Learning Linear Influence Models in Social Networks from Transient Opinion Dynamics

ABIR DE, SOURANGSHU BHATTACHARYA, PARANTAPA BHATTACHARYA, and NILOY GANGULY, Indian Institute of Technology, Kharagpur, India
SOUMEN CHAKRABARTI, Indian Institute of Technology, Bombay, India

99

Social networks, forums, and social media have emerged as global platforms for forming and shaping opinions on a broad spectrum of topics like politics, sports and entertainment. Users (also called 'actors') often update their evolving opinions, influenced through discussions with other users. Theoretical models and their analysis on understanding opinion dynamics in social networks abound in the literature. However, these models are often based on concepts from statistical physics. Their goal is to establish various regulatory phenomena like steady state consensus or bifurcation. Analysis of transient effects is largely avoided. Moreover, many of these studies assume that actors' opinions are observed globally and synchronously, which is rarely realistic. In this paper, we initiate an investigation into a family of novel data driven influence models that accurately learn and fit realistic observations. We estimate and do not presume edge strengths from observed opinions at nodes. Our influence models are linear, but not necessarily positive or row stochastic in nature. As a consequence, unlike the previous studies, they do not depend on system stability or convergence during the observation period. Furthermore, our models take into account a wide variety of data collection scenarios. In particular, they are robust to missing observations for several time steps after an actor has changed its opinion. In addition, we consider scenarios where opinion observations may be available only for aggregated clusters of nodes — a practical restriction often imposed to ensure privacy. Finally, to provide a conceptually interpretable design of edge influence, we offer a relatively frugal variant of our influence model, where the strength of influence between two connecting nodes depend on the node attributes (demography, personality, expertise etc.). Such an approach reduces the number of model parameters, reduces overfitting, and offers a tractable and explicable sketch of edge-influences in the context of opinion dynamics. With six real-life datasets crawled from Twitter and Reddit, as well as three more datasets collected from in-house experiments (with 102 volunteers), our proposed system gives significant accuracy boost over four state-of-the-art baselines. We also observe that a careful design of edge strengths using node properties is crucial, since it offers substantially better performance than the one with independent edge weights.

CCS Concepts: •**Social networks** →**xx;** •**Opinion dynamics** →*xx;* •**Linear models** →xx ; •**Influence learning** →xx;

Additional Key Words and Phrases: Social newtorks, opinion dynamics

ACM Reference format:
Abir De, Sourangshu Bhattacharya, Parantapa Bhattacharya, Niloy Ganguly, and Soumen Chakrabarti. 2017. **Learning Linear Influence Models in Social Networks from Transient Opinion Dynamics.** ***ACM Trans. Web*** **99, 99, Article 99 (2017), 31 pages.**
DOI: 10.1145/nnnnnnn.nnnnnnn

XX.

DOI: 10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

A colossal recent growth has been witnessed in the number of social media users, who use them as digital pinboards to express their opinions through extensive discussions on breaking news, political issues, sports events, celebrities, new products, etc. Thus, these platforms have come to play a crucial role in forming and shaping people's opinion on a topic. In fact, various agencies routinely use social media to tap people's opinion on the issues of interest. Naturally, modeling and estimating opinion dynamics over social networks has been studied widely by sociologists and psychologists [5, 9, 10, 16, 24, 26, 33].

In this paper we initiate a thorough study of graph models for *opinion dynamics*, where a user, modeled as a node in a social network, forms her opinion about a topic by observing the opinions of her neighbors. In general, opinion can be polarized or categorical. In our most general setup, we can capture only incomplete and asynchronous opinion readings at the nodes, and need to model and estimate the polarity and intensity of influence each node exercises on its neighbors via edges.

1.1 Prior work and limitations

Research on opinion dynamics has been initiated long before from the inception of online social networks, predominantly following models based on statistical physics [5, 9, 16, 24, 26, 33]. They were primarily designed to capture various specific phenomena in the context of opinion exchange e.g., consensus, polarization, etc. However, the parameters of such models are rarely data driven. Therefore, the weights or influences of neighbors are often set to be identical or arbitrary [9, 24, 33], without regard to observed behavior. In addition, since these models bank on some conditions catering to one or more specific scenarios, they [16, 24, 26] often implicitly favor that opinions converge and/or consensus or polarization is reached as a steady state. However, the most critical time to model social influence is arguably *before* steady state is attained — when the system is still showing *transient* behavior. The feasibility and need for studying the *transient* behavior has arisen from the large amount of user-generated content, e.g., tweets, which are now available for analysis. Subsequently, market survey has become ubiquitous on social media. For example, people's ratings on leaders are collected almost every month through various polls, rather than just before elections; the sentiment of people on various issues can be continuously assessed from the comments/tweets they post. These ratings or sentiments continuously fluctuate over time and do not really settle to a fixed value. Hence, any current state of opinion observed at some point of time is merely a transience, which quickly changes after some time. Therefore, assumptions like convergence, consensus or polarization are too restrictive, and may not reflect realistic situations. Models inherited from statistical Physics perform quite poorly in a data driven scenario, as our experimental results emphatically establish (Section 8).

