An I iteration Gibbs sampling does '$IPCXZ$' computations for assigning topic, community and types to each post and '$I \times \sum_{p \in P} W_p = IW$' computations for drawing each word in the posts. Hence, the algorithm has the worst time complexity $O(IPCXZ + IW)$.

Topics can be computed using the approximation :

$$P(w|z) = \frac{n_{wz}^{-p} + \delta}{\sum_{w'} n_{w'z}^{-p} + V\delta}$$

Another approach is to find the maximum likelihood estimate for $P(w|z)$ using $P(\mathbf{W}|z)$.

$$P(w_i|z) = \sum_{j=1}^{P} P(w_i|\mathbf{W_j}) P(\mathbf{W_j}|z)$$

where, $P(w_i|\mathbf{W_j}) = \frac{N_{w_i}^p}{N_p}$ and P is the number of posts in the corpus. We observed that the results are similar in both approaches. For the sake of simplicity, we choose the MLE approach.

The community memberships $P(u|c)$ and topic proportions for a community $P(z|c)$ are also estimated in a similar manner. The $P(u|c)$ computation gives us the community as a distribution over users while $P(z|c)$ gives us the topical interest in each community.

3.2 Topic User Recipient Community Model 1

Now, we describe the model to capture user interests in the second kind of networks which do not allow mass messaging. In such networks, the sender typically sends out messages to his/her acquaintances. Consequently, the posts are on a topic of interest to both the sender and the recipient of the post. Figure 1b shows the TURCM-1 model where the post topics and communities are generated corresponding to sender-recipient pair for the post.

The generative scheme for TURCM-1 closely follows that of the TUCM described above. Multinomials $\vec{\eta}_{u_i,u_j}$ and $\vec{\theta}_{u_i,u_j}$ are drawn for every sender-recipient pair and the post topic and community assignments are drawn from them respectively using appropriate sender-recipient indices.

The Gibbs Sampling updates for TURCM-1 are similar with user u replaced by a user-recipient pair ur. Since the recipient information is only used as an index (to signify mutual interest:there are no extra draws) the worst case time complexity continues to be $O(IPCXZ+IW)$ as for TUCM.

3.3 Topic User Recipient Community Model 2

TURCM -1 accounted for recipient information of posts by ensuring that topic and community assignments for posts are not only based on users but user-recipient pairs of posts. However, this is not the same as accounting for links in a social graph as it does not in any way ensure formation of closely knit communities as closeness in the the link structure is not explicitly captured.

Next, we describe another variation which models recipients differently. Network neighborhood is modeled as an interaction. We again go back to the Social interaction profile (SIP) formulation described earlier and generate post recipients corresponding to the community drawn. This is the same as representing each user in the social graph as a document with user indices of its neighbors as words. Communities will cluster nodes with similar neighborhood in one community just as LDA clusters similar words as topics! Figure 1c shows the TURCM-2 model.

The generative scheme for TURCM-2 is also on the lines of TUCM. Multinomials $\vec{\eta}_{u_i}$ and $\vec{\theta}_{u_i}$ are drawn for every user. The recipients in the set of recipients $\mathbf{R_p}$ of post p are drawn from the multinomial $\vec{\psi}_{c_p}$ based on the community assignment drawn for the post.

The Gibbs Sampling updates to infer TURCM 2's parameters contain an additional update for the recipient draw.

$$P(\mathbf{R_p} = \mathbf{r}|\mathbf{R_{-p}}, \mathbf{C}) = \frac{\prod_{r \in \mathbf{R_{p_i}}} (n_{cr_{p_i}}^{-j} + \epsilon)}{\prod_{j=0}^{R_{p_i}-1} (\sum_{r'} (n_{r'c}^{-j} + i + U\epsilon))}$$

where P_i is the post under consideration.

The additional draws for recipients have an additional overhead of '$I \times \sum_{p \in P} R_p = IR$' computations over TUCM, where R is the sum of number of recipients for all the posts in the corpora. Hence, the worst case time complexity for Full TURCM becomes $O(IPCXZ + IW + IR)$.

