Understanding the Interaction between Interests, Conversations and Friendships in Facebook

Qirong Ho
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15217
qho@cs.cmu.edu

Rong Yan, Rajat Raina
Facebook
10 Hacker Way
Menlo Park, CA 94025
rongyan,rajatr@fb.com

Eric P. Xing
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15217
epxing@cs.cmu.edu

ABSTRACT

In this paper, we explore salient questions about user interests, conversations and friendships in the Facebook social network, using a novel latent space model that integrates several data types. A key challenge of studying Facebook's data is the wide range of data modalities such as text, network links, and categorical labels. Our latent space model seamlessly combines all three data modalities over millions of users, allowing us to study the interplay between user friendships, interests, and higher-order network-wide social trends on Facebook. The recovered insights not only answer our initial questions, but also reveal surprising facts about user interests in the context of Facebook's ecosystem. We also confirm that our results are significant with respect to evidential information from the study subjects.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; G.3 [**Probability and Statistics**]

General Terms

Algorithms, Experimentation

Keywords

Facebook data, user interest visualization, multi-view model, topic model, network model

1. INTRODUCTION

From blogs to social networks to video-sharing sites and still others, online social media have grown dramatically over the past half-decade. These media host and aggregate information for hundreds of millions of users, and this has sired an unprecedented opportunity to study people on an incredible scale, and over a broad spectrum of open problems. In particular, the study of user interests, conversations and friendships is of special value to the health of a social network ecosystem. As a classic example, if we had a good guess as to what a user likes (say, from explicit labels or conversations), we could serve her more appropriate content, which may increase her engagement with the media, and potentially help to obtain more structured data about her interests. Moreover, by providing content that is relevant to the user *and her friends*, the social network can increase engagement beyond mere individual content consumption — witness the explosive success of social games, in which players are rewarded for engaging in game activities with friends, as opposed to solitary play.

These examples illustrate how social networks depend on the interplay between user interests, conversations and friendships. In light of this, we seek to answer several questions about Facebook:

- How does Facebook's social (friendship) graph interact with its interest graph and conversational content? Are they correlated?
- What friendship patterns occur between users with similar interests?
- Do users with similar interests talk about the same things?
- How do different interests (say, camping and movies) compare? Do groups of users with distinct interests also exhibit different friendship and conversational patterns?

To answer these questions on the scales dictated by Facebook, it is vital to develop tools that can visualize and summarize user information in a salient and aggregated way over large and diverse populations of users. In particular, it is critical that these tools enable macroscopic-level study of social network phenomena, for there are simply too many individuals to study at fine detail. Through the lens of these tools, we can gain an understanding of how user interests, conversations and friendships make a social network *unique*, and how they make it *function*. In turn, this can shape policies aimed at retaining the special character of the network, or at enabling novel utilities to drive growth.

1.1 Key Challenges

Much research has been invested in user interest *prediction* [6, 4, 17, 13, 3], particularly methods that predict user interests by looking at similar users. However, existing works are mostly built on an incomplete view of the social media data, often solely restricted to user texts. In particular, the network itself acts a conduit for information flow among users, and we cannot attain a complete view of the social media by ignoring it. Thus, a deep, holistic understanding of user interests and of the network as a whole requires a perspective over diverse data modalities (views) such as text, network links and categorical labels. To the best of our knowledge, a principled approach that enables such capability has yet to be developed. Hence, our goal is to produce such a system for understanding the relationships between user interests, conversations and friendships.

In developing this system, at least two challenges must be properly addressed. For one, the data scale is unprecedented — Facebook has hundreds of millions of active users, with diverse modalities of information associated their profiles: textual status updates, comments on other user's pages, pictures, and friendships, to name a few. Any method that does not scale linearly in the amount of data is bound to fail. The other challenge is the presence of complex

structure in Facebook's data; its information is not presented as a simple feature vector, but as a cornucopia of structured inputs, multimodal in the sense that text, networks, and label data each seemingly requires a different approach to learning. Even the text alone cannot be treated as a simple bag of words, for it is separated into many comments and posts, with potentially sharp changes of topics and intents. One cannot fully model this rich structure with methods that require user data to be input as flat feature vectors, or that require a similarity function between them.

1.2 Solutions

With these challenges in mind, we present a scalable machine learning system that we use to visualize and explore the interests of millions of users on Facebook, and that potentially scales to tens or hundreds of millions of users. The key to this system is a unified latent space model jointly over text, network and label data, where some of its building blocks have been inspired by earlier successful attempts on certain modalities, such as the supervised Latent Dirichlet Allocation model over text and labels [6], the Mixed Membership Stochastic Blockmodel over networks [1], and the joint text/citation topic models of Nallapati *et al.* [18]. We call our model the Supervised Multi-view Mixed Membership Model (SM^4), which surmounts the multimodal data challenge by transforming user text, network and label data into an integrated latent feature vector for each user, and overcomes the scalability challenge by first training model parameters on a smaller subset of data, after which it infers millions of user feature vectors in parallel. Both the initial training phase and the integrated feature vector inference phase require only linear time and a single pass through the data.

Our system's most important function is visualization and exploration, which is achieved by deriving other kinds of information from the data in a principled, statistical manner. For instance, we can summarize the textual data as collections of related words, known as *topics* in the topic modeling literature [6, 5]. Usually, these topics will be coherent enough that we can assign them an intuitive description, e.g. a topic with the words "basketball", "football" and "baseball" is best described as a "sports" topic. Next, similar to Blei *et al.* [6], we can also report the correlation between each topic and the label under study — for instance, if we are studying the label "I vote Democratic", we would expect topics containing the words "liberal" and "welfare" to be positively correlated with said label. The value of this lies in finding *unexpected* topics that are correlated with the label. In fact, we will show that on Facebook, certain well-known brands are positively correlated with generic interests such as movies and cooking, while social gaming by contrast is negatively correlated. Finally, we can explain each friendship in the social network in terms of two topics, one associated with each friend. The motivation behind this last feature is simple: if we have two friends who mostly talk about sports, we would naturally guess that their friendship is due to mutual interest in sports. In particular, interests with a high degree of mutual interest friendships are valuable from a friendship recommendation perspective. As an example, perhaps "sports" is highly associated with mutual interest friendships, but not "driving". When ranking potential friends for a user who likes sports and driving, we should prefer friends that like sports over friends that like driving, as friendships could be more likely to form over sports.

From this latent topical model, we can construct visualizations like Figure 3 that summarize all text, network and label data in a single diagram. Using this visualization, we proceed with the main application of this paper, a cross-study of four general user interests, namely "camping", "cooking", "movies", and "sports". Our goal is to answer the questions posed earlier about user interests, conversations and friendships in Facebook, and thus glean insight into what makes Facebook unique, and how it functions. We also justify our analyses with quantitative results: by training a linear classifier [9] on the four interest labels and our system's user feature vectors, we demonstrate a statistically significant improvement in prediction accuracy over a bag-of-words baseline.

2. ALGORITHM OVERVIEW

Our goal is to analyze Facebook user data in the context of a general concept, such as "movies" or "cooking". Each Facebook user is associated with three types of data: text such as (but not limited to) user "status updates", network links between users based on friendships, and binary labels denoting interest in the concept ("I like movies") or lack thereof ("I don't like movies"). Intuitively, we want to capture the relationship between concepts, user text and friendships: for a given concept, we seek words correlated with interest in that concept (e.g. talking about actors may be correlated with interest in movies), as well as words that are most frequently associated with each friendship (e.g. we might find two friends that often talk about actors). By learning and visualizing such relationships between the input text, network and label data (see Figure 1), we can glean insight into the nature of Facebook's social structure.

Combining text and network data poses special challenges: while text is organized into multiple documents per user, networks are instead *relational* and therefore incompatible with feature-based learning algorithms. We solve this using an algorithm that learns a *latent feature space* over text, network and label data, which we call SM^4. The SM^4 algorithm involves the following stages:

1. Train the SM^4 probabilistic model on a subset of user text, network and label data. This learns parameters for a K-dimensional latent feature space over text, network and labels, where each feature dimension represents a "topic".
2. With these parameters, we find the best feature space representations of all users' text, network and label data. For each user, we infer a K-dimensional feature vector, representing her tendency towards each of the K topics.
3. The inferred user features have many uses, such as (1) finding which topics are most associated with friendships, and (2) training a classifier for predicting user labels.