Modeling influence during transience is only one aspect where idealized models fall short of reality. Another critical need is to model the ways in which opinion data can be practically captured to estimate influence models. The most desired way of such data acquisition is collecting the opinion of an individual as soon as it is updated (push mode). Such a setting would capture the most information for opinion dynamics. For corporations like Facebook or Twitter, a real time and exact copy of the data is available, and therefore, they can afford to collect data in push mode. However, in many cases where the exhaustive data collection is costly, it may be practical to collect it only intermittently (pull mode). For example, opinion about political leaders may be collected in monthly surveys; companies may estimate brand sentiments aggregated to a granularity of weeks, etc. The number of actual value updates between polls may vary widely across actors and time. E.g., people may update their opinions much more frequently before and during an election. Another crucial example of sporadic data collection is crawling from Twitter. Currently, Twitter allows only a 1% sample to be crawled free of cost, that too, within a weekly timeline. Therefore, any such collection must miss a large fraction of tweets. Any model that assumes complete or perfect knowledge of opinion updates will be quite fragile in practice. A major challenge we faced in this work is to develop influence models that are robust in the face of sporadic and incomplete data updates.

1.2 Present work

We initiate investigation into the following type of influence models. We assume that agents (nodes in a social graph) have quantitative opinions which are real continuous numbers, and these agents influence each other through the edges. In some applications (such as polls involving ratings), this numeric opinion is directly visible. In applications where users express themselves via text (e.g., Twitter, Zomato reviews) text can be converted into estimates of numeric opinion [44]. We further assume that influence is linear in nature, *i.e.* the opinion of a user changes as a linear function of the opinions of her neighbors. The weights of this linear function reflect the corresponding influence of one user on another. Such an assumption makes the proposed models tractable, explainable and learnable. However, unlike the DeGroot consensus model [16], we do not enforce a row-stochastic structure on the influence matrix or assume the existence of a steady state consensus, polarization, or fragmentation.

In addition to the above modeling choice— that considers influence for each edge independently— we also consider a relatively frugal variant of our models (having fewer parameters) where influence between two nodes mainly depends on their properties (in case of people, this might be clubbed into "personalities"). From detailed experimentation, we notice that, despite having fewer parameters, such a model provides a more accurate predictive performance than its per-edge counterpart, which is a surprising observation to us.

Subsequently, we consider a wide variety of data observation regimes described below, which requires us to devise significant modifications to the basic linear model. To that aim, we design a family of opinion dynamical models and the corresponding parameter estimation methods, where the models are several variants of the basic unrestricted linear opinion propagation system. Each of these variants works in different data observation setting, and by doing so, the proposed approach is able to flesh out the inchoate idea of a simple linear model into a complete robust modeling suit, capable of operating over different aggressive realistic setting.

Observation regimes. We assume four diverse data-collection settings. In what follows, a 'message' or 'post' is the (local) announcement of change in opinion value at a node, such as one tweet.

—*Full observation:* Here, we assume that *all* posts that are made on the timeline of the users are available.

—*Periodic observation:* Here, we assume that the data collector misses data in regular (equal) intervals.

—*Aperiodic observation:* This is a more realistic situation where we assume that, the posts are collected intermittently at irregular (not equal) intervals.

—*Mesoscaled observation:* As opposed the above three scenarios where opinions are collected for each individual, here, we consider, opinion is collected aggregated over clusters of nodes, rather than polling individual nodes.

Experimental validation. We report on a series of experiments with nine data sets to validate our influence model and estimation algorithms. Three of these were collected by running controlled, in-house, social opinion exchange processes. In these processes, we attempted to capture every opinion change of all participants. They were told to form opinions based solely on discussions with designated social network neighbors. These datasets helped us to validate our model in a *full observation* setting which is difficult to avail in practice. We collect six more data sets from Twitter and Reddit. For each of these datasets, our proposed methods offer substantial accuracy gains in predicting the opinion of users, beyond several strong baselines, for all data collection scenarios. In addition, we observe that mesoscaling can provide better performance in forecasting collective opinion of a group of users.

Summary of contributions. We make the following contributions in this paper.

— *Models for learning transient opinion dynamics:* We learn a linear opinion propagation dynamical model from observed opinion values of the individual users agents without appealing to steady state behavior. To the best of our knowledge, our framework (a) is the first that makes no assumption about consensus, polarization, or fragmentation; (b) works in a potentially transient setting, and (c) regularizes edge parameter estimates using node properties, and (d) faithfully matches real-life network observations.

—*In-house games for full observation data:* In the real world, users are influenced by multiple sources of information. Since it is practically impossible to collect all the posts from all the different sources of information that influence users, we conducted three in-house games with around 100 users who were connected in a specified network structure. They expressed their views on three topics. The users directly provided their opinion along with their messages for each post. This clean, full observation data set helps make initial assessments of models while avoiding the complications arising from missed observations.

