamma distributions with flexible hyperpriors similar to:

$$p(\lambda_U) = \mathcal{G}(\lambda_U; a_{U0}, b_{U0}) = \frac{1}{\Gamma} b_{U0}^{a_{U0}} \lambda_U^{a_{U0}-1} e^{-b_{U0}\lambda_U} \tag{8}$$

4.3 Model Inference

Given the Bayesian framework defined in the previous section, inference for this model can be performed through Gibbs sampling [6]. Gibbs sampling generates a number of samples from an aperiodic and irreducible Markov chain, and involves sampling from the conditional distribution for each latent variable to approximate the joint distribution given that sampling from the joint distribution of the model is difficult. In the above model, we denote the random variables by $\theta = \{U, E, A, \lambda_U, \lambda_E, \lambda_A, \lambda_R, \lambda_F\}$, and we derive the following conditional distributions for all the random variables based on Equation 7.

λ_U is sampled from a Gamma distribution:

$$\begin{aligned}
& \lambda_U | \theta_{\setminus \lambda_U} \sim \mathcal{G}(\lambda_U; a_U, b_U) \\
& a_U = a_{U0} + \frac{|\mathcal{U}| K}{2} \\
& b_U = b_{U0} + \frac{1}{2} \sum_{i \in \mathcal{U}} \|\mathbf{U}_i\|^2
\end{aligned} \tag{9}$$

λ_E and λ_A are sampled from similar conditional distributions.

λ_R is also sampled from a Gamma distribution:

$$\begin{aligned}
& \lambda_R | \theta_{\setminus \lambda_R} \sim \mathcal{G}(\lambda_R; a_R, b_R) \\
& a_R = a_{R0} + \frac{|\mathcal{R}|}{2} \\
& b_R = b_{R0} + \frac{1}{2} \sum_{u,i \in \mathcal{R}} (R_{u,i} - U_u^T E_i)^2
\end{aligned} \tag{10}$$

U_u is conditionally sampled from a multivariate Gaussian distribution:

$$\begin{aligned}
& U_u | R, F, \theta_{\setminus U_u} \sim \mathcal{N}(U_u; \mu_u, \Sigma_u) \\
& \mu_u = \Sigma_u (\lambda_R \sum_{i=1}^M R_{u,i} E_i + \lambda_F \sum_{k=1}^P W_{u,k} F_{u,k} A_k) \\
& \Sigma_u = (\lambda_R \sum_{i=1}^M E_i \cdot E_i^T + \lambda_F \sum_{k=1}^P W_{u,k} A_k \cdot A_k^T + \lambda_U \mathbf{I})^{-1}
\end{aligned} \tag{11}$$

E_i is conditionally sampled from a multivariate Gaussian

distribution:

$$\begin{aligned}
& E_i | R, F, \theta_{\setminus E_i} \sim \mathcal{N}(E_i; \mu_i, \Sigma_i) \\
& \mu_i = \Sigma_i (\lambda_R \sum_{u=1}^N R_{u,i} U_u) \\
& \Sigma_i = (\lambda_R \sum_{u=1}^N U_u \cdot U_u^T + \lambda_U \mathbf{I})^{-1}
\end{aligned} \tag{12}$$

A_k is also conditionally sampled from a multivariate Gaussian distribution:

$$\begin{aligned}
& A_k | R, F, \theta_{\setminus A_k} \sim \mathcal{N}(A_k; \mu_k, \Sigma_k) \\
& \mu_k = \Sigma_k (\lambda_F \sum_{u=1}^N W_{u,k} F_{u,k} U_u) \\
& \Sigma_k = (\lambda_F \sum_{u=1}^N W_{u,k} U_u \cdot U_u^T + \lambda_U \mathbf{I})^{-1}
\end{aligned} \tag{13}$$

The Gibbs sampling approach described above computes an approximation of the posterior distribution, which allows us to infer users' participation in events, but it does not find the maximum point of the posterior. Therefore, it is difficult to compute a point estimate of the latent matrices U and E which result in the maximum function value from the Gibbs sampling results. However, U and E describe the event context we need, and thus we need good point estimate for these variables to calculate the similarities between users and events. To this end, we also propose a maximum a posteriori (MAP) estimation for the model which estimates the maximum point (mode) of the posterior distribution, and therefore generates point estimates for U and E . The MAP estimator empirically converges faster than the Gibbs sampling approach.