The feature space consists of K topics, representing concepts and communities that anchor user conversations, friendships and interests. Each topic has three components: a vector of word probabilities, a vector of friendship probabilities to each of the K topics, and a scalar correlation w.r.t the user labels. As an example, we might have a topic with the frequent words "baseball" and "basketball", where this topic has a high self-friendship probability, as well as a high correlation with the positive user label "I like sports". Based on this topic's most frequent words, we might give it the name "American sports"; thus, we say that users who often talk about "baseball" and "basketball" are talking about "American sports". In addition, the high self-friendship probability of the "American sports" topic implies that such users are likely to be friends, while the high label correlation implies that such users like sports in general. Note that topics can have high friendship probabilities to *other* topics, e.g. we might find that "American sports" has a high friendship probability with a "Restaurants and bars" topic containing words such as "beer", "grill" and "television".

3. SUPERVISED MULTI-VIEW MIXED MEMBERSHIP MODEL (SM^4)

Formally, SM^4 can be described in terms of a *probabilistic generative process*, whose dependencies are summarized in a graphical model representation (Figure 2). Let P be the number of users, V

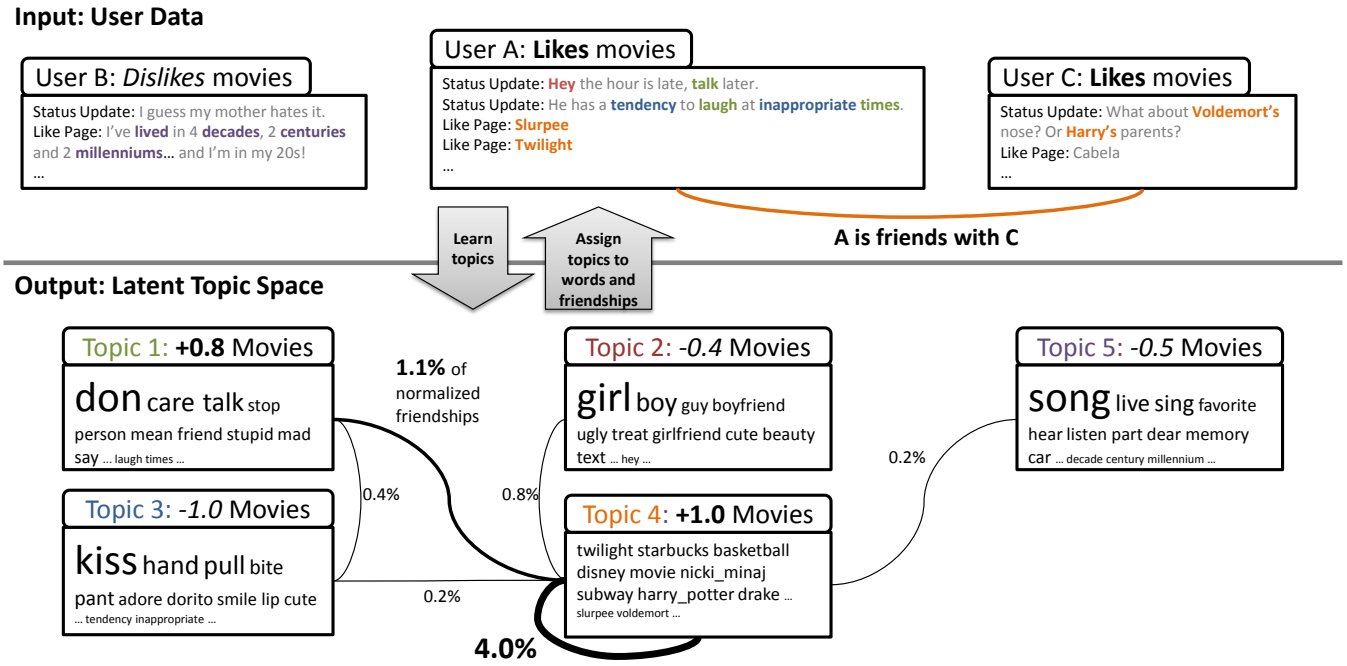

Figure 1: From user data to latent topic space, and back (best viewed in color). User data in the form of text (status updates and like page titles), friendships and interest labels (e.g. likes/dislikes movies) is used to learn a latent space of topics. Topics are characterized by a set of weighted keywords, a positive or negative correlation with the interest (e.g $+1.0$ Movies), and topic-topic friendship probabilities (expressed as the percentage of observed friendships, normalized by topic popularity). After learning the topics, we can assign the most probable topic to each user word, as well as the most probable topic-pair to each friendship — these assignments are represented by word and link colors. Observe that users with lots of green/orange words/friendships are likely to be interested in movies, as the corresponding topics (1,4) are detected as positive for movies.

the text vocabulary size, and K the desired number of topics. Also let D_i be the number of documents for user i, and W_{ik} the number of words in user i's k-th document. The generative details are described below:

- **Topic parameters:**
  - For the background vocabulary β_{back}, draw:
    - V-dim. word distribution $\beta_{back} \sim \text{Dirichlet}(\eta)$
  - For each topic $a \in \{1, \ldots, K\}$, draw:
    - V-dim. topic word distribution $\beta_{a\cdot} \sim \text{Dirichlet}(\eta)$
  - For each topic pair $(a, b) \in \{1, \ldots, K\}^2$, $a \leq b$, draw:
    - Topic-topic link probability $\Phi_{ab} \sim \text{Beta}(\lambda_1, \lambda_0)$
- **User features:** For each user $i \in \{1, \ldots, P\}$, draw:
  - User feature vector $\theta_i \sim \text{Dirichlet}(\alpha)$
- **Text:** For each user document $(i, k) \in \{1, \ldots, P\} \times \{1, \ldots, D_i\}$:
  - Draw document topic $z_{ik} \sim \text{Discrete}(\theta_i)$
  - For each word $\ell \in \{1, \ldots, W_{ik}\}$, draw:
    - Foreground-background indicator $f_{ik\ell} \sim \text{Bernoulli}(\delta)$
    - Word $w_{ik\ell} \sim \text{Discrete}((\beta_{z_{ik}})^{f_{ik\ell}}(\beta_{back})^{1-f_{ik\ell}})$
- **Friendship Links:** For each $(i, j) \in \text{EdgeList}$, $i < j$, draw:
  - User i's topic when befriending user j, $s_{ij} \sim \text{Discrete}(\theta_i)$
  - User j's topic when befriending user i, $s_{ji} \sim \text{Discrete}(\theta_j)$
  - Link $e_{ij} \sim \text{Bernoulli}(\Phi_{s_{ij}, s_{ji}})$ if $s_{ij} \leq s_{ji}$, else $e_{ij} \sim \text{Bern.}(\Phi_{s_{ji}, s_{ij}})$
- **Labels:** For each user $i \in \{1, \ldots, P\}$, draw:
  - Label $y_i \sim \text{Normal}(\hat{\theta}_i^\top \nu, \sigma^2)$, where $\hat{\theta}_i = \frac{\sum_k z_{ik} + \sum_j s_{ij}}{D_i + |\text{Neighbors}(i)|}$

While this generative process may seem complicated at first glance, we shall argue that each component is necessary for proper modeling of the text, network and label data. Additionally, the model's complexity does not entail a high runtime — in fact, our SM[4] algorithm runs in linear time with respect to the data, as we will show.

Topics and user data. Each user i has 3 data types: text data w_i, network links e_{ij}, and interest labels $y_i \in \{+1, -1\}$. In order to learn salient facts about all 3 datatypes seamlessly, we introduce a latent space feature vector for each user i, denoted by $\theta_i = (\theta_{i1}, \ldots, \theta_{iK})$. Briefly, a high value of θ_{ia} indicates that user i's text w_i, friendship patterns e_i and label y_i are similar to topic a.