—*Practical data-acquisition settings:* To make our proposal practically effective, we consider several realistic data-collection scenarios e.g. periodic, aperiodic or aggregated observations. Such scenarios present additional challenges to modeling and learning the transient opinion dynamics. We present experiments on several real data sets with these characteristics, based on Twitter and Reddit. We establish that our model offers significantly better performance than other existing baselines.

A preliminary version of this work can be found in [11] that only discusses about the basic version of the models excluding those of community level opinions and node features, and with fewer datasets and experimental analysis.

Organization. The following section provides a comprehensive review of the previous literature. The next three section provides the key technical expositions of this paper. More specifically, Section 3 reports the two basic influence models for opinion dynamics– one considering independent edge-weights, and another modeling the edge-weight as a function of the corresponding node properties. Section 4 describes the variants of the models in various data acquisition scenarios. In Section 5, we describe algorithms to estimate model parameters for different variants of our model. Section 7 describes the collection of real and in house datasets. Section ?? presentes the evaluation metrics. Section 8 illustrates the experimental setup and the results. Finally in Section 9, we conclude our paper.

2 RELATED WORK

While they are often used interchangeably, Merriam-Webster defines an opinion as “a conclusion thought out yet open to dispute” and a sentiment as “a settled opinion reflective of one’s feelings” [37]. In this work, we use the term ‘opinion’, in view of their dynamic nature.

Not all influence is propagated along social network edges; external events also impact agents. However, Myers *et al.* [35] developed a detailed model for blending external and social influence, and found that 71% of the information transfer volume (suitably characterized) in Twitter can be attributed to information diffusion across social links. Here we will focus exclusively on influence conveyed by social links.

2.1 Discrete opinion based approach

Discrete models assume that the opinions are discrete (binary or ordinal/quantized). The voter model [9] belongs to this category. At each step, a node is selected at random; this node chooses one of its neighbors uniformly at random (including itself) and adopts that opinion as its own. This model always leads to consensus which is rare in many social scenarios. A modified version of the voter model is called label propagation [47] where the node adopts the majority opinion from among its neighbors. However, these models always converge to consensus, irrespective of the transient dynamics.

One way to overcome such a limited outcome is to incorporate stubborn agents [46]. Another way [5] is to have each agent adopt its neighbors' opinion, but depending on the similarity with her own. This model leads to polarization instead of consensus. This was entirely a data driven study with no rigorous analysis. A further unifying variation was analyzed by Lanchier [33]. In that model, an agent adopts another agent's opinion if those opinions are within a certain distance or difference called the confidence threshold. Lanchier showed that small (large) threshold values lead to polarization (consensus) with high probability. Kempe *et al.* [29] brought forward the concept of influence-selection whereby an agent is not only influenced by other agents which has similar opinion but also selects for interaction agents who are similar to itself. They proved that such behavior can stabilize over arbitrary graphs and precisely characterize the set of all stable equilibria.

Discrete opinions are a natural model for some applications, but not others. E.g., opinion about world population at a future date, or the concentration of atmospheric CO_2, or the number/fraction of votes a politician might get, are all effectively continuous.

2.2 Continuous opinion based approach

Our present work is in the other regime of continuous opinions. Many models for continuous opinion assume, like us, that neighbors influence *linearly* the opinion of an agent [16], reaching limited consensus. Analysis is frequently grounded in the mathematics of matrix eigensystems, Physics and theoretical Biology. They are based, for example, on bird flocking models [24] and Cellular Automata [26]. In the flocking model, a node i (agent) with opinion x_i first selects the set of neighbors j having opinions x_j so that $|x_i - x_j| \leq \epsilon$, and then updates its own opinion by averaging them. A class of variants of Flocking models takes into account of random interactions between users [15, 45]. These models have shown that final distribution of opinion values across the networks strongly depends on the choice of the threshold. For example, high threshold values result in opinion convergence around the initial average opinion, whereas low thresholds yield several opinion clusters across the graphs. Moreover, these works have considered several variations of Flocking models. Yet, they observed a similar clustering behavior for a large parameter space.

There is also a large body of work (see [7, 34] and references therein) that has sought to characterize the convergence of bounded confidence dynamics to either absolute consensus or some clustering (polarization). But not all papers focus on convergence. Bindel *et al.* [6] state that in many social settings consensus may never be attained. They characterize the cost of disagreements in a game-theoretic setting. Of course, there are other occasions where only a discrete opinion model will fit, and network averaging in the continuous sense is not meaningful [8]. Agents must choose from a fixed discrete set of options. Various formulations of graphical games showed that characterizing stability even for a two-strategy game is very difficult.