MAP estimation works by maximizing the conditional distributions of U and E iteratively, where:

$$\begin{aligned}
& U_u = \mu_u \\
& E_i = \mu_i \\
& A_k = \mu_k
\end{aligned} \tag{14}$$

Since all the conditional distributions are Gaussian distributions, the Gaussian's mean will define the curvature and how far to step towards the maximum point for each iteration. This approach is similar to the inference algorithm described in [20].

The complete inference algorithm is shown in Algorithm 1:

Although the sampling of U_u , E_i and A_k must be conducted sequentially, the sampling of different U s, E s and A s can be conducted in parallel. This saves significant computation time in practice. We implemented a parallelized Gibbs sampler and MAP estimator using the thread pool mechanism in the Java standard library. Empirically, the Gibbs sampler converges after 200 iterations with 50 burn-in samples and finishes within 1 hour. The MAP estimator converges within 100 iterations and finishes within 0.5 hours. The results are based on our own dataset described in Section 5.

5. EXPERIMENTAL EVALUATION

In this section, we evaluate AnchorMF, our proposed event context identification solution, using real-world events that we have collected. Our evaluation aims to answer the following questions:

Algorithm 1 AnchorMF

```

for  $t = 1$  to  $\text{num\_of\_samples}$  do
  if Gibbs sampling and  $t < \text{burn\_in\_samples}$  then
     $\{\lambda_U, \lambda_E, \lambda_A, \lambda_R, \lambda_F\} = \text{preset\_value}$ 
  else
    Sample  $\{\lambda_U, \lambda_E, \lambda_A, \lambda_R, \lambda_F\}$  (Eq. 9 and Eq. 10)
  end if
  for users  $u = 1$  to  $N$  do
    if Gibbs sampling then
      Sample  $U_u$  in parallel (Eq. 11)
    else if MAP then
      Use mean to compute  $U_u$  in parallel (Eq. 14)
    end if
  end for
  for events  $i = 1$  to  $M$  do
    if Gibbs sampling then
      Sample  $E_i$  in parallel (Eq. 12)
    else if MAP then
      Use mean to compute  $E_i$  in parallel (Eq. 14)
    end if
  end for
  for anchors  $k = 1$  to  $P$  do
    if Gibbs sampling then
      Sample  $A_k$  in parallel (Eq. 13)
    else if MAP then
      Use mean to compute  $A_k$  in parallel (Eq. 14)
    end if
  end for
end for

```

- Does AnchorMF provide good predictive performance for user participation in events?
- Is the identified event context interpretable?
- Is event context effective for retrieving relevant events?
- Is event context useful for friendship recommendation?

5.1 Experimental Setup

We collect data using the Twitter API. We monitor daily Twitter trending topics and get a list of ranked popular keywords by considering both how long they stay on the trending topics and their rank. Then a human review process is used to review the top 200 keywords and identify the ones that match real-world ongoing events. The selected keywords are then filtered on real-time Twitter streams to continue collection of messages which contain the keywords. At the same time, we search for historical tweets which contain the keywords for up to 7 days. Since the selected keywords are mostly filtered on the day they became popular, we believe a 7-day look-back window is enough to collect complete events based on keywords. The data collection process introduces some noise into the dataset, but we carefully choose representative keywords to ensure events are not too general. For example, the *2013 Obama inauguration* event was collected based on the keywords *Obama inauguration*, rather than *Obama*, which tends to have a much broader scope. After we collect all the desired events, we only consider users who have participated in at least 5 of these events. This procedure helps us to remove much of the noise in the dataset. We believe most of our data consists of complete and coherent events. After obtaining the users who participated in

Category	#Events
Sports	248
Entertain	134
Tech	17
Social	25
Politics	37

Table 3: Event categories.

Users	1.1M
Events	461
Anchors	0.59M
User-event pair	20.79M
User-anchor pair	175.99M
friendship pair	92.72M

Table 4: Data statistics.

each event, we also collect friends of the users, users’ profiles, and lists (a group of followed users with a group name). In total, over a one-month period of time from Jan 4th to Feb 3rd, 2013, we collected 461 events consisting of 20.79M tweets and 1.1M users. All the data are stored in MongoDB and the total volume of the data is 554 GB. Statistics for this dataset are shown in Table 4. From Table 3 we can see that events are divided mainly into five categories. Sports and Entertainment events dominate the event type distribution. The CDF of the number of users per event is shown in Figure 5. We see that our event dataset consists of both large and small events and event size follows a heavy-tailed distribution.