3.4 Full TURCM

In this section we describe the Full TURCM model which is another modification to the earlier models. In previous models we have assumed that each post generated by a user is based on a single topic. This assumption may not hold true for networks which have large post sizes - for Example, the Enron email network data set. The Full TURCM model removes this assumption by generating a topic for each word in a post (instead of generating a topic per post). Further, as a result of this modification we now, generate posts based on the community that an author belongs to. The generative model shown in figure 1d is explained below in detail:

1. For each of the topics, $1 \leq z \leq Z$, sample a V dimensional multinomial, $\vec{\lambda}_z \sim Dir_V(\delta)$. This distribution represents the topic as distributions over words.
2. For each of the communities, $1 \leq c \leq C$ sample a X dimensional social type interaction mixture $\vec{\phi}_c \sim Dir_X(\beta)$.
3. For each of the communities, $1 \leq c \leq C$ sample a U dimensional social recipient interaction mixture $\vec{\psi}_c \sim Dir_U(\epsilon)$.
4. For the i^{th} user u_i, $1 \leq u_i \leq U$:
   (a) Sample a C dimensional multinomial, $\vec{\theta}_{u_i} \sim Dir_C(\alpha)$, representing the community proportions for that sender.
   (b) For each community $c \in C$, sample a Z dimensional multinomial, $\vec{\eta}_{u_i,c} \sim Dir_Z(\nu)$, representing the topic proportions for community and sender.
   (c) For each post p ($1 \leq p \leq P_i$) generated by the sender u_i: having N_p words:
      i. Choose a community assignment $c_p \sim Mult(\vec{\theta}_{u_i})$ $c_p \in [1 : C]$ for the post.
      ii. For each recipient slot i, $1 \leq i \leq R_p$ of the post p:
         A. Choose a recipient $r_p \sim Mult(\vec{\psi}_{c_p})$ $r_{p_i} \in [1 : R_p]$ for the post.

iii. Choose a social interaction type $X_p \sim Mult(\vec{\phi}_{c_p})$, $X_p \in [1:X]$ for the post.

iv. For each word slot j, $1 \leq j \leq N_p$ in p:

A. Choose a topic assignment $z \sim Mult(\vec{\eta}_{u_i,c_p})$, $z \in [1:Z]$.

B. Choose a word $w_j \sim Mult(\vec{\lambda}_{z_{w_j}})$.

3.4.1 Parameter Estimation

The Gibbs sampling update equations for the Full TURCM model are attained as:

$$P(c_p = c|\mathbf{C}_{-\mathbf{p}}, \mathbf{U}, \mathbf{R}, \mathbf{X}, \mathbf{Z}) \propto \frac{n_{cu}^{-p} + \alpha}{\sum_{c'} n_{cu}^{-p} + C\alpha} \frac{n_{xc}^{-p} + \beta}{\sum_{x'} n_{xc}^{-p} + X\beta} \times$$

$$\prod_{r \in \mathbf{R_p}} \frac{n_{cr} + \epsilon}{\sum_{r'} n_{r'c} + i + R_{p_i}\epsilon} \frac{\prod_{z=1}^{Z} \Gamma(e_{p,zu} + n_{z(c_p u_p)}^{-p} + \beta)}{\Gamma(\sum_{z=1}^{Z} e_{p,zu} + n_{z(c_p u_p)}^{-p} + Z\beta)}$$

where, $e_{p,zu}$ is the number of times topic z was generated from user u other than post p.

$$P(z_{(p,i)} = z|\mathbf{Z}_{-(\mathbf{p},\mathbf{i})}, \mathbf{C}, \mathbf{U}, \bar{\mathbf{W}}) \propto \frac{n_{z(cu)}^{-(p,i)} + \nu}{\sum_{z'} n_{z'(cu)}^{-(p,i)} + Z\nu} \frac{n_{wz}^{-(p,i)} + \delta}{\sum_{w'} n_{w'z}^{-(p,i)} + Z\delta}$$

The Full TURCM model, by relaxing the assumption that a post is always on a single topic incurs the overhead of drawing the topic assignment for each word (and not each post) in the corpus. The number of computations for assigning topics to each word grows to IZW, and the overall complexity to $O(IPCX + IZW)$. However this computational overhead is compensated by improvements in quality of community discovery, especially when posts are too long and hence on more than one topic.