We chose to address continuous opinions to retain some theoretical handle in the face of our newly-introduced complications such as possible transience and asynchronous observations. However, there are some important distinctions with earlier work that may appear similar. DeGroot [16] assumed a row-stochastic influence matrix with $w_{ij} \geq 0$, and opinions in the range $[0, 1]$ (which stochastic updates preserved). Another paper [4] aims to consider the effect of prior knowledge of the users on one topic, over the dynamics of DeGroot model. DeMarzo *et al.* in the paper [17] consider the effect of persuasion bias on opinion formation. They suggest that persuasion bias is closely connected with social influence between two users. Therefore, the influence of one user on another not only depends on the authenticity of the information she receives, but also the connection she shares with others in the graph. In other words, they infer that the persuasion bias of the users are not simply the traits of the users. Rather they are strongly correlated with the network structure. A recent work [14] attempts to combine the temporal dynamics of posts along with opinion propagation. However, it does not consider many realistic scenarios, e.g. intermittent observations or mesoscaled setting.

In our case, opinions can be unbounded, updates are not stochastic (influence can be negative, and an agent's combination rule is not convex), and zero is a special opinion value separating two polarities of opinion.

2.3 Hybrid approach

A more recent paper [10] proposes a hybrid model, somewhere between discrete and continuous. It proposes a *biased voter model*, which is a unification of the voter model with flocking. Each agent is driven by a mix of three forces: stubbornness (ignoring others' opinions), DeGroot's permissive averaging with neighbors, and biased conformance, which chooses influencing agents biased toward those whose opinions are already somewhat close to that of the base agent. A preliminary data study is used to justify the tension between these forces, and the resulting model is analyzed to the following two ends. First, even if an individual agent changes opinion continually, the relative population sizes of different opinions converge. Second, consensus still happens under certain conditions. This paper is not concerned with influence estimation on individual edges, which is our main goal.

2.4 Modeling influence in information propagation

Yet other works [30, 40] assume fixed topology and edge weights or propagation rules, and seek to select an initial set of active (or 'infected') so as to maximize some kind of cascading effect to the rest of the network. We do not seek to maximize influence; we *observe* a dynamic influence process and estimate influence strength of all edges. A set of works analyze peer pressure and also external influence in the context of information propagation [3, 39].

The vast majority of the work discussed above assume some kind of fixed influence strength on each edge. A notable exception [21], which, however, returns to the domain of some discrete action on part of one agent, that precipitates the same action in another agent at some subsequent time. Given the temporal ordering, influence propagation is acyclic, an assumption at odds with any kind of reciprocal, continual influence. But this simpler setup allows them to estimate an edge parameter $p_{v,u}$ from a form of soft-OR influence model at each node: $p_u(S) = 1 - \prod_{v \in S}(1 - p_{v,u})$, where S is the set of neighbors of u that have already committed the action, and $p_u(S)$ is compared to a threshold to decide if u should also commit it. Another notable example of influence estimation is by Shahrampour *et al.* [42], who provide a purely theoretical analysis of the online continuous case, but do not deal with asynchronous observations, or validate on real data.

2.5 Modeling influence in other contexts

Apart from information diffusion, influence modeling has been extensively researched in Bayesian network modeling [28, 36] and very recently in graph representation learning [20, 22, 25, 31, 41]. However, none of these works consider stream of networked data, and therefore cannot be applied in case of opinion propagation or information flow. Moreover, the objective of influence modeling in case of opinion dynamics is drastically different from graph representation learning, where the latter aims to embed the entire graph into low dimensional vectors called "embeddings" as opposed to opinion dynamics where the influence modulate opinion flow over the network– an entirely different scenario.

3 MODEL FORMULATION

In this section, we introduce the two key influence models which will drive the subsequent opinion models for different data collection scenarios.

— *Linear model with independent edge weights:* A linear model with latent independent edge-weights that reflect fixed user-user influences in the network.

— *Linear model with latent node labels:* A linear model where edge influence weights (including polarity) depend on latent attributes of the two connected nodes. These attributes may represent user demographies, personalities or some other properties.

At the very outset, we model opinion as an *arbitrary real number* describing an agent's opinion or sentiment on an issue, real world event, product, etc. Our notion of opinion is more akin to opinion mining or sentiment analysis (see e.g. Pang *et al.*[37]), where both polarity (+ve or $-$ve) and magnitude are important. For example, on a recently-launched product, an opinion value of $+1$, 0 and -1 could mean that the product is 'good', 'neutral', and 'bad'' respectively, while an opinion value of 1 is considered more positive than opinion value 0.1. We denote the opinion of an agent i ($i = 1, \dots, N$) at time instant k as $x_k^i \in \mathbb{R}$. Next we describe two opinion dynamics / propagation models through time.