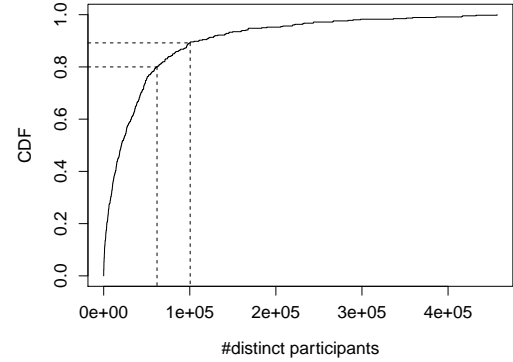


Figure 5: Event size distribution.

All the experiments have been conducted on a 2.4 GHz 16-core machine with 48GB of memory. This machine runs Ubuntu 12.04.2 and JVM 1.6.0.27. All of the implementation and experiments are written in Java.

5.2 Prediction Performance

In this experiment we want to examine the effectiveness of our event context identification in terms of predicting user participation in events. We run 10-fold cross validation and in each fold randomly select 10% of all the events in our dataset as test events and the remaining 90% as training events. For each test event, we sort all the users according to the time they participated in the event, and use the top 10%, 20%, 30%, 40%, or 50% as training users for the test event. Our goal is to evaluate the predictions of the other 90%, 80%, 70%, 60% and 50% of users for test events. We use average rank percentile [8] as our main evaluation metric. The users who actually participated in the events should be ranked highly among all users.

In Table 5, each row represents a method of ranking test users for a test event. These methods are:

1. *Random* ranking predicts testing users in a randomized order. This method is the baseline for prediction.
2. *Popularity* based ranking predicts test users who are popular in the training data as ranked higher for test

Training Users	10%	20%	30%	40%	50%
Random	0.5	0.5	0.5	0.5	0.5
Popularity	0.349	0.349	0.349	0.349	0.349
Baseline PMF	0.313	0.276	0.258	0.247	0.240
SocialMF	0.313	0.277	0.258	0.246	0.240
AnchorMF-Gibbs	0.213	0.204	0.197	0.196	0.193
AnchorMF-MAP	0.212	0.202	0.197	0.193	0.192

Table 5: Prediction performance comparison.

events. This method does not consider event context and is the baseline for contextual prediction.

3. *Baseline* ranking makes prediction based on event context identified by the PMF matrix factorization technique [24] using only user and event information. This method serves as the baseline for prediction based on event context.
4. *SocialMF* identifies event context based on the matrix factorization technique considering user friendship information. This method shows better performance than the baseline PMF approach [11].
5. *AnchorMF* is the model we propose to identify event context based on the matrix factorization technique that leverages anchor information. We compare the performance of AnchorMF using either the Gibbs sampler or the MAP estimator.

The average rank percentile evaluation metric that we use is a recall-based metric from [8], since the implicit dataset we use does not include complete data for precision based measurements. Users who actually participated in the events should be ranked higher in the prediction results. The average rank percentile is computed as follows:

$$\overline{rank} = \frac{\sum_e (\sum_u rank_{ue} / |u|)}{|e|}, \quad (15)$$

where $rank_{ue}$ is the average rank percentile for each user u in the event e . 0 represents the highest rank, while 1.0 represents the lowest rank.

As shown in Table 5, AnchorMF outperforms SocialMF by 20.0% when using 50% of the users as training users for the test events. As the percentage of training users decreases, we see a larger performance boost for AnchorMF compared to SocialMF, up to 32.2%. We also notice that for both SocialMF and baseline PMF, in the case where we only use 10% of the training data, the performance is almost as bad as the non-contextual popularity-based method. However, AnchorMF with 10% training data performs better than the best cases for both SocialMF and baseline PMF. This shows the effectiveness of the identified anchor information, and indicates that it is particularly helpful to identify event context and predict user participation in the early stage of an event. When we compare the SocialMF and the baseline PMF approaches, we do not see much performance difference for our dataset. One possible explanation, as we will see in Section 5.5, is that on Twitter users tend to have friends who are very dissimilar in terms of the latent trait space. Therefore the use of aggregated friends’ interests, as performed in SocialMF, may not be beneficial. Additionally, since we removed users who have participated in fewer than 5 events from our dataset, and since SocialMF has been