4. EXPERIMENTS

In this section, we evaluate our models on two real world data sets and compare them with the CUT and CART models. Most well known definitions of communities lay emphasis on two things: How tightly users in a community are inter-connected, and how strongly the users in a community share interests?

Our models discover shared interests from content produced by the users. To begin with, we give a qualitative evaluation of the communities obtained and try to argue how topics, links and types combine to produce communities effectively. We evaluate the strength of inter-connections in the community structure by computing the fuzzy modularity [7] of the communities obtained. Modularity is a popular measure used to quantify the quality of division of a network into modules or communities. Finally, we show learning time improvements over the two models.

4.1 Datasets

We use two different datasets for our experiments, one is the freely available collections of tweets crawled from Twitter over a period of six months in 2009 [14] [6] and the other is the Enron Email corpus[5]. Twitter is a social networking and micro blogging service where users communicate by short text messages (up to 140 characters) called 'tweets'. Tweets are publicly visible by default. However senders can restrict message delivery. Users may subscribe to other users' tweets by following them. These follower relationships impose an underlying graph structure. On the other hand, the Enron dataset contains email exchanges from about 150 employees, mostly senior management. We choose these two datasets for the diversity and challenges they bring along with them. While Twitter imposes a restriction on the length of posts, the number of followers of a user can run into millions. In a social graph, these nodes (users) are sometimes called 'star nodes'. This is a case when the graph is dense but the associated content is much smaller. On the other hand, while the Enron dataset has fewer nodes, emails can be arbitrarily long; case of a sparse graph but rich content. Scaling a technique that integrates both content and link to such diverse datasets is an important challenge. Probabilistic techniques like those employed by us that work on word frequencies have the added advantage that they are less susceptible to noise. We do not have to undergo extensive data cleansing or data preparation.

We crawled a sub-graph from Twitter for our analysis using a simple heuristic approach. We began from a set of influential seed nodes and grew the graph by using follower relationships. The Twitter dataset used for our experiments has 5405 nodes, 13214 edges and 23043 posts. The link types as described in section 3 are retained and the posts are preprocessed to remove stop words on both datasets.

[5]http://www.cs.cmu.edu/ enron/

Travel	Internet	H1N1	Stock Markets
arrive	free	influenza	curve
monday	download	symptom	shift
pm	msn	america	daily
friday	explorer	chicken	market
midday	internet	cure	numbers

Table 1: Topics extracted from Twitter dataset

California Power	Gas Transportation	Trading	Deals
power	gas	price	meeting
transmission	energy	market	contract
energy	enron	dollar	report
calpx	transco	nymex	enron
california	chris	trade	deal

Table 2: Topics extracted from Enron

4.2 Results

In all the models presented above, Z and C, the number of topics and the bumber of communities respectively are free parameters. This requires some sensitivity analysis and introduction of quality functions (such the modularity) in order to fix the best values. We will describe our strategy for optimal parameter setting later.

For all our simulations, we set the number of communities C at 10 and topics Z at 20 unless stated otherwise. These choices are later proved to be close to the optimal in Sections 4.2.2 and 4.2.3 We ran 1000 iterations to burn in and took 250 samples (every fourth sample) in the next 1000 iterations.

4.2.1 Qualitative Analysis

In Table 2 and 3, we list a few topics ($\vec{\lambda}_z$) discovered by TUCM over both datasets. We give top 5 words to visualize

Management	Engineering	Analyst
contract	Power	price
agreement	Transmission	dollar
meeting	Electric	cash
corporation	epsa	database
budget	calpx	meeting

Table 3: Role discovery for users

each topic. Here "calpx" is the California Power Corporation, "transco" is a gas transportation company and "nymex" is the New York Mercantile Exchange. We see that TUCM is able to discover meaningful topics, thus validating the assumption that each post is associated with a single topic. This is particularly true for the Twitter corpus (where post lengths are constrained) and generally true for the Enron corpus as seen in later experiments. Topic visualization was similar in other models, with only minor changes.