Every topic $a \in \{1, \ldots, K\}$ is associated with 3 objects: (1) a V-dim. word probability vector β_a, (2) link formation probabilities $\Phi_{ab} \in [0, 1]$ to each of the K topics b, and (3) a coefficient ν_a that models the linear dependence of labels y_i with topic a. The vector β_a shows which words are most salient for the topic, e.g. a "US politics" topic should have high probabilities on the words "Republican" and "Democrat". The link probabilities Φ_{ab} represent how likely users talking about topic a are friends with users talking about topic b, e.g. "American sports" having many friendships with "Restaurants and bars". Finally, the coefficients ν_a show the correlation between topic a and the user interest labels y_i.

Text model. We partition user text data w_i into D_i documents $\{w_{i,1}, \ldots, w_{i,D_i}\}$, where each doc ik is a vector of W_{ik} words $(w_{ik,1}, \ldots, w_{ik,W_{ik}})$. Each document represents a "status update" by the user, or the title of a page she "likes". Compared to other forms of textual data like blogs, Facebook documents are very short. Hence, we assume each document corresponds to *exactly one topic* z_{ik}, and draw all its words $w_{ik\ell}$ from the topic word distribution $\beta_{z_{ik}}$ — a notable departure from most topic models [6, 8], which are tailored for longer documents such as academic papers.

Moreover, Facebook documents contain many keywords irrelevant to the main topic. For example, the message "I'm watching football with Jim, enjoying it" is about sports, but the words "watching" and "with" are not sports-related. To prevent such generic words from influencing topic word distributions β_a, we introduce per-

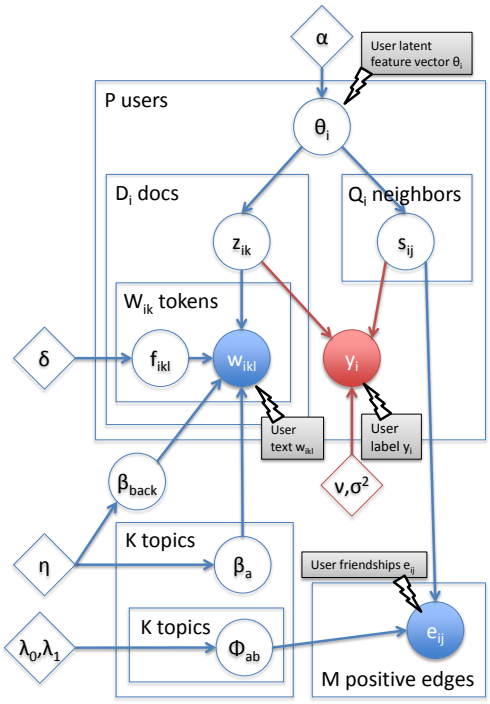

Figure 2: Graphical model representation of SM^4. Tuning parameters are diamonds, latent variables are hollow circles, and observed variables are filled circles. Variables pertaining to labels y_i are shown in red.

word foreground-background boolean indicators $f_{ik\ell} \sim \text{Bernoulli}(\delta)$, such that we draw $w_{ik\ell}$ from $\beta_{z_{ik}}$ as usual when $f_{ik\ell} = 1$, otherwise we draw $w_{ik\ell}$ from a "background" distribution β_{back}. By relegating irrelevant words to a background distribution, we can assign topics to entire documents without diluting the topic word distributions with generic words. More generally, the idea of having separate *classes* of word distributions was explored in [20, 12].

Network model. Let Neighbors(i) denote user i's friends, and let EdgeList denote all friendships (i, j) for $i < j$. Also, let $e_{ij} \in \{0, 1\}$ be the adjacency matrix of friendships, where $e_{ij} = 1$ implies $(i, j) \in$ Edgelist. In our model, friendships arise as follows: first, users i, j draw topics s_{ij} and s_{ji} from their feature vectors θ_i, θ_j. Then, the friendship outcome e_{ij} is generated from s_{ij}, s_{ji} — this is in contrast to words $w_{ik\ell}$, which are generated from only one topic z_{ik}. Specifically, e_{ij} is drawn from a upper-triangular $K \times K$ matrix of Bernoulli parameters Φ; we draw e_{ij} from $\Phi_{s_{ij},s_{ji}}$ if $s_{ij} < s_{ji}$, otherwise we draw from $\Phi_{s_{ji},s_{ij}}$. Essentially, Φ describes friendship probabilities between topics.

Because the Facebook network is *sparse*, we only model positive links; the variables s_{ij}, s_{ji}, e_{ij} exist if and only if $e_{ij} = 1$. The zero links $e_{ij} = 0$ are used in a Bayesian fashion: we put a $\text{Beta}(\lambda_1, \lambda_0)$ prior on each element of Φ, and set $\lambda_0 = \ln(\#[\text{zero links}]/K^2)$ and $\lambda_1 = 0.1$, where $\#[\text{zero links}] = P(P-1)/2 - |\text{EdgeList}|$. Thus, we account for evidence from zero links without explicitly modeling them, which saves a tremendous amount of computation.

Label model. We extract labels $y_i \in \{+1, -1\}$ from users' "liked" pages, e.g. "music" and "cooking". By including labels, we can learn which topics are positively/negatively correlated with user interests. Similar to sLDA [6], we draw user labels $y_i \sim \text{Normal}(\hat{\theta}_i^\top \nu, \sigma^2)$, where $\hat{\theta}_i$ is the average over user i's text topic indicators z_{ik} and network indicators s_{ij} (represented as *indicator vectors*). Put simply, a user's label is a linear regression over her topic vector θ_i.

3.1 Training Algorithm

Our SM^4 system proceeds in two phases: a training phase to estimate the latent space topic parameters $\beta, \Phi, \nu, \sigma^2$ from a smaller subset of users, followed by a parallel prediction phase to estimate user feature vectors θ_i and friendship topic-pair assignments s_{ij}, s_{ji} for each friendship $e_{ij} = 1$. In particular, the s_{ij}, s_{ji} provide the most likely "explanation" for each friendship, and this forms a cornerstone of our data analysis in Section 6.

Right now, we shall focus on the details of the training algorithm. Our first step is to simplify the training problem by reducing the number of latent variables, through analytic integration of user feature vectors θ and topic word/link parameters β, Φ via Dirichlet-Multinomial and Beta-Binomial conjugacy. Hence, the only random variables that remain to be inferred are $\mathbf{z}, \mathbf{f}, \mathbf{s}$ (which now depend on the tuning parameters α, η, δ). Once $\mathbf{z}, \mathbf{f}, \mathbf{s}$ have been inferred, we can recover the topic parameters β, Φ from their values. We also show that our algorithm runs in linear time w.r.t the amount of data, ensuring scalability.

Training Algorithm (1) alternates between Gibbs sampling on $\mathbf{z}, \mathbf{f}, \mathbf{s}$, Metropolis-Hastings on tuning parameters α, η, δ, and direct maximization of ν, σ^2. This hybrid approach is motivated by simplicity — Gibbs samplers for models like ours [11] are easier to derive and implement than alternatives such as variational inference, while α, η, δ are easily optimized through the Metropolis-Hastings algorithm. As for the Gaussian parameters ν, σ^2, the high dimensionality of ν makes MCMC convergence difficult, so we resort to a direct maximization strategy similar to sLDA [6].