Location	P	Description	P	Tags	P
new orleans	0.42	sports	0.06	sports	0.08
LA	0.12	music	0.04	travel	0.05
Louisiana	0.05	world	0.03	politics	0.03
city	0.03	god	0.03	music	0.02
usa	0.02	football	0.03	entertain	0.01

Table 6: Event case study: #nola

Location	P	Description	P	Tags	P
pa	0.31	social	0.12	twibes	0.07
lancaster	0.12	manager	0.05	pa	0.06
harrisburg	0.12	business	0.05	social	0.04
pennsylvania	0.06	marketing	0.05	local	0.04
county	0.06	foodie	0.05	leader	0.04

Table 7: Event case study: #hbgsme

proved to be most effective for cold start users in recommender systems, SocialMF is not effective in our scenario since there are no cold start users.

5.3 Event Context Case Study

As shown above, our proposed event context identification algorithm is effective and outperforms other existing related approaches. We would now like to see how to interpret the identified context. The experiment described in this subsection examines three different scenarios, where each scenario has a different event context. The results show that the identified event context is interpretable and meaningful.

We select 6 events from all predicted results we get from the experiment described in Section 5.2. Each event consists of the predicted users for that event; the information for each user includes Twitter profile and list data. We aggregate the user information for each event and use this data to populate a table, as shown in Tables 6 through 11. Each column represents a source or dimension of user information that we will examine, including location, the self-provided user profile description, and tags from users’ list information. We extract all keywords from this aggregated user information and list the top five keywords ranked by probability of occurrence ($P = frequency_count / total_frequency$). We study the context of events by looking at these keywords and manually verify whether they have coherent semantic meaning.

5.3.1 Location Specific Event Context

First, we look at two cases where users discuss events on Twitter based on location. Table 6 shows results from an event about a local famous cafe that moved to a new location, which happened on Jan 17, 2013. As we can see from the results, the location dimension has a concentration of probability on the keywords *new orleans*, which matches the actual location of this event. The rest of the keywords, such as *LA* and *Louisiana* in the location dimension also have coherent meaning. Although *city* and *usa* are general location terms which do refer to a specific location, they have much lower probability compared with higher ranked keywords. If we look at both the description and tag dimensions, the keywords all have fairly low probability without much concentration, and they also lack coherent semantic meaning.

Table 7 shows results from an event about a local social club meetup in Harrisburg, Pennsylvania that happened on

Location	P	Description	P	Tags	P
new york	0.08	media	0.05	news	0.37
ca	0.08	news	0.04	media	0.13
usa	0.06	tech	0.04	tech	0.09
canada	0.04	writer	0.03	marketing	0.02
tx	0.03	marketing	0.02	business	0.02

Table 8: Event case study: #2013ces

Location	P	Description	P	Tags	P
new york	0.12	film	0.11	news	0.48
los angeles	0.06	tv	0.09	entertain	0.10
london	0.06	writer	0.09	tv	0.02
torronto	0.04	news	0.09	fashion	0.02
canada	0.03	movies	0.07	media	0.02

Table 9: Event case study: #oscarnoms

Jan 21, 2013. The results are very similar to what we see from the #nola event. The location dimension has a concentration and coherent meaning, while the tag dimension does not. We do see that the description dimension has the *social* keyword with higher probability. The reason for this is the type of the event is essentially a social event and people participating in the event are self-identified with the keyword “social”.

5.3.2 Interest Specific Event Context

We now look at two examples that are based on users’ interests. We focus on the description dimension and the tag dimension to see if the keywords extracted give us meaningful information.

Table 8 shows results from an event about the International Consumer Electronics Show from Jan 8 to Jan 11, 2013. As we can see from the results, the tag dimension has a concentration on news, media, and tech, which match the event’s semantic meaning. Also as expected, the location dimension shows a broad coverage of different locations and does not have a concentration as compared to location-specific events. However, we do not see significant concentration in the description dimension, although the top keywords have coherent semantic meaning. We will discuss this result further in Section 5.3.3.

Table 9 shows results from an event about the Oscar nominations which happened on Jan 10, 2013. The results show the same pattern as what we find in the International Consumer Electronics Show event.

5.3.3 Location and Interest Specific Event Context

Next, we look at two events that are both location and interest specific. Good examples of these types of events are local sports events. We will focus on all of the three dimensions to see if there are any interesting patterns.