3.1 Linear propagation model with independent edge weights

Let $G = (V, E)$ be a directed graph representing a social network. V is the set of vertices or nodes representing agents who are forming and propagating opinions ($|V| = N$). We assume opinion values of agents evolve as a linear function of their own and their neighbors' previous opinions; i.e., at time $k+1$, we have $x_{k+1}^i = \sum_{j=1}^N A_{i,j} x_k^j$, $\forall k = 1 \dots K$. $A_{i,j}$ represents the stationary weight or intensity with which, agent j's opinion x_k^j at time k influences the formation of agent i's opinion at time $(k+1)$. Further, node j cannot influence node i if they are not connected: $(i, j) \notin E \implies A_{i,j} = 0$. Also, $A_{i,i}$ represents the weight with which agent i influences itself (a measure of stubbornness). Thus, $\mathbf{A}$ can be thought as a weighted adjacency matrix of the graph G with all self loops present. Let $\mathbf{x}_k = [x_k^1, \dots, x_k^N]^T$ denote the vector of all opinions at time k. We have the following equation representing the opinion dynamics:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k \tag{1}$$

Note that for $(i, j) \in E$, $A_{i,j}$ can be either positive or negative. A negative $A_{i,j}$ implies that agent i does get influenced by j's opinion, but to the opposite polarity. As a common example from real life, person i may know that her taste in colors is the opposite of person j. Hence, person j liking a new paint may negatively influence person i's opinion about it. This effect is not possible in DeGroot's model [16], since $A_{i,j}$s are restricted to be positive and sum to 1. On the other hand, this assumption keeps the opinions predicted by DeGroot's model at time $k+1$ in the same range as the opinions in time k. The opinions predicted by the model proposed above are not thus bounded. However, it is easy to check that:

$$\|\mathbf{x}_{k+1}\| \leq \|\mathbf{A}\| \|\mathbf{x}_k\| \leq \sqrt{\lambda_{max}(\mathbf{A}^T\mathbf{A})} \|\mathbf{x}_k\|$$

where, $\lambda_{max}(\mathbf{A}^T\mathbf{A})$ is the largest eigenvalue of $\mathbf{A}^T\mathbf{A}$. Hence, $\sqrt{\lambda_{max}(\mathbf{A}^T\mathbf{A})}$ imposes a dynamic bound for the single round of update.

Another aspect of our study is that we focus on short-term or bounded-horizon opinion dynamics, as opposed to asymptotic behavior of opinion dynamics models. Therefore, we can allow the use of models for which $\sqrt{\lambda_{max}(\mathbf{A}^T\mathbf{A})} \neq 1$. In the familiar asymptotic scenario, $\sqrt{\lambda_{max}(\mathbf{A}^T\mathbf{A})} > 1$ leads to the unbounded opinions as $k \to \infty$, while all opinions shrink to 0 if $\sqrt{\lambda_{max}(\mathbf{A}^T\mathbf{A})} < 1$. The focus on short term dynamics is fueled by the thought that A_{ij}, the influence of a person j on a person i, changes with time. In the experiments, we try to predict the opinions of $(k+1)^{th}$ time point using opinions of previous k time points.

3.2 Linear propagation model with latent node types (SBLM)

The aforesaid linear model tries to learn influences for each edge in G, which represents a large number of parameters (up to $O(N^2)$) and degrees of freedom. This entails not only high computational complexity, but also lack of interpretability and potential overfitting.

In this section, we propose a model with a fewer number of parameters, which are potentially more interpretable. We assume that nodes are naturally clustered into groups. For example, in case of opinion exchange on a technical or knowledge-based topic, there may be experts and non-experts, whereas, for a political topic, there may be leftists or rightists, and so on. In our models, we conceptualize these node properties as representing cluster labels in the underlying network. In the above example, we can think political ideology (leftists and rightists) as the clustering node-property. Let there be C labels (node-properties), $\{1, \ldots, C\}$, one corresponding to each cluster. Also, let $z_i \in \{1, \ldots, C\}$ be the random variable denoting the cluster label of the i^{th} agent, $i = 1, \ldots, N$. Note that z_i is usually a latent entity. Let $\boldsymbol{\theta}_i \in [0,1]^C$ be a probability vector with its element $\boldsymbol{\theta}_i(j)$ being the chance that node i belongs to cluster j. That is, $\boldsymbol{\theta}_i(j) = P(z_i = j)$. Hence, $\sum_{j=1}^{C} \boldsymbol{\theta}_i(j) = 1$. In other words, we assume a time-invariant probabilistic model for cluster membership: z_i follows a multinomial distribution with parameters $\boldsymbol{\theta}_i$.

Further, we can assume that agents belonging to a group display similar behavior towards the phenomenon of opinion propagation. For example, all experts have similar influence on other experts or non-experts. Let $B_{l_1,l_2} \in \mathbb{R}, 1 \leq l_1, l_2 \leq C$ denote the influence of a member of the l_1^{th} cluster on a member of the l_2^{th} cluster. The random influence of j^{th} agent on i^{th} agent is given by $A_{ij}|(z_i, z_j) \sim \mathbb{D}(\boldsymbol{\xi}_{z_i}^T \boldsymbol{B} \boldsymbol{\xi}_{z_j})$, where $\mathbb{D}$ is a pre-specified distribution and $\boldsymbol{\xi}_z$ is the one-hot representation of the cluster label. Since $\mathbb{E}(\boldsymbol{\xi}_{z_i}) = \boldsymbol{\theta}_i$, one may write $A_{ij} \sim \mathbb{E}_{z_i,z_j}[\mathbb{D}(\boldsymbol{\xi}_{z_i}^T \boldsymbol{B} \boldsymbol{\xi}_{z_j})]$. Finally, an opinion stream $\mathbf{x}_k,\ k = 1, \ldots, K$ is generated as $\mathbf{x}_{k+1} \sim \mathcal{N}(A\mathbf{x}_k, \sigma^2)$. Hence, the final generative model is written as:

$$z_i|\boldsymbol{\theta}_i \sim \text{Multinomial}(\boldsymbol{\theta}_i) \qquad \forall i = 1, \ldots, N \tag{2}$$

$$A_{ij}|(z_i, z_j) \sim \mathbb{D}(\boldsymbol{\xi}_{z_i}^T \boldsymbol{B} \boldsymbol{\xi}_{z_j}) \qquad \forall i, j = 1, \ldots, N \tag{3}$$

$$\mathbf{x}_{k+1} \sim \mathcal{N}(\boldsymbol{A}\mathbf{x}_k, \sigma^2) \qquad \forall k = 1, \ldots, K \tag{4}$$

In practice, $\boldsymbol{\theta}_i$, the apriori cluster-membership probability vector is not known. Therefore, following the traditional approaches to stochastic block modeling [1], we assume that $\boldsymbol{\theta}_i \sim \text{Dir}(\boldsymbol{\alpha})$ for all $i \in V$ with $\boldsymbol{\alpha}$ being the concentration parameter-vector of a Dirichlet distribution. It is interesting to note that if $\mathbb{D}(\mu)$ is a normal distribution with mean μ, then $\mathbb{E}(A_{ij}) = \boldsymbol{\theta}_i^T \boldsymbol{B} \boldsymbol{\theta}_j$. However, for other distribution we estimate the expected edge-influence by taking average over a large number of simulated values. Such a model for generation of cluster influences is inspired by mixed-membership or weighted stochastic block models [1] introduced recently. Hence, we call this the **stochastic block linear model** (SBLM). In Section 5, we will describe algorithms for estimating parameters

$\boldsymbol{B}$ and $\boldsymbol{\theta}_i, i = 1, \ldots, N$. The number of parameters is $O(CN + C^2)$, which is much less than linear model ($O(N^2)$) described in the previous section.

4 DATA ACQUISITION SCENARIOS

The models proposed above are specified by the set of parameters: $A_{i,j} : i, j \in \{1, \ldots, N\}$ for the linear model with independent edge-weighting, and $B_{u,v} : u, v \in \{1, \ldots, C\}$ with $\boldsymbol{\theta}_i : i \in \{1, \ldots, N\}$ for linear model with latent node type SBLM. Usually, there can be no direct observation of these parameters in a real social network. However, it is possible to obtain the opinions of various agents x_k^i at different time instants (see Section 6). In our application, agents express their opinion continually via *posts*, which can take the form of ratings, tweets, or comments. Textual posts can be converted into numeric opinion via sentiment analysis [37]. We use the dynamics of these opinions to estimate the parameters. Thus, the problem of automatically learning the parameters $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{\theta}_i$, given the observed x_k^is, is critical. In this section, we explore various scenarios in which opinion data can be acquired. Figure 1 shows an illustrative explanation of the three data collection scenarios, viz., full, periodic and aperiodic data acquisitions. Before describing them, we first describe one *hypothetical* data collection scenario, called *omniscient* data observation in the following.

Omniscient observations. Note that, as described in Figure 1, with each of these data collection regimes, we associate a data collection scenario called *omniscient* data collection–where posts from all the users together are available in each time-step. However, this is an extremely ideal situation–since, in practice, the messages are posted asynchronously, and at each time-step, only one message is available.

Translating the observed data into omniscient-like stream. Our basic dynamical model (Eq. 1) also operates similarly, since it updates the opinions of *all* the users together in each timestep. Hence, to make our model operable and trainable for the observed dataset, it is necessary to translate the observed data into the omniscient-like stream. Therefore, it is crucial to assign opinions of *all* the users in each timestep even if the some of the opinions are not observed in the collected data. To that aim, we assign the opinion x_k^u of user u at time k as the last opinion she posted a user, if she has not posted at time k. More specifically, our basic opinion model can be described as following.

In general, the opinions are posted asynchronously, i.e., at time k, one agent j may post his opinion, whereas another agent i may not. Let S be the set of all time instants when some agent has posted his opinion. Moreover, let $S_i = \{k | x_k^i \text{ exists}\} \subseteq S$, $\forall i = 1, \ldots, N$, be the set of all time instants when agent i has expressed an opinion. Also, let x_{k-}^i, be the last posted opinion by an agent i before and excluding time k.

$$
\begin{aligned}
x_k^i = A_{i,i} x_{k-}^i + \sum_{j \in \mathcal{N}(i)} A_{i,j} x_{k-}^j = \mathbf{A}_i^T \mathbf{x}_{k-} \text{ , } \forall k \in S_i \text{ and } 1 \leq i \leq |V| \\
\implies \mathbf{y}_{k+1} = \mathbf{A} \mathbf{y}_k \qquad (5)
\end{aligned}
$$

where $\mathcal{N}(i)$ is the set of neighboring vertices of i, $\boldsymbol{A}_i^T$ is the i-th row vector of matrix $\boldsymbol{A}$, and $\mathbf{y}_k$ is defined as follows.

$$
y_k^i = \begin{cases} x_k^i & \text{if } k \in S_i \\ x_{k-}^i & \text{if } k \notin S_i \end{cases}
$$

In the following, we describe the actual data collection scenarios. For each of these cases, we suitably translate the observed data as well as the underlying model as described above. From now on, we describe $\mathbf{y}_k$ as the observed opinion during k^{th} time-step in the data.