3.1.1 *Gibbs sampler for latent variables* $\mathbf{z}, \mathbf{f}, \mathbf{s}$

Document topic indicators $\mathbf{z}$. A Gibbs sampler samples every latent variable, conditioned on the current values of all other varibles. We start by deriving the conditional distribution of z_{ik}:

$$\begin{aligned}
&\mathbb{P}(z_{ik} = m \mid \mathbf{z}_{-ik}, \mathbf{f}, \mathbf{w}, \mathbf{s}, \mathbf{e}, \mathbf{y}) \qquad (1)\\
&\propto \mathbb{P}(y_i \mid z_{ik} = m, \mathbf{z}_{i,-k}, \mathbf{s}_i)\mathbb{P}(w_{ik\cdot} \mid z_{ik} = m, \mathbf{z}_{-ik}, \mathbf{f}_{ik\cdot}, \mathbf{w}_{-ik\cdot})\\
&\quad \times \mathbb{P}(z_{ik} = m \mid \mathbf{z}_{i,-k}, \mathbf{s}_i)\\
&\propto \exp\left\{-\frac{(y_i - \hat{\theta}_i^\top \nu)^2}{2\sigma^2}\right\} \frac{\Gamma(V\eta + \sum_{v=1}^V A_v)}{\prod_{v=1}^V \Gamma(\eta + A_v)} \frac{\prod_{v=1}^V \Gamma(\eta + B_v + A_v)}{\Gamma(V\eta + \sum_{v=1}^V B_v + A_v)}\\
&\quad \times \left(\#[\{\mathbf{z}_{i,-k}, \mathbf{s}_i\} = m] + \alpha\right),
\end{aligned}$$

where we use the fact that $\mathbb{P}(w_{ik\ell} \mid z_{ik} = m, f_{ik\ell} = 0, \mathbf{z}_{-ik}, \mathbf{w}_{-ik\cdot})$ is independent of z_{ik}, and where we define

$$A_v = |\{(x, y, u) \mid (x, y) \neq (i, k) \wedge f_{xyu} = 1 \wedge z_{xy} = m \wedge w_{xyu} = v\}|,$$
$$B_v = |\{u \mid f_{iku} = 1 \wedge w_{iku} = v\}|,$$

where A_v is the number of non-background words $\mathbf{w} = v$ assigned to topic m and not belonging to user i and document k, and B_v is similar but for words belonging to user/document ik. Note that $\hat{\theta}_i$ in the exp is a function of z_{ik}, and was defined in Section 3.

The distribution of z_{ik} is composed of a prior term for $z_{ik} = m$ and two posterior terms, one for user i's label y_i, and one for document ik's words $w_{ik\cdot}$. The posterior term for y_i is a Gaussian, while the posterior term for $w_{ik\cdot}$ is a Dirichlet Compound Multinomial (DCM) distribution, which results from integrating the word distribution β_m. Notice that background words, i.e. $w_{ik\ell}$ such that $f_{ik\ell} = 0$, do not show up in this posterior term. Finally, the z_{ik} prior term is the DCM from integrating the feature vector θ_i.

Importantly, the counts A_v, B_v can be cached and updated in constant time for each z_{ik} being sampled, and therefore Eq. (1) can be

computed in constant time w.r.t. the number of documents. Hence, sampling all $\mathbf{z}$ takes linear time in the number of documents.

Word foreground-background indicators $\mathbf{f}$. The conditional distribution of $f_{ik\ell}$ is

$$\begin{aligned}
&\mathbb{P}(f_{ik\ell}=1 \mid \mathbf{z}, \mathbf{f}_{-ik\ell}, \mathbf{w}, \mathbf{s}, \mathbf{e}, \mathbf{y}) \qquad (2)\\
&= \mathbb{P}(w_{ik\ell} \mid \mathbf{z}, f_{ik\ell}=1, \mathbf{f}_{-ik\ell}, \mathbf{w}_{-ik\ell})\mathbb{P}(f_{ik\ell}=1)\\
&\quad \times [\mathbb{P}(w_{ik\ell} \mid \mathbf{z}, f_{ik\ell}=1, \mathbf{f}_{-ik\ell}, \mathbf{w}_{-ik\ell})\mathbb{P}(f_{ik\ell}=1)\\
&\qquad + \mathbb{P}(w_{ik\ell} \mid \mathbf{z}, f_{ik\ell}=0, \mathbf{f}_{-ik\ell}, \mathbf{w}_{-ik\ell})\mathbb{P}(f_{ik\ell}=0)]^{-1}\\
&= \left(\frac{(\eta+E_{w_{ik\ell}})\delta}{V\eta+\sum_{v=1}^{V}E_v}\right)\left(\frac{(\eta+E_{w_{ik\ell}})\delta}{V\eta+\sum_{v=1}^{V}E_v}+\frac{(\eta+F_{w_{ik\ell}})(1-\delta)}{V\eta+\sum_{v=1}^{V}F_v}\right)^{-1},
\end{aligned}$$

where $E_v = |\{(x,y,u) \mid (x,y,u) \neq (i,k,\ell) \wedge f_{xyu}=1 \wedge z_{xy}=z_{ik} \wedge w_{xyu}=v\}|$,

and $F_v = |\{(x,y,u) \mid (x,y,u) \neq (i,k,\ell) \wedge f_{xyu}=0 \wedge w_{xyu}=v\}|$.

E_v is the number of non-background words $\mathbf{w}=v$ assigned to topic z_{ik}, excluding $w_{ik\ell}$. F_v is similar, but for background words (regardless of topic indicator z).

Ignoring the normalizer, the distribution of f_{ikl} contains a posterior term for $w_{ik\ell}$ and a prior term for f_{ikl}. Again, the $w_{ik\ell}$ term is a DCM; this DCM comes from integrating $\beta_{z_{ik}}$ if $f_{ik\ell}=1$, otherwise it comes from integrating the background word distribution β_{back}. The $f_{ik\ell}$ prior is a simple $\text{Bernoulli}(\delta)$. As with Eq. (1), the counts E_v, F_v can be cached with constant time updates per $f_{ik\ell}$, thus sampling all $\mathbf{f}$ is linear time in the number of words $\mathbf{w}$.

Link topic indicators $\mathbf{s}$. Recall that we only model s_{ij}, s_{ji}, e_{ij} for positive links $e_{ij}=1$. For convenience, let $e_{ji}=e_{ij}$ for all $i<j$. The resulting conditional distribution of s_{ij} is

$$\begin{aligned}
&\mathbb{P}(s_{ij}=m \mid \mathbf{z}, \mathbf{f}, \mathbf{w}, \mathbf{s}_{-ij}, e_{ij}=1, \mathbf{e}_{-ij}, \mathbf{y}) \qquad (3)\\
&\propto \mathbb{P}(y_i \mid \mathbf{z}_i, s_{ij}=m, \mathbf{s}_{i,-j})\mathbb{P}(e_{ij}=1 \mid s_{ij}=m, s_{ji}, \mathbf{s}_{-\{ij,ji\}}, \mathbf{e}_{-ij})\\
&\quad \times \mathbb{P}(s_{ij}=m \mid \mathbf{z}_i, \mathbf{s}_{i,-j})\\
&\propto \exp\left\{-\frac{(y_i-\hat{\theta}_i^\top\nu)^2}{2\sigma^2}\right\}\frac{\lambda_1+C}{\lambda_1+\lambda_0+C}\left(\#[\{\mathbf{z}_i,\mathbf{s}_{i,-j}\}=m]+\alpha\right),
\end{aligned}$$

$$C = \begin{cases} |\{(x,y)\in \text{EdgeList} \mid (x,y)\neq(i,j) \wedge [(s_{xy},s_{yx})=(m,s_{ji}) \\ \quad \vee (s_{xy},s_{yx})=(s_{ji},m)]\}| & \text{if } i<j \\ |\{(y,x)\in \text{EdgeList} \mid (x,y)\neq(i,j) \wedge [(s_{xy},s_{yx})=(m,s_{ji}) \\ \quad \vee (s_{xy},s_{yx})=(s_{ji},m)]\}| & \text{if } i>j. \end{cases}$$

C is the number of positive links $\mathbf{e}\setminus e_{ij}$ whose topic indicators (s_{xy},s_{yx}) are identical to the topics (s_{ij},s_{ji}) of e_{ij}. The OR clauses simply take care of situations where $s_{xy}>s_{yx}$ and/or $s_{ij}>s_{ji}$. The distribution of s_{ij} contains a prior term for $s_{ij}=m$ (the DCM from integrating θ_i), a Gaussian posterior term for y_i, and a link posterior term for e_{ij} (the Beta Compound Bernoulli distribution from integrating out the link probability $\Phi_{m,s_{ji}}$).