Table 10 shows the results of NBA basketball game event

Location	P	Description	P	Tags	P
ca	0.15	sports	0.10	sports	0.32
los angeles	0.10	life	0.08	nba	0.08
california	0.05	love	0.08	basketball	0.03
tx	0.04	fan	0.06	fans	0.02
san antonio	0.03	music	0.03	lakers	0.02

Table 10: Event case study: #lakers

Location	P	Description	P	Tags	P
london	0.16	fan	0.10	football	0.14
uk	0.08	football	0.07	sports	0.13
England	0.07	sports	0.05	sport	0.07
manchester	0.05	united	0.04	soccer	0.05
liverpool	0.02	arsenal	0.04	friends	0.03

Table 11: Event case study: liverpool

that involved the Los Angeles Lakers vs. the San Antonio Spurs, which happened on Jan 9, 2013. The location dimension shows an interesting concentration on both Los Angeles and San Antonio, which are the expected locations. The description dimension shows a somewhat noisy results, but the sport keyword is apparent. The tag dimension shows good concentration and gives us confidence that this is indeed a local sports event.

Table 11 shows the results of an event involving two English soccer teams in the Premier League, Manchester United vs. Liverpool, which happened on Jan 13, 2013. Similar to the results of the previous example, the location and tag dimensions show the expected results, while the description is relatively noisy.

From all these event case studies, we find that it is relatively easy for humans to understand the event context by looking at the location and tag dimensions. This results from the fact that the location field is specifically designed for users to provide their location on Twitter, and most people tend to follow this rule. Tags are provided by users’ followers and they serve identification purposes, and so tend to include location and interest information. The results also indicate the usefulness of the textual data in these dimensions and may lead us to incorporate this information into our model in the future. We also notice that the description dimension is noisy for all three types of events, because self descriptions are very informal and users do not usually include their location and interest information in their self-provided profile description.

5.4 Retrieval of Relevant Events Based on Event Context

By looking at the common attributes of predicted users for some events, we can understand the meaning of event context. The next important issue to investigate is how we can use the identified context to better understand events. In this experiment, our goal is to demonstrate the feasibility of building an event-based search engine by leveraging event context information. The challenge here is that we do not consider the text of events and the relevance is only based on event context.

We have built a proof-of-concept system to evaluate the effectiveness of the relevant event retrieval process. We construct queries from all of the 461 events in our dataset with different number of events. We first randomly select 46 events as length-1 queries. Then 46 length-2 events are randomly selected by combining any two of the length-1 queries. Next, 46 length-3 events are randomly selected by combining any three of the length-1 queries. After this process, we have 138 queries in total. For each query, all of the returned results are assessed as either relevant or irrelevant; there are 42,733 labeled judgment pairs in total. Standard information retrieval evaluation metrics [29], including precision@3, precision@5, precision@10, Mean Reciprocal Rank (mRR)

	P@3	P@5	P@10	mRR	mAP
Q1	0.717	0.625	0.555	0.889	0.691
Q2	0.725	0.675	0.586	0.855	0.668
Q3	0.759	0.722	0.623	0.856	0.683
Q1-Ran	0.017	0.010	0.013	0.057	0.028
Q2-Ran	0.012	0.011	0.012	0.064	0.029
Q3-Ran	0.021	0.019	0.019	0.079	0.032

Table 12: Event retrieval ranking results.

and Mean Average Precision (mAP), are used to evaluate the results. The first three precision based metrics are considered good metrics for results returned from mobile devices or Web searches. mRR measures the rank of the first relevant result, and mAP considers recall as well as precision in the measurement.

For each query, we divide the query q into separate events. Each potential result i has a relevance score $S(i, j)$ according to Equation 5 given event j . Therefore, the relevance score RS_i is computed according to:

$$RS_i = \frac{\sum_j S(i, j)}{|q|} \quad (16)$$

The first three rows of Table 12 show three sets of contextual retrieval results based on length-1-to-3 queries. The last three rows show three sets of randomized retrieval results based on length-1-to-3 queries.

As we can see from Table 12, using event context, the retrieved results have very good top-K accuracy and very high performance for first relevant result retrieval. By considering recall, the mAP also shows high performance. We also see that queries with different length show very similar performance. This means contextual retrieval is consistent for queries of different lengths. Compared to randomized retrieval methods shown in the last three rows, contextual retrieval can return events that are much more relevant.