4.1 Full observations

Here, we assume that all the posts that are made on the timeline of the users are available. This is an ideal situation, when an exact and exhaustive set of opinions posted is available over time.

> Crawler does not miss data at all. Each post made on any user's timeline is present in the collected data.

It is a rare scenario in practice, but a potentially useful baseline on which to evaluate any influence estimation model. Therefore we conducted three in-house *social influence games* where users were connected in a small network to exchange their views on some controversial topics. The datasets collected from these in-house systems offer access to all the posts of the users, providing the most favorable scenario in which we can evaluate a proposed system. For such system, the effective dynamics is same as Eq. (5).

4.2 Periodic observations

The full observation setting described above, despite being simple and elegant, is practically difficult to achieve. For example, Twitter witnesses over 500 million tweets per day[1]. A crawler can collect only 1% sample from Twitter. Only a few posts of each user are likely to be collected within this budget. The ones collected are likely to skip irregular numbers of other posts. In this section, we assume that the data collector omits data in regular (equal) intervals.

> Crawler misses data at regular (equal) intervals. Between two consecutive readings in the data, we miss t posts, where t is constant throughout the timeline.

That said, in this model, we assume that the crawler misses posts with a constant frequency, say t per time window between any two consecutive posts made in k and $k+1$[2]. This means that, between two observed time-stamps opinion propagated t times across various users in the network. Thus, using simple calculations, we can write the propagation model as:

$$\mathbf{y}_{k+1} = \mathbf{A}^t \mathbf{y}_k \quad \forall k = 1, \ldots, K \tag{6}$$

Here, $\mathbf{A}^t$ is the influence matrix $\mathbf{A}$, defined previously, raised to the t-th power. We refer to this model as the **periodic linear model** (PLM). As is well known, A^t_{ij} aggregates over all paths of length t between nodes i and j.