Next, in figure 2 we illustrate the probability density over topics ($\vec{\eta}_{u_i}$) for a particular Twitter user (user 93). It shows that this user is primarily interested in topic 14 (Stock Markets) and also likes topics 4 (Internet). This analysis is useful in finding individual user's interests and tastes. [9] showed how one can discover social roles of people by associating words with users (i.e. $P(w|u)$) through their topical interests. In Table 3 we give top words for a few roles using the Enron corpus. From these words we can see that social roles are nothing but work profiles for people working in Enron, like management, engineering and analyst. These social roles were confirmed against their true roles in Enron.

It is also possible to uncover community membership proportions (i.e. $P(c|u)$) for every user. For instance, in figure 3, user 93 has a membership in community 4 to a high degree. Besides, the model suggests that the user also participates in community 6 to some extent. This analysis gives us an insight to the extent of participation of a user in various social groups. This probabilistic notion of membership has clear advantages in modeling user tastes and preferences to the hard clustering approach taken in previous non probabilistic approaches for community discovery.

Now that we know how to estimate user interests and community memberships, we can also compute the topical interest of the communities formed by those users. With this analysis we can corroborate the intuition that communities are formed when users with similar interests aggregate together. In Figure 4, we show the topic distributions for different communities. Topical peaks for a community indicate the dominant topics for that particular community (i.e. $P(z|c)$). For example, after looking at the topic proportions for community 4 (see figure 5), which is the primary community for user 93, it was found that topics 14(Stock Markets) is the dominant topic in this community. Also topic 4 (Internet) is the dominant topic in community 6 (recall that user 93 also has a high membership in community 6). This kind of analysis supports our hypothesis that users tend to communicate frequently over certain topics (based on interests) and form communities which discusses them to varying degrees. Similar visualization is possible for all four models.

4.2.2 Community Analysis

Next, we evaluate the quality of communities discovered by these models against communities discovered by the CUT and the CART models.

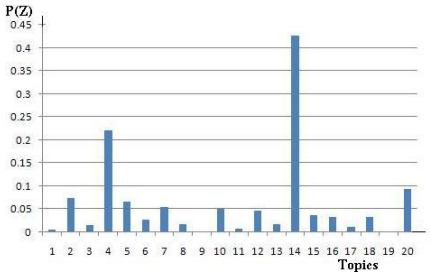

Figure 2: Topic proportions for a user

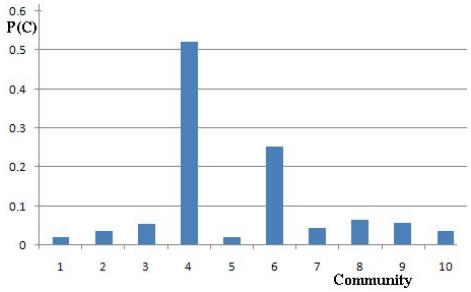

Figure 3: Community proportions for a user

Newman [10] proposed modularity, a measure of goodness for community structure. It assumes that a good division of the network is not merely one in which the number of edges running between groups is small. Rather, it is one in which the number of edges between groups is smaller than expected. Modularity, Q is defined as:

$$Q = \begin{array}{l}(\textit{number of edges within communities}) \\ -(\textit{expected number of such edges})\end{array}$$

While the application of modularity has been questioned from time to time (like in [3] and [4]), it continues to be the most popular and widely accepted measure of the goodness of community structure. Since the output of all our probabilistic models is a fuzzy community structure, in which each node has a certain probability of belonging to a certain community; we use a fuzzy variant of modularity Q_f proposed in [7]. Hence, Fuzzy Modularity provides a measure of goodness for the fuzzy community structure in networks.