The results above show that the semantic relevance of events as labeled by a human. However, there are some cases where, although human may think two events are semantically unrelated, they share the same context. These cases show interesting results for relevant events that can not be captured by semantics. One good example that we found in our dataset is the *2013 Obama inauguration* event, which happened from Jan 19 to Jan 21, 2013 and the *The 2013 International Consumer Electronics Show (CES)* event happened from Jan 8 to Jan 11, 2013. At a first glance by human, these two events are completely unrelated. However, our model shows that these two events have a common context of people who like technology. The discovery of non-semantic but contextually relevant events is the unique benefit of event context identification.

5.5 Friendship and Event Context

To emphasize the importance of identified event context, we will show another possible application. In this experiment, we investigate the correlation of identified event context with user friendship. We will demonstrate that users who belong to similar event context are more likely to be friends. This observation allows us to build better friendship recommendation services.

We randomly sample 100,000 users and obtain all of their friends. For each user and friend pair, we calculate the user-friend similarity of their latent factor vectors based on the

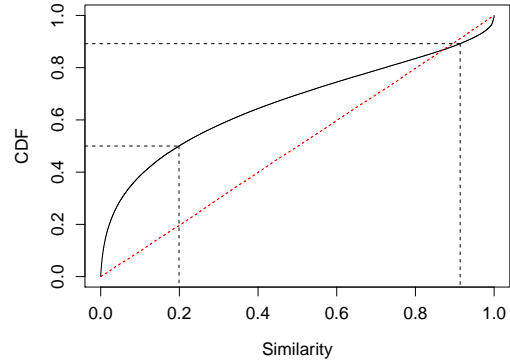


Figure 6: CDF of user-friend similarity based on event context.

similarity defined in Equation 5 that are generated from the matrix factorization. Let $Rank_f$ be the rank of a friend based on similarity, the relative similarity is calculated for each user as:

$$Rank_f / \#friends \quad (17)$$

The reason for defining the relative similarity is that the rank is important, and we want to normalize the rank. A similarity value of 1 indicates the most dissimilar user-friend pair.

We plot the CDF of all the user-friend similarity pairs in Figure 6. As we can see from the black line in Figure 6, users tend to have many similar friends. 50% of the friends are within 0.2 similarity. The red line shows the case where user-friend similarity is randomly calculated. At the end of the CDF curve, we do see there is a trend of increasing dissimilarity. This means users also have many dissimilar friends. This can affect performance when we consider the use of friends' interests to help predict users' interests; we will verify this further in the next experiment.

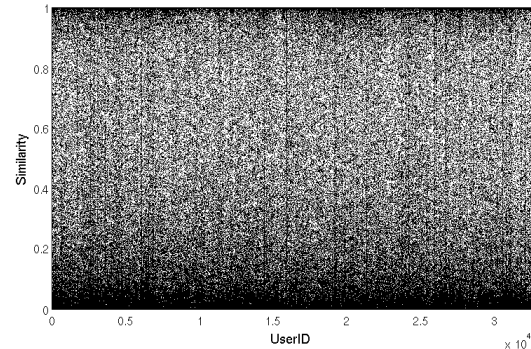


Figure 7: Friend similarity distribution per user.

The above analysis shows only aggregated similarity between users and their friends. In our second experiment, we want to see the friendship similarity distribution for each user. In Figure 7, the x-axis represents users, the y-axis indicates relative friendship similarity as defined above, and each dot represents a friend. As we can see from the figure, users have both similar and dissimilar friends. The dissimilar friends are likely to be loosely-connected social friends and they may not share common interests with the user. This also explains why a direct average of friends' interests, as performed in the SocialMF model, will not help infer users' interests. In reviewing all of the results from the analysis in this subsection, we see that better friend-

ship recommendation can be made based on event context information.

6. CONCLUSIONS

In this paper we have presented AnchorMF, a matrix factorization technique to solve the event context identification problem. AnchorMF selects anchors from users' followers and incorporates anchor information into an extended PMF framework. We have also presented several applications of using identified event context to predict users' participation in events, retrieve relevant events, and recommend friends based on event context. Our evaluation using real-world Twitter data shows that AnchorMF outperforms existing matrix factorization techniques by 20.0%. In our future work, we would like to explore other potential features, such as location information in users' profiles and tag information from users' followers, and consider how these features can be used in our model for better event context identification.