Like Eq. (1,2), C can be cached using constant time updates per s_{ij}, thus sampling all $\mathbf{s}$ is linear in the number of friendships $|\text{EdgeList}|$. Combined with the constant time sampling for Eq. (1,2), we see that the SM^4 algorithm requires linear time in the amount of data.

3.1.2 *Learning tuning parameters* α, η, δ *and* ν, σ^2

We automatically learn the best tuning parameters α, η, δ using Independence Chain Metropolis-Hastings, by assuming α, η are drawn from $\text{Exponential}(1)$, while δ is drawn from $\text{Beta}(1,1)$. For ν, σ^2, we take a Stochastic Expectation-Maximization [10] approach, in which we maximize the log-likelihood with respect to ν, σ^2 based on the current Gibbs sampler values of $\mathbf{z}, \mathbf{s}$. The maximization has a closed-form solution similar to sLDA [6], but without the expectations:

$$\nu \leftarrow \left(A^\top A\right)^{-1} A^\top b, \qquad \sigma^2 \leftarrow \frac{1}{P}\left[b^\top b - b^\top A\nu\right] \qquad (4)$$

where A is a $P \times K$ matrix whose i-th row is the current Gibbs sample of $\hat{\theta}_i$, and b is a P-vector of user labels y_i.

Updating all parameters $\alpha, \eta, \delta, \nu, \sigma^2$ requires linear time in the amount of data, so we update them once per Gibbs sampler sweep over all latent variables $\mathbf{z}, \mathbf{f}, \mathbf{s}$. This ensures that every iteration (Gibbs sweep plus parameter update) takes linear time.

Algorithm 1 SM^4 Training Algorithm

Input: Training user text data $\mathbf{w}$, links $\mathbf{e}$ and labels $\mathbf{y}$
Randomly initialize $\mathbf{z}, \mathbf{f}, \mathbf{s}$ and parameters $\alpha, \eta, \delta, \nu, \sigma^2$
Set λ_1, λ_0 according to Section 3, Network Model
repeat
  Gibbs sample all $\mathbf{z}, \mathbf{f}, \mathbf{s}$ using Eqs. (1,2,3)
  Run Metropolis-Hastings on tuning parameters α, η, δ
  Maximize parameters ν, σ^2 using Eq. (4)
until Iteration limit or convergence
Output: Sufficient statistics for $\mathbf{z}, \mathbf{f}, \mathbf{s}$, and all parameters $\alpha, \eta, \delta, \lambda_1, \lambda_0, \nu, \sigma^2$

Algorithm 2 SM^4 Parallelizable Prediction Algorithm

Input: Parameters $\beta, \Phi, \alpha, \delta, \nu, \sigma^2$ from training phase
Input: Test user p's text data $\mathbf{w}_p$
Randomly initialize $\mathbf{z}_p, \mathbf{f}_p$ for the test user
repeat
  Gibbs sample $\mathbf{z}_p$ using Eq. (1), and $\mathbf{f}_p$ using Eq. (2)
until Iteration limit or convergence
Estimate test user's feature vector θ_p from his $\mathbf{z}_p$
Use θ_p to predict s_{pj}, s_{jp} for all friends j
Output: Test user's θ_p, s_{pj}, s_{jp}

3.2 Parallelizable Prediction Algorithm

Our training algorithms learns topic parameters β, Φ, ν, so that we can use our Prediction Algorithm (2) to predict feature vectors θ_p and friendship topic-pair assignments s_{pj}, s_{jp} for *all* users p. For each user p independently and *in parallel*, we Gibbs sample her text latent variables $z_{p\cdot}, f_{p\cdot\cdot}$ based on her observed documents $w_{p\cdot\cdot}$ and the learnt parameters $\beta, \Phi, \nu, \sigma^2$. Then, using the definition of our SM^4 generative process, we estimate p's feature vector θ_p by averaging over her $z_{p\cdot}$. Finally, we use θ_p and the learnt topic parameters Φ to predict p's most likely friendship topic-pair assignments s^*_{pj}, s^*_{jp} to each of her friends j, using this equation:

$$(s^*_{pj}, s^*_{jp}) = \arg\max_{(a,b)\text{ s.t. }a<b} \theta_{p,a}\Phi_{a,b}\theta_{j,b}. \qquad (5)$$

We use these assignments to discover the topics that friendships are most frequently associated with. Like the training algorithm, the Prediction Algorithm also runs in linear time.

4. EXPERIMENTAL SETTING

Our goal is to analyze Facebook users in the context of their interests, friendships and conversations. Facebook users typically express interests such as "movies" or "cooking" by establishing a "like" relation with the corresponding Facebook pages, and our experiments focus on four popular user interests in Facebook: camping, cooking, movies and sports. We selected these concepts because of their broad scope: not only are they generic concepts, but each of their pages was associated with more than 5 million likes

as of May 2011, ensuring a sufficiently large user base for data collection. For each interest C, we collected our data as follows:

1. Construct the complete data collection $S(C)$ by randomly selecting 1 million users who like interest C ($y_i = +1$), and 1 million who do not explicitly mention liking C ($y_i = -1$).
2. For each user $i \in S(C)$, collect the following data[1]:
   - **User text documents** $w_{ik.}$: The text documents for user i contain all of her "status updates" from March 1st to 7th, 2011 (each status update is one document), as well as titles of Facebook pages that she likes by March 7th 2011 (each page title is one document)[2]. We preprocessed all documents using typical NLP techniques, such as stopword removal, stemming, and collocation identification [14].
   - **User-to-user friendships**: We obtained these symmetric friendships using the friend lists of user i recorded on March 7th 2011.
3. Randomly sample 2% of $S(C)$ to construct a 40,000-user training collection $\bar{S}(C)$. Across the four concepts, $\bar{S}(C)$ contained 340,128 to 385,091 unique words, 6,650,335 to 8,771,298 documents, 16,421,601 to 22,521,507 words, and 1,292 to 2,514 links[3].

We first trained the SM^4 model using the training collection $\bar{S}(C)$ and $K = 50$ latent features (topics), stopping our Gibbs sampler at the 100th iteration because 1) the per-iteration increase in log-likelihood was $< 1\%$ of the cumulative increase, and 2) more iterations had negligible impact on our validation experiments. This process required 24 hours for each concept, using one computational thread. We note that one could subsample larger training collections $\bar{S}(C)$, thus increasing the accuracy of parameter learning at the expense of increased training time. A recently introduced alternative is to apply approximate parallel inference techniques such as distributed Gibbs sampling [16, 2], but these introduce synchronization and convergence issues that are not fully understood yet.

After learning topic parameters from the training collection $\bar{S}(C)$, we invoke Algorithm 2 on all users $p \in S(C)$ to obtain their predicted feature vectors θ_p, and the friendship topic-pair "explanations" s_{pj}, s_{jp} for each of p's friends j. Note that Algorithm 2 is parallelizable over every user in $S(C)$, and we observe that it only requires a few minutes per user; a sufficiently large cluster finishes all 2M users in a single day — in fact, given enough computing power, it is possible to scale our prediction to all of Facebook. In the following sections, we shall apply the predicted θ_p, s_{pj}, s_{jp} to various analyses of Facebook's data.

4.1 Mapper Data Imbalance

Our cluster completed every 2M user experiment within 24 hours, but there is more to performance than this number alone. In fact, for every experiment, the first mapper (CPU) to complete sampling its assigned users did so within a few hours, while the last mapper took close to 24 hours (hence the total runtime).

We believe this runtime asymmetry is due to data imbalance; most users only have a few documents (status updates and like pages), but a few users have *thousands* of documents or even more. Moreover, not all documents have the same length; most contain just a few words, yet the occasional document has tens or hundreds of words. As a result, the Hadoop scheduler fails to partition data equally among mappers, thus some mappers receive several times more data than others. Because the algorithm has to wait for the last mapper to finish, this leads to a several-fold increase in runtime.