For a particular fuzzy partition of a graph with n nodes and m edges where $\{\rho_k(x)\}_{k=1}^n$ is the fuzzy membership of n nodes in the k communities $(S_1 \dots S_k)$, the approach is to classify nodes according to the majority rule, i.e. if $k = argmax_l \rho_l(x)$ for a given node x then we set $x \in S_k$. Then the fuzzy modularity Q_f is defined as:

$$Q_f = \frac{1}{2m} \sum_{k=1}^{n} \sum_{x,y \in S_k} \left(\frac{\rho_k(x) + \rho_k(y)}{2} e(x,y) - p_f^E(x,y)\right)$$

where $p_f^E(x,y)$ is the expected probabilistic number of edge $e(x,y)$ with the form $p_f^E(x,y) = \frac{d_f(x) d_f(y)}{2m}$; $x, y \in S_k$ and $d_f(x)$ is the extended degree of node x in community S_k under the probabilistic setting and given by

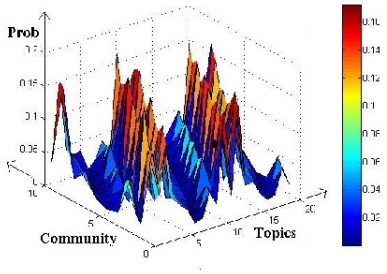

Figure 4: Distribution of topics within communities

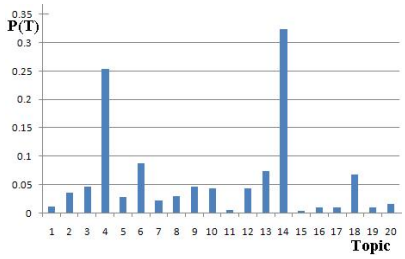

Figure 5: Distribution of topics in community 4

$$d_f(x) = \sum_{z \in S_k} \frac{\rho_k(x) + \rho_k(z)}{2} e(x,z) + \sum_{z \notin S_k} \frac{\rho_k(x) + (1 - \rho_k(z))}{2} e(x,z)$$

Tables 4 and 5 compares the fuzzy modularity of our models with their peers (CUT and CART). Full TURCM finds most meaningful communities as it not only integrates all three links, post types and content but also does not force every post to be on one topic. TUCM does well on the Twitter dataset as it can better model the short messages, poor content and noise present in Twitter data. On the other hand, in the Enron email dataset broadcasting is not allowed. Since users post directed messages to other users, the topic is of interest to both the user and the recipient. Consequently, recipient information becomes very relevant. Hence, TURCM-1 and TURCM-2 do well on Enron. TURCM-2 outperforms TRUCM-1 in both datasets due to better modeling of recipients.

Finally, the high modularity values support our assumption that people who share common interests and are interconnected with each other in the social graph often form communities. Modularity is an important measure in our claim as one is always interested in strong-knit communities where people know each other as well as share common interests for reasons such as networking and task assignment. Methods that form communities purely on interest can end up with disparate people (who do not know each other and are disconnected in the graph) in one community. This is shown by much weaker numbers for the CUT model in Tables 4 and 5 . The number of topics was set to 20 for these experiments as we vary the number of communities.

These results not only establish better community modeling by our models than its peers (CUT and CART) but also

Number of Communities	6	8	10	12	14
TUCM	0.167	0.263	0.321	0.313	0.262
TURCM-1	0.168	0.261	0.309	0.287	0.241
TURCM-2	0.166	0.265	0.324	0.309	0.261
Full TURCM	0.171	0.272	0.332	0.316	0.267
CART	0.157	0.227	0.243	0.235	0.196
CUT	0.159	0.244	0.299	0.285	0.237

Table 4: Fuzzy modularity on the Twitter dataset

Number of Communities	6	8	10	12	14
TUCM	0.148	0.243	0.291	0.287	0.246
TURCM-1	0.198	0.271	0.339	0.331	0.283
TURCM-2	0.203	0.278	0.346	0.337	0.289
Full TURCM	0.215	0.294	0.363	0.350	0.299
CART	0.152	0.249	0.302	0.294	0.255
CUT	0.133	0.231	0.266	0.278	0.227

Table 5: Fuzzy modularity on the Enron dataset

give interesting insights on how important role the choice of model plays. While models like TURCM-1 and TURCM-2 and CART are less suitable for sparse datasets like Twitter where link information is not dominant, they become increasingly important when the social graph information is rich. Similar trends are also observed in perplexity comparisons that are provided next.

4.2.3 *Perplexity Analysis*

Perplexity is one of the most important measures used to evaluate language models, especially topic models. Intuitively, it measures the log likelihood of generating unseen data after learning from a fraction of data. A higher value of perplexity implies a lesser model likelihood and hence lesser generative power of the model. As against most topic models where data is just the set of words, we compute the perplexity of observing both link types and words.