7. REFERENCES

- [1] J. Allan, R. Papka, and V. Lavrenko. On-line new event detection and tracking. *SIGIR '98*, pages 37–45.
- [2] L. Chen and A. Roy. Event detection from flickr data through wavelet-based spatial analysis. *CIKM '09*, pages 523–532.
- [3] A. Clauset, C. R. Shalizi, and M. E. Newman. Power-law distributions in empirical data. *SIAM review*, 51(4):661–703, 2009.
- [4] G. P. C. Fung, J. X. Yu, P. S. Yu, and H. Lu. Parameter free bursty events detection in text streams. *Vldb '05*, pages 181–192.
- [5] M. Gartrell, U. Paquet, and R. Herbrich. A bayesian treatment of social links in recommender systems. *CU Technical Report CU-CS-1092-12*, 2012.
- [6] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):721–741, 1984.
- [7] M. Gupta, J. Gao, C. Zhai, and J. Han. Predicting future popularity trend of events in microblogging platforms. volume 49 of *Proceedings of the American Society for Information Science and Technology*, pages 1–10.
- [8] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. *ICDM'08*, pages 263–272.
- [9] J. Huang, X.-Q. Cheng, J. Guo, H.-W. Shen, and K. Yang. Social recommendation with interpersonal influence. *ECAI '10*, pages 601–606.
- [10] A. Ihler, J. Hutchins, and P. Smyth. Adaptive event detection with time-varying poisson processes. *KDD '06*, pages 207–216.
- [11] M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. *RecSys '10*, pages 135–142.
- [12] M. Jiang, P. Cui, R. Liu, Q. Yang, F. Wang, W. Zhu, and S. Yang. Social contextual recommendation. *CIKM '12*, pages 45–54.
- [13] J. Kleinberg. Bursty and hierarchical structure in streams. *KDD '02*, pages 91–101.
- [14] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? *WWW '10*, pages 591–600.
- [15] V. Lampsos, T. De Bie, and N. Cristianini. Flu detector-tracking epidemics on twitter. In *Machine Learning and Knowledge Discovery in Databases*, pages 599–602. 2010.
- [16] T. Lappas, B. Arai, M. Platakis, D. Kotsakos, and D. Gunopulos. On burstiness-aware search for document sequences. *KDD '09*, pages 477–486.
- [17] C. X. Lin, B. Zhao, Q. Mei, and J. Han. Pet: a statistical model for popular events tracking in social communities. *KDD '10*, pages 929–938.
- [18] H. Ma, I. King, and M. R. Lyu. Learning to recommend with social trust ensemble. *SIGIR '09*, pages 203–210.
- [19] H. Ma, H. Yang, M. R. Lyu, and I. King. Sorec: social recommendation using probabilistic matrix factorization. *CIKM '08*, pages 931–940.
- [20] M. C. Mozer, B. Link, and H. Pashler. An unsupervised decontamination procedure for improving the reliability of human judgments. *NIPS '11*, pages 1791–1799.
- [21] S. Petrović, M. Osborne, and V. Lavrenko. Streaming first story detection with application to twitter. *HLT '10*, pages 181–189.
- [22] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. *WWW '10*, pages 851–860.
- [23] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. *ICML '08*, pages 880–887.
- [24] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. *Advances in neural information processing systems*, 20:1257–1264, 2008.
- [25] K. Starbird and L. Palen. (how) will the revolution be retweeted?: information diffusion and the 2011 egyptian uprising. *CSCW '12*, pages 7–16.
- [26] K. Starbird, L. Palen, A. L. Hughes, and S. Vieweg. Chatter on the red: what hazards threat reveals about the social life of microblogged information. *CSCW '10*, pages 241–250.
- [27] Twitter. Following rules and best practices. <https://support.twitter.com/articles/68916-following-rules-and-best-practices>, 2013.
- [28] X. Wang, C. Zhai, X. Hu, and R. Sproat. Mining correlated bursty topic patterns from coordinated text streams. *KDD '07*, pages 784–793.
- [29] W. E. Webber. *Measurement in information retrieval evaluation*. 2011.
- [30] J. Weng and B.-S. Lee. Event detection in twitter. In *Proceedings of the 5th International AAAI Conference on Weblogs and Social Media*, volume 3, 2011.
- [31] Y. Yang, T. Ault, T. Pierce, and C. W. Lattimer. Improving text categorization methods for event tracking. *SIGIR '00*, pages 65–72.
- [32] Y. Yang, T. Pierce, and J. Carbonell. A study of retrospective and on-line event detection. *SIGIR '98*, pages 28–36.