One solution would be to subsample data from the largest users and documents, so as to limit the total number of words per user. We expect subsampling from a user with large amounts of data to have limited statistical impact on our model's parameter and latent variable estimates, while allowing the Hadoop scheduler to better partition data across mappers. While we did not test this solution, we expect its implementation to reduce our total runtime dramatically.

5. VALIDATION

Before interpreting our results, we must validate the performance of our SM^4 model and algorithm. Because our model spans multiple data modalities, there is arguably no single task or metric that can evaluate all aspects of SM^4. What we shall do is test how well the SM^4 latent space and feature vectors predict held-out user interest labels y_p from our data collections $S(C)$. We believe this is the best task for several reasons: for one, we are concerned with interpreting user interests in the context of friendships and conversations, thus we must show that the SM^4 latent space accurately captures user interests. For another, predicting user interests is a simple and well-established task, and its results are therefore easier to interpret than model goodness-of-fit measures such as perplexity (as used in [7]).

It is well-understood that textual latent space methods like Latent Dirichlet Allocation (LDA), while useful for summarization and visualization, normally do not improve classification accuracy — in fact, with large amounts of training data, they may actually perform worse than a naive Bag-of-Words (BoW) representation [7]. This stems from the fact that latent space methods are dimensionality reduction techniques, and thus distort the data by necessity. In our case, the picture is more complicated: the text aspect of our model loses information with respect to BoW, yet some non-textual information comes into play from the friendship links and labels in the small training collections $\bar{S}(C)$. We believe the best way to use SM^4 is to concatenate SM^4 features to the BoW features — this avoids the information loss from reducing the dimensionality of the text, while allowing the network and label information to come into play. We expect this to yield a modest (but statistically significant) improvement in accuracy over a plain BoW baseline.

Our task setup is as follows: recall that for each interest C, we obtained a 2M data collection $S(C)$ with ground truth labels for all user interests y_p. The SM^4 algorithm predicts feature vectors θ_p for all users $p \in S(C)$, which can be exploited to learn a linear Support Vector Machine (SVM) classifier for the labels y_p. More specifically, we use θ_p concatenated with user p's original BoW as feature inputs to LIBLINEAR [9], and then performed 10-fold cross-validation experiments on the labels y_p. This was done for each of the four data collections $S(C)$, and each experiment took $<$ 1 hour. As a baseline, we compare to LIBLINEAR trained on BoW features only. The BoW features for user p are just the normalized word frequencies over all her documents.

Table 1 summarizes our results. To determine if the improvement from SM^4 is statistically significant, we conducted a χ^2-test (one degree of freedom, 2M trials) against the BoW Baseline as a null hypothesis. The p-values are far below 0.001, suggesting that the improvement provided by SM^4 features is statistically very signif-

[1]We use only non-private user data for our experiments, e.g. chat logs or user messages are never looked at.

[2]We remove the page title of concept C, because its distribution is highly correlated with the labels.

[3]The relatively small number of links arises from unbiased random sampling of users; more links can be obtained by starting with a seed set of users and picking their friends, but this introduces bias. Also, our method uses evidence from negative links, so the small number of positive links is not necessarily a drawback.

Table 1: User interest classification accuracy (in percent) under a 10-fold cross-validation setup, for a Bag-of-Words baseline, and BoW plus SM^4 feature vectors. Each experiment is performed over 2 million users. We also report χ^2-statistics and p-values (1 degree of freedom), which show that adding SM^4 features yields a highly significant improvement in accuracy.

Features	Sports	Movies	Camping	Cooking
BoW Baseline	78.91	78.51	79.85	77.22
Plus SM^4	**80.23**	**80.48**	**81.08**	**78.57**
χ^2-statistic	2.1×10^5	4.6×10^5	1.9×10^5	2.1×10^5
p-value	$\ll 0.001$	$\ll 0.001$	$\ll 0.001$	$\ll 0.001$

icant. This confirms our hypothesis that the SM^4 features improve classification accuracy, by virtue of encoding network and label information from the small training collections $\bar{S}(C)$. We expect that classification accuracy will only increase with larger training collections $\bar{S}(C)$, albeit at the expense of more computation time.

6. UNDERSTANDING USER INTERESTS AND FRIENDSHIPS IN FACEBOOK

In the introduction, we posed four questions about Facebook:

- How does Facebook's social (friendship) graph interact with its interest graph and conversational content? Are they correlated?
- What friendship patterns occur between users with similar interests?
- Do users with similar interests talk about the same things?
- How do different interests (say, camping and movies) compare? Do groups of users with distinct interests also exhibit different friendship and conversational patterns?

We shall answer these questions by analyzing our SM^4 output over the four user interests: camping, cooking, movies and sports. Such analysis is not only useful for content recommendation, but can also inform policies targeted at increasing connectivity (making more friends) and interaction (having more conversations) within the social network. Through continuous study of user interests, conversations and friendships, we hope to learn what makes the social network unique, and what must be done to grow it.

6.1 Visualization procedure

In Figure 3, we combine SM^4's output over all four user interests into one holistic visualization, and the purpose of this section is to describe how we constructed said visualization. First, recall that for each interest C, our SM^4 system learns topic parameters from a training subset $\bar{S}(C)$ of user text documents, friendship links, and labels. These parameters are then used to infer various facts about the full user dataset $S(C)$: (1) user feature vectors θ_p that give their propensities towards various topics, and (2) each friendship's most likely topic-pair assignments s_{ij}, s_{ji}, which reveal the topics a given pair of friends is most likely to talk about.

With these learnt parameters, we search for the 6 most strongly-recurring topics across all four interests, as measured by cosine similarity. These topics, shown in the middle of Figure 3, represent commonly-used words on Facebook, and provide a common theme that unites the four user interests. Next, for each interest, we search for the top 4 topic-pairs (including pairs of the same topic) with the highest friendship counts (which come from the topic-pair assignments s_{ij}, s_{ji}). Note that we first normalize each topic-pair friendship count by the popularity[4] of both topics, in order to avoid selecting popular but low-friendship topics. We show these 4 topic-pairs in the corners of Figure 3, along with their normalized friendship counts. These topic-pairs represent conversations between friends; more importantly, if the topics are also positively correlated with the user interest — say, camping — then they reveal what friends who like camping actually talk about. This context-specificity is especially valuable for separating generic chatter from genuine conversation about an interest.

Figure 3 was constructed by these rules, but with one exception: we include a Movies topic (heading Mo (0.6%) $+1.64$) that lacks strong friendships, yet is positively correlated with interest in movies. This anomaly demonstrates that interest-specific conversations do not always occur between friends — in other words, the presence of an interest-specific conversation does not imply the existence of friendship, which is something that text-only systems may fail to detect. In turn, this highlights the need for holistic models like SM^4 that consider interests, conversations and friendships jointly.

6.2 Observations and Analysis

Common Topics. Throughout these sections, we shall continually refer to Figure 3. The most striking observation about the four interests (camping, cooking, moving, sports) is their *shared topical content*, shown in the middle of the Figure. These topics represent a common lingo that permeates throughout Facebook, and that can be divided into two classes: "Facebook fanpages", consisting of named entities that have pages on Facebook for users to like, and "Informal conversation in status updates", which encompasses the most common, casual words from user status updates.

We observe that the fanpage topic starting with "adam_sandler" is dominant, with popularity $> 10\%$ across all four user interest datasets. Additionally, this topic has a mild positive correlation with all interests, meaning that users who have any of the four interests are more likely to use this topic. In contrast, the fanpage topic starting with "cash" only has average popularity (between $1 - 2\%$) and mild negative correlation with all interests. Observe that this topic is dominated by social gaming words ("farmville", "mafia_wars"), whereas the other, popular topic is rich in popular culture entities such as "Disney", "Dr Pepper", "Simpsons" and "Starbucks". This data provides evidence that users who exhibit any of the four interests tend to like pop culture pages over social gaming pages. Notably, none of these four interests are related to internet culture or gaming, which might explain this observation.