We analyze and compare the perplexity of our models for the two datasets with the CUT and CART models. We divided the data into training and test portions randomly in different proportions and investigated how perplexity behaves as more and more data is used for training. Let N_{total} be the size of the corpus. Let $p_1 \dots p_N$ be the posts used for training and $p_{N+1} \dots p_{N_{total}}$ be used for testing.

$$Perplexity = \exp\left(-\frac{\log P(p_{N+1} \dots p_{N_{total}} | p_1 \dots p_N)}{N_{total} - N}\right)$$

Since each post is generated independently,

$$P(p_{N+1} \dots p_{N_{total}} | p_1 \dots p_N) = \prod_{i=N+1}^{test} P(p_i | p_1 \dots p_N)$$

For the TUCM model, post likelihood can be computed as:

$$P(p_i | p_1 \dots p_N) = \sum_{z \in \mathbf{Z}} \eta_{z u_{p_i}} \sum_{c \in \mathbf{C}} \theta_{c u_{p_i} z} \phi_{c x_{p_i}} \prod_{w \in \mathbf{W_{p_i}}} \lambda_{wz}$$

For the rest, the likelihoods can be computed accordingly.

First, we explore the perplexity of our models on the two datasets. We also compare our models with CUT and

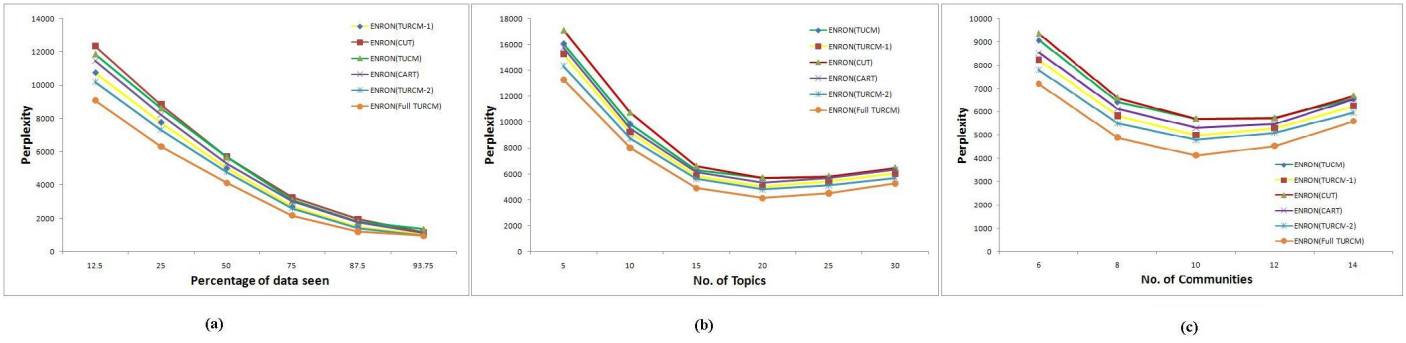

Figure 6: Perplexity on Enron vs (a) Percentage of data seen (b) No. of Topics (c) No. of Communities

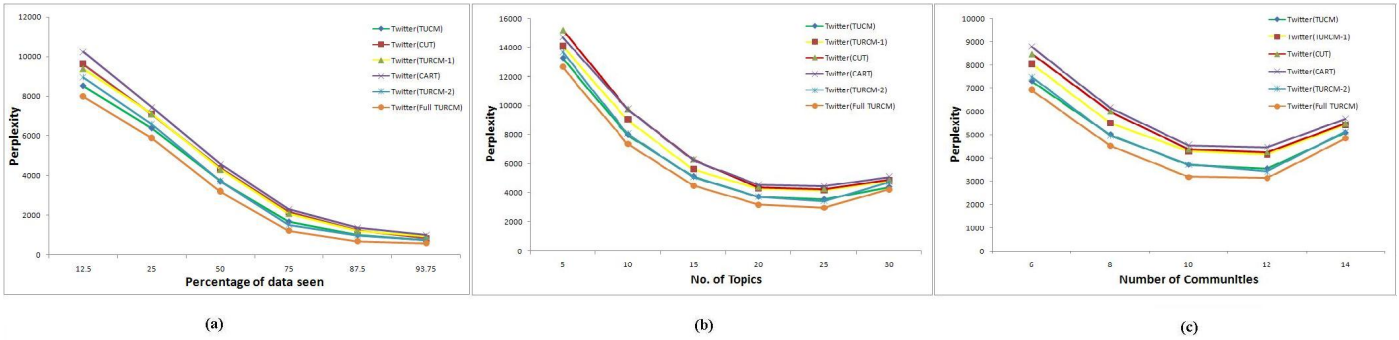

Figure 7: Perplexity on Twitter vs (a) Percentage of data seen (b) No. of Topics (c) No. of Communities

CART in terms of perplexity. Figures 6 and 7 give the comparison. As expected perplexities fall with the amount of data seen, suggesting an improvement in model tuning as more and more data is encountered. Again, Full TURCM does the best on both datasets which can be accredited to its better modeling of topics. TUCM does a better on the Twitter dataset than Enron. This is because the post length constraint in Twitter better suits the mixture of unigrams assumption. Also, we can observe TUCM outperform TURCM-1 and TURCM-2 for the Twitter dataset. This is because the large volume of broadcast tweets in the Twitter data set makes the recipient interest in a topic less relevant. However, TURCM-1 has an edge over TUCM in the Enron dataset due to the absence of broadcast messages. Consistently good values for perplexity on the two diverse datasets for all the models indicates the generalization ability of the models despite variability in graph density, link types, noise and post lengths. The results also establish significant improvements by the models over the CUT and CART models. This is because our models not only generate topic based communities but also account for graph topology and link types which are important descriptors of the strength of relationship between users. The results also confirm to our intuition that in some cases (like Twitter), the post topics model user's (author's) interests alone, while in other cases they model joint interest of the sender and the recipient. For these reasons, CART doesn't perform too well on the Twitter dataset and CUT doesn't perform well on ENRON.

Finally, in order to comment on optimal model parameter settings (number of topics and communities), we analyze how model perplexities behave as the model parameters are changed. Figures 6 (b) and 7 (b) plot the perplexities against the number of topics. The number of communities was set to 10 for this experiment. In both datasets for all the models, it can be roughly concluded that the peplexities attain their minimum at around 20-25 topics. Figures 6 (c) and 7 (c) plot the perplexities against the number of communities. The number of topics was set to 10 for this experiment. Again, for both the datasets, it can be concluded that the perplexities attain their minimum at around 10-12 communities. Similar insights were obtained about the number of communities from Tables 4 and 5 where the fuzzy modularities optimize around 10 communities for both datasets. This analysis not only helps us in concluding that our models outperform the two baselines (CUT and CART) independent of model parameter settings (Number of Topics and Number of Communities) but also in obtaining an estimate on optimal model parameter settings for both datasets.

4.2.4 *Runtime Analysis*

One major bottleneck with probabilistic models for social networks is their scalability. Social networks like Twitter and Facebook run into millions of nodes and probabilistic models due to an involved inference mechanism cannot scale to these sizes. Next, we compare the training time of the TUCM model against CUT and CART models. Table 6 gives the speed-ups: proportional increase in training times (the ratio of training times of the models over that of TUCM) against the number of nodes in the network. All simulations have been done on the same architecture and platform. In all models, training is assumed to finish when the model likelihoods converge (change in likelihood over consecutive iterations falls below 1% of the likelihood recorded in the previous iteration). The improvements

Number of Nodes	CUT	CART	TURCM1	TURCM2	Full TURCM
152	1.10	1.22	1.02	1.07	1.29
351	1.21	1.35	1.04	1.16	1.42
1049	1.49	1.64	1.13	1.29	1.95
2558	1.63	1.77	1.19	1.34	2.97
5405	1.81	1.99	1.23	1.52	4.23
8486	2.37	2.73	1.34	1.86	8.12

Table 6: Training time speed up of TUCM over other models

are primarily because of the mixture of unigrams approach adopted in TUCM where we draw one topic for each post unlike CUT and CART models which encounter the overhead for each word in the post. CUT model does marginally better than CART. This is because the recipient information is less relevant community modeling in Twitter.