The informal conversation topics are more nuanced. Notice how the topic starting with "buddy" is both popular and strongly correlated with respect to cooking and movies, implying that the conversations of cooking/movie lovers differ from camping/sports lovers. Also, notice that the topic starting with "beauty" is dominated by romantic words such as "boyfriend" and "girlfriend", and is popular/correlated only with sports — perhaps this lends some truth to the stereotype that school athletes lead especially active romantic lives. Finally, the topic starting with "annoy" and containing words such as "dad", "mom" and "house" carries a slight negative sentiment for all interests (in addition to being unpopular). This seems reasonable from the average teenager's perspective, in which parents normally have little connection with personal interests.

High-Friendship Topics. We turn to the high-friendship topics in the corners of Figure 3. Some of these contain a high degree of self-friendships, implying that friends usually converse about the same topic, rather than different ones. To put it succinctly, in Facebook, the interest graph is correlated with the social (friendship) graph. In fact, the average proportion of same-topic friendships ranges from 0.2% to 0.6% depending on interest, whereas the average proportion of inter-topic friendships is an order of magnitude lower at 0.02% to 0.04%. Intuitively, this makes sense: any coherent dialogue between friends is necessarily about a single topic;

[4]The sum of a topic's weight over all user feature vectors θ_p.

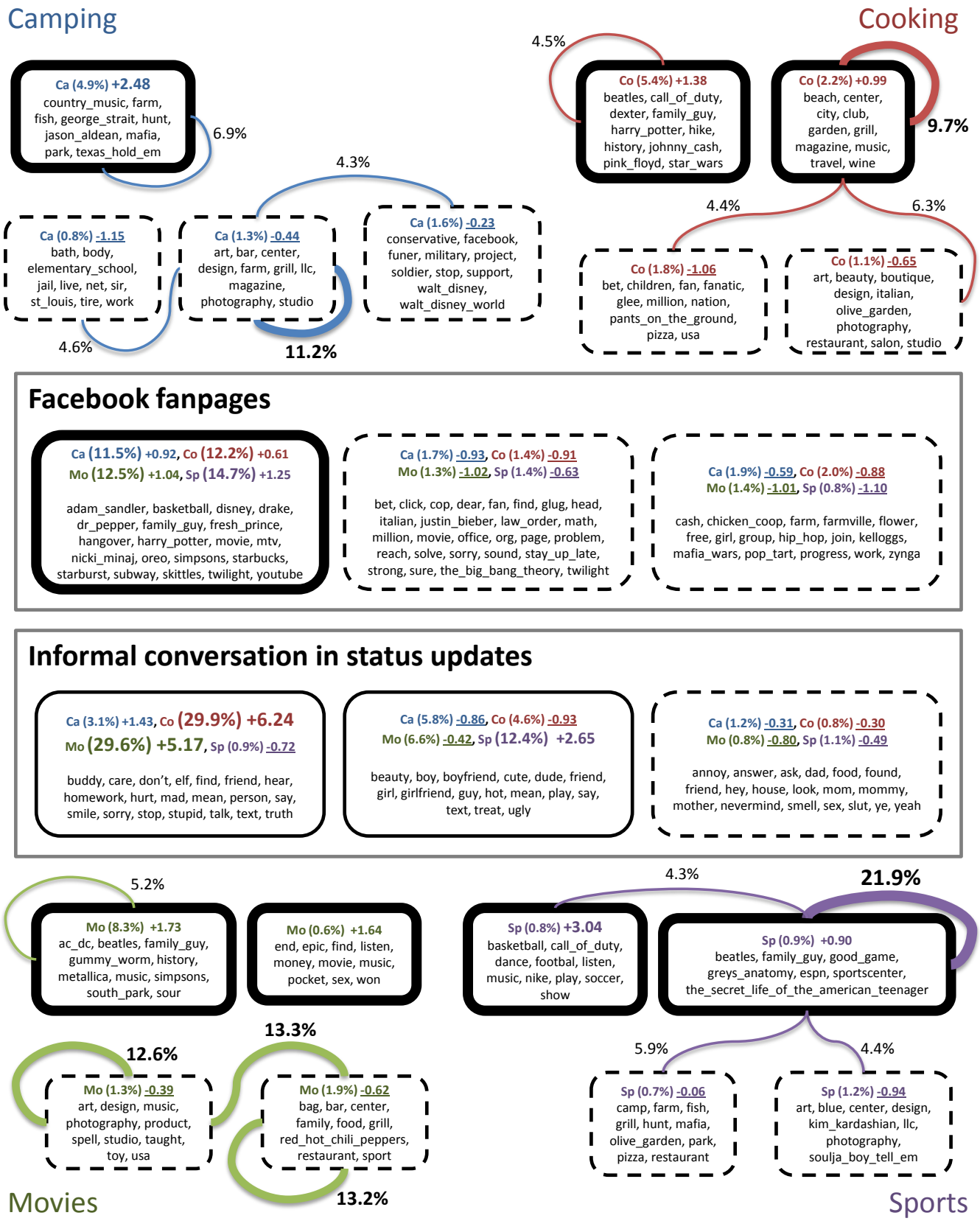

Figure 3: **A visual summary of the relationship between Facebook friendships, user conversations, and 4 types of user interests (best viewed in color). Topics specific to a particular interest are found in the corners, while common topics are found in the middle, divided into topics containing Facebook fanpage titles or status update lingo — note that we manually introduced this distinction for the sake of visualization; the SM^4 algorithm discovers all topics purely from the data. Thick borders highlight topics positively correlated with user interests, while dashed borders highlight negative correlation. Font colors highlight information relevant to a specific interest: blue for camping (ca), red for cooking (co), green for movies (mo), and purple for sports (sp). The colored heading in each topic describes its popularity, and its correlation with user interests: for example, "Ca (4.9%) +2.48" means this topic accounts for 4.9% of user text in the camping dataset, and has a moderate positive correlation with interest in camping. Finally, an edge between a pair of topics shows the proportion of friendships attributed to that pair (normalized by topic popularity).**

multiple-topic conversations are hard to follow and thus rare.

One interpretation of inter-topic friendships is that they signify two friends who rarely interact, hence their conversations on the whole are topically distinct. In other words, inter-topic friendships may represent socially weaker ties, compared to same-topic friendships. As an example, consider the cooking topics starting with “art" and “conservative" respectively. The former topic is about the visual arts (“design", “photography", “studio"), whereas the latter topic is about political conservatives in America (“military", “soldier", “support"). It seems implausible that any conversation would be about both topics, and yet there are friendships between people who talk about either topic — though not necessarily with each other.

A second observation is that most interests have more than one positively correlated topic (with the exception of camping). A good example is cooking: notice the topics starting with “beach" and “beatles" respectively. The former topic has connotations of fine living, with words like “city", “club", “travel" and “wine", whereas the latter is associated with entertainment culture, containing phrases like “beatles", “family_guy", “pink_floyd" and “star_wars". Both topics have statistically much in common: moderate popularity, positive interest correlation with cooking, and a significant proportion of self-topic friendships. Yet they are semantically different, and more importantly, do not have a significant proportion of friendships between them. Hence, these two topics represent separate communities of cooking lovers: one associated with the high life, the other with pop culture. The fact that cooking lovers are not homogenous has significant implications for policy and advertising; a one-size-fits-all strategy is unlikely to succeed.

Similar observations can be made about sports and movies: for sports, both a television topic (“family_guy", “greys_anatomy", "espn") and an actual sports topic (“basketball", “football", “soccer") are positively correlated with interest in sports, yet users in the former topic are likely *watching* sports rather than *playing* them. As for movies, one topic is connected with restaurants and bars (“bar", “food", “grill", “restaurant"), while the other is connected with television (“family_guy", “simpsons", “south_park").

Our final observation concerns the “friendliness" of users in positive topics — notice that the users of some positively correlated topics (“country_music" from camping, “ac_dc" from movies, “beatles" from cooking") have plenty of within-topic friendships, yet possess almost no friendships with other topics. In contrast, users in topics like “beach" from cooking or “beatles" from sports are highly gregarious, readily making friends with users in other topics. The topic words themselves may explain why: notice that the “beach" cooking topic has words like “club", “grill" and “travel" that suggest highly social activities, while the “beatles" sports topic contains television-related words such as “family_guy" and “espn", and television viewing is often a social activity as well.

In closing, our analysis demonstrates how a multi-modal visualization of Facebook’s data can lead to insights about network connectivity and interaction. In particular, we have seen how fanpages and casual speech serve as a common anchor to all conversations on Facebook, how same-topic friendships are far more common (and meaningful) than inter-topic friendships, and how users with common interests can be hetorogenous in terms of conversation topics. We hope these observations can inform policy directed at growing the social network, and increasing the engagement of its users.

7. RELATED WORK

The literature contains other topic models that combine several data modalities; ours is distinguished by the assumptions it makes. In particular, existing topic models of text and network data either treat the network as an outcome of the text topics (RTM [8]), or define new topics for each link in the network (ART [15]). The Pairwise Link-LDA model of Nallapati *et al.* [18] is the most similar to ours, except (1) it does not model labels, (2) it models asymmetric links only, and crucially, (3) its inference algorithm is infeasible for even $P = 40,000$ users (the size of our training $\bar{S}(C)$'s) because it models all $\mathrm{O}(P^2)$ positive *and* zero links. Our model escapes this complexity trap by only considering the positive links.

We also note that past work on Facebook’s data [19] used the network implicitly, by summing features over neighboring users. Instead, we have taken a probabilistic perspective, borrowing from the MMSB model [1] to cast links into the same latent topic space as the text. Thus, links are neither a precursor to nor an outcome of the text, but *equals*, resulting in an intuitive scheme where both text and links derive from specific topics. The manner in which we model the labels is borrowed from sLDA [6], except that our links also influence the observed labels $\boldsymbol{y}$.

8. FUTURE WORK

An open question raised by our work is how to efficiently compute parameter estimates from very large networks, under a particular network model. In our SM^4 model, which uses the Mixed-Membership Stochastic Blockmodel (MMSB) as a sub-component, the challenge is to infer topic assignments for every network edge (whether 0 or 1). Because the total number of network edges is $\mathrm{O}(P^2)$ (where P is the number of users), inference on every edge is completely infeasible for large networks. Our solution was to incorporate evidence from the 0-edges directly into the prior, spread evenly across all elements of the link probability matrix Φ. By using the 0-edge evidence in this manner, we only have to model the 1-edges, which are far less numerous than the 0-edges.

Another approach would be to subsample the edges — ideally, we want at most $\mathrm{O}(P)$ edges, resulting in an (amortized) constant amount of work per user. Under a stochastic blockmodel, if we knew the true topic-pair assignments of all edges — i.e., if we knew which edges corresponded to which elements of the link-probability matrix — then the obvious solution is to find, for all elements of the link-probability matrix, the set of edges corresponding to that element, and then subsample $\mathrm{O}(P)$ edges from that set. This ensures that every matrix element gets $\mathrm{O}(P)$ samples to use for estimation. Of course, we *do not know* the topic-pair assignments in advance — the question to ask, then, is can we still construct a sampling strategy with a *lower bound* on the number of edges picked per matrix element, under “reasonable" assumptions about the true blockmodel? The existence of such a strategy implies that we can subsample the network, and still be confident in our estimate of the link-probability matrix.

A third solution is to design a completely different network model, one that relies on *higher-order network motifs* such as triangle relationships between three nodes. Such motifs can be emitted by a topic/admixture model, except that each motif is determined by not one topic indicator, but several (just like how MMSB emits edges using two topic indicators, one per participating node). By restricting ourselves to just those interesting (but not *too* frequent) motifs, we can design novel admixture network models that avoid the 0-edge complexity trap that MMSB and other probabilistic network models suffer from (this is because real networks have $\mathrm{o}(P^2)$

1-edges, implying $O(P^2)$ 0-edges). We are currently developing such an admixture network model, and we expect its inference algorithm runtime to scale better than MMSB w.r.t. the number of users P.

9. CONCLUSION

In conclusion, we have tackled salient questions about user interests and friendships on Facebook, by way of a system that combines text, network and label data to produce insightful visualizations of the social structure generated by millions of Facebook users. Our system's key component is a latent space model (SM^4) that learns the aggregate relationships between user text, friendships, and interests, and this allows us to study millions of users at a macroscopic level. The SM^4 model is closely related to the supervised text model of sLDA [6] and the network model of MMSB [1], and combines features of both models to address our challenges. We ensure scalability by splitting our learning algorithm into two phases: a training phase on a smaller user subset to learn model parameters, and a parallel prediction phase that uses these parameters to predict the most likely topic vectors θ_p for each user, as well as the most likely friendship topic-pair assignments s_{ij}, s_{ji} for all friendships $e_{ij} = 1$. Because the inference phase is trivially parallelizable, our system potentially scales to all users in Facebook.

10. REFERENCES

[1] E. Airoldi, D. Blei, S. Fienberg, and E. Xing. Mixed membership stochastic blockmodels. *The Journal of Machine Learning Research*, 9:1981–2014, 2008.

[2] A. Asuncion, P. Smyth, and M. Welling. Asynchronous distributed learning of topic models. *Advances in Neural Information Processing Systems*, 21:81–88, 2008.

[3] C. Basu, H. Hirsh, W. Cohen, and C. Nevill-Manning. Technical paper recommendation: A study in combining multiple information sources. *JAIR*, 14(1):231–252, 2001.

[4] R. Bell and Y. Koren. Lessons from the netflix prize challenge. *ACM SIGKDD Explorations Newsletter*, 9(2):75–79, 2007.

[5] D. Blei and J. Lafferty. Topic models. *Text mining: classification, clustering, and applications*, 10:71, 2009.

[6] D. Blei and J. McAuliffe. Supervised topic models. In *NIPS*, pages 121–128. MIT Press, Cambridge, MA, 2008.

[7] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[8] J. Chang and D. Blei. Relational topic models for document networks. *AISTATS*, 9:81–88, 2009.

[9] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

[10] W. Gilks, S. Richardson, and D. Spiegelhalter. *Markov chain Monte Carlo in practice*. Chapman & Hall/CRC, 1996.

[11] T. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1):5228, 2004.

[12] T. Griffiths, M. Steyvers, D. Blei, and J. Tenenbaum. Integrating topics and syntax. *Advances in neural information processing systems*, 17:537–544, 2005.

[13] W. Hill, L. Stead, M. Rosenstein, and G. Furnas. Recommending and evaluating choices in a virtual community of use. In *SIGCHI*, pages 194–201. ACM Press/Addison-Wesley Publishing Co., 1995.

[14] B. Krenn. Collocation mining: Exploiting corpora for collocation identification and representation. In *Journal of Monetary Economics*, 2000.

[15] A. McCallum, X. Wang, and A. Corrada-Emmanuel. Topic and role discovery in social networks with experiments on enron and academic email. *JAIR*, 30(1):249–272, 2007.

[16] D. Mimno and A. McCallum. Organizing the oca: Learning faceted subjects from a library of digital books. In *the 7th ACM/IEEE-CS joint conference on digital libraries*, pages 376–385. ACM, 2007.

[17] R. Mooney and L. Roy. Content-based book recommending using learning for text categorization. In *the fifth ACM conference on Digital libraries*, pages 195–204. ACM, 2000.

[18] R. Nallapati, A. Ahmed, E. Xing, and W. Cohen. Joint latent topic models for text and citations. In *KDD*, pages 542–550. ACM, 2008.

[19] C. Wang, R. Raina, D. Fong, D. Zhou, J. Han, and G. Badros. Learning relevance in a heterogeneous social network and its application in online targeting. In *SIGIR 2011*. ACM, 2011.

[20] C. Zhai and J. Lafferty. Two-stage language models for information retrieval. In *SIGIR 2002*, pages 49–56. ACM, 2002.