Among the four models presented in this paper; although TURCM-1 has the same worst case complexity as TUCM, it has a slightly greater overhead of accounting for recipient indices. TURCM-2 makes additional computations in drawing recipients of the posts. More importantly, due to the overhead of topic generation for each word, Full TURCM takes significantly longer for the likelihood to converge. This speedup over models which do not make a unigram assumption continues to grow with data sizes rendering such models practically unusable in real size social networks. Consequently, the value of models like TUCM which model type information intelligently and efficiently can be realized.

5. CONCLUSION

We began by positing that communities are formed by users who communicate on topics of mutual interest, are connected to each other in the social graph and share frequent personal communication. We discriminated between datasets where posts modeled author interests alone or mutual interest between the author and the recipient. To cover both cases, we proposed probabilistic schemes that incorporate topics, social relationships and nature of posts for more effective community discovery. We argued that interaction types are important indicators of the strength of association between users. Our models give us the capability of visualizing the topics a community is interested in besides offering topic modeling capabilities and discovering topics of interest for a user. Finally, we show superior community discovery results in the form of fuzzy-modularity and perplexity improvements. Our models also show significant reduction in training time over their closest peers making them scalable to large real-life social networks.

6. REFERENCES

[1] J. Chang and D. Blei. Relational topic models for document networks. In *AIStats*, 2009.

[2] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75 – 174, 2010.

[3] S. Fortunato and M. Barthelemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41, 2007.

[4] B. H. Good, Y.-A. de Montjoye, and A. Clauset. Performance of modularity maximization in practical contexts. *Phys. Rev. E*, 81:046106, Apr 2010.

[5] K. Henderson, T. Eliassi-Rad, S. Papadimitriou, and C. Faloutsos. Hcdf: A hybrid community discovery framework. In *SDM 10*, 2010.

[6] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a social network or a news media? In *Proceedings of the International Conference on World Wide Web*, 2010.

[7] J. Liu. Fuzzy modularity and fuzzy community structure in networks. *The European Physical Journal B - Condensed Matter and Complex Systems*, 77:547–557, 2010. 10.1140/epjb/e2010-00290-3.

[8] Y. Liu, A. Niculescu-Mizil, and W. Gryc. Topic-link lda: joint models of topic and author community. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 665–672, New York, NY, USA, 2009. ACM.

[9] A. Mccallum, A. Corrada-emmanuel, and X. Wang. Topic and role discovery in social networks. In *In IJCAI*, pages 786–791, 2005.

[10] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[11] N. Pathak, C. DeLong, A. Banerjee, and K. Erickson. Social topics models for community extraction. In *Proceedings of the 2nd SNA-KDD Workshop*, 2008.

[12] M. A. Porter, J.-P. Onnela, and P. J. Mucha. Communities in networks. *Notices of the American Mathematical Society*, 56(9):1082 – 1097, 2009.

[13] M. Sachan, D. Contractor, T. A. Faruquie, and L. V. Subramaniam. Probabilistic model for discovering topic based communities in social networks. In *CIKM*, pages 2349–2352, 2011.

[14] J. Yang and J. Leskovec. Patterns of temporal variation in online media. In *ACM Conference on Web Search and Data Mining*, 2010.

[15] H. Zhang. Hsn-pam: Finding hierarchical probabilistic groups from large-scale networks, 2010.

[16] H. Zhang, C. L. Giles, H. C. Foley, and J. Yen. Probabilistic community discovery using hierarchical latent gaussian mixture model. In *Proceedings of the Conference on Artificial intelligence*, 2007.

[17] H. Zhang, B. Qiu, C. L. Giles, H. C. Foley, and J. Yen. An lda-based community structure discovery approach for large-scale social networks. In *In IEEE Conference on Intelligence and Security Informatics*, pages 200–207, 2007.

[18] D. Zhou, E. Manavoglu, J. Li, C. L. Giles, and H. Zha. Probabilistic models for discovering e-communities. In *Proceedings of the International Conference on World Wide Web*, 2006.