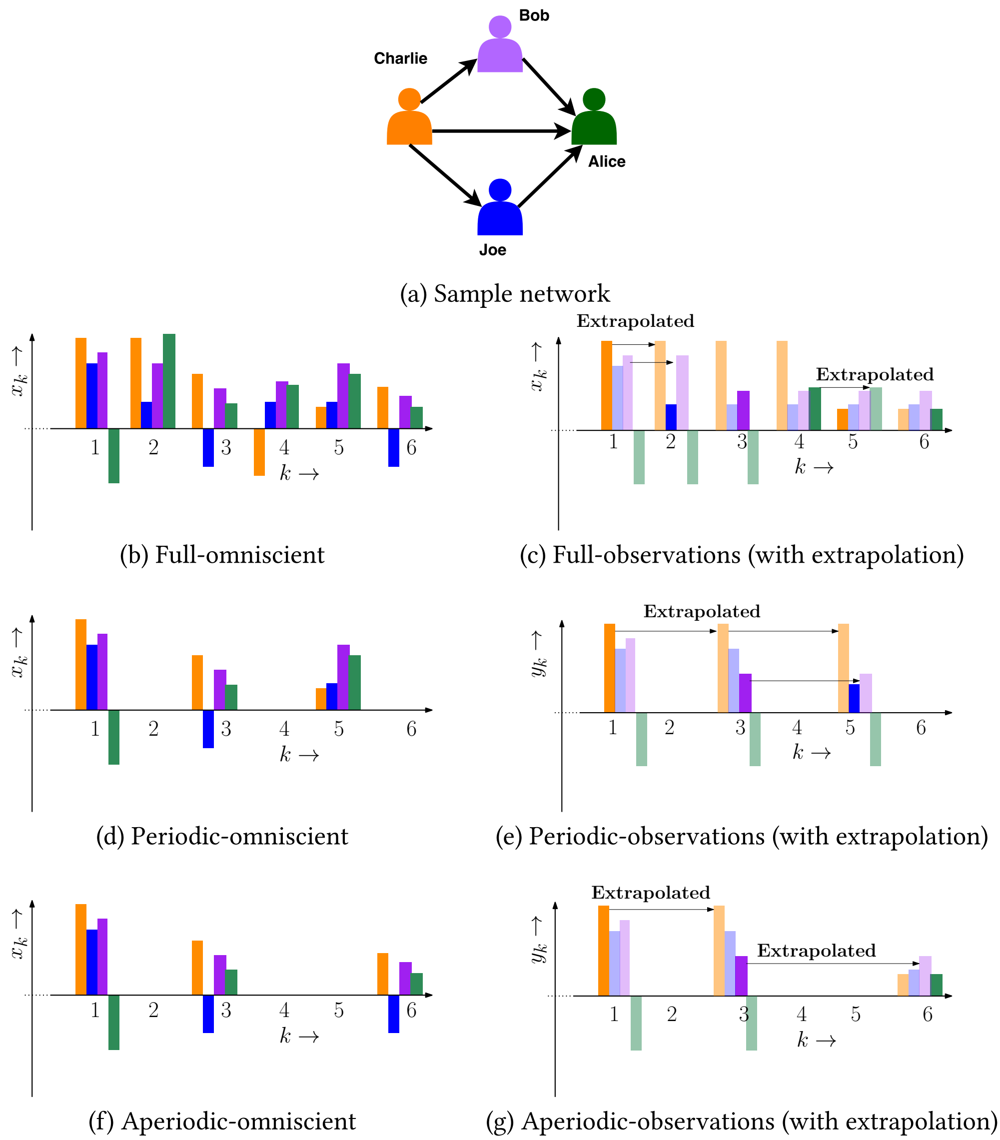
4.3 Aperiodic observervations

In PLM, we assumed a constant period, t (equivalent to the maximum number of messages missed between two consecutive time-stamps), for all consecutive time steps. However, human activities happen in bursts. For example, people post more messages on social network during the day, than at night. Hence, it is expected that number of missing messages posted during daytime would be more than at night.

> Crawler misses data at irregular intervals. Between two consecutive timesteps (k and $k+1$), a non-constant number t_k messages are missing in general.

[1] http://www.telegraph.co.uk/technology/twitter/9945505/Twitter-in-numbers.html

[2] The corresponding time-window can be define as $(k,\ k+1)$.



(a) Sample network

(b) Full-omniscient

(c) Full-observations (with extrapolation)

(d) Periodic-omniscient

(e) Periodic-observations (with extrapolation)

(f) Aperiodic-omniscient

(g) Aperiodic-observations (with extrapolation)

Fig. 1. Illustration of various data acquisition scenarios. Panel (a) shows a hypothetical network of users for which opinions are shown in figures (b) to (g). The users are color coded, and the color of the bars in (b) through (g) correspond to the color coding of the users. Panels (b), (d) and (f) show omniscient timelines, that is the hypothetical case, where we would know the opinion of every user at every time step. However, since that is not possible, we translate real *observations*, where opinions of only a subset of users is available for a subset of time steps, into an *omniscient-like stream*, by extrapolating from previous opinions, that we call actual timelines. This is shown in Panels (c), (e), and (g). Without loss of generality, suppose, at each time-stamp, only one user posts her opinion. If a user has not posted, but one of her neighbors have posted, in a given timestep, then her opinion is taken to be the same as her last observed opinion. Panels (b) and (c) show data for full observations where all posts that are made by all users are available. Panels (d) and (e) consider data for the periodic case, which indicate that between any two consecutive observations, two posts are missed. Panels (f) and (g) consider data for the aperiodic case, where the number of missing observations between two consecutive observed messages varies across time.

In *aperiodic linear model* (ALM), we assume that the number of missing posts between two consecutive time-windows varies from one time window to another. As before, let $\mathbf{y}_k$ denote the opinion vector for all agents at time k. Let $t_k, k = 1, \dots, K$ be the number of times opinion propagates during k^{th} time-window. The opinion dynamics is then given by:

$$\mathbf{y}_{k+1} = \mathbf{A}^{t_k}\mathbf{y}_k, \forall k = 1, \dots, K \tag{7}$$

Note that the model is characterized by parameters $t_k, k = 1, \dots, K$, in addition to the weighted adjacency matrix parameters A. The set of parameters $\mathcal{T}_K = \{t_k | k = 1, \cdots, K\}$ is called the *skip set*, with t_k denoting the number of iterations, which has been "skipped" at k^{th} time-window. We denote the above model as **aperiodic linear model** (ALM).

4.4 Mesoscaled data acquisition

In Physics and Meteorology, *mesoscale* or *mesoscopic* analysis refers to an intermediate scale between the finest (microscopic) and coarsest (macroscopic) observable levels of analysis or observation. In many real-life scenarios about opinion dynamics, fine-grained per-node opinion data may not be feasible or profitable to acquire. For example, it is expensive to collect individual ratings for a particular movie from everyone exiting a theater, or individual political sentiments from exit polls at elections. Instead, it may be easier to acquire an aggregated or average opinion for a group or community c of people, e.g., box-office estimate in one movie theater, or sampled political sentiments of people in a polling booth. Aggregated observations may also be regarded as protecting the privacy of individuals.

In this scenario, opinion evolves under the usual setting as described in Equation (1). However, the individual node opinions are not accessible. Instead, one can only observe the mesoscaled community level opinions $\bar{x}_k^c$, where $\bar{x}_k^c$ is the average of sentiments of all nodes in community c at time-stamp k.

5 MODEL PARAMETER ESTIMATION

Our final step is to estimate the parameters in models described in Section 3 from data acquired in scenarios described in Section 4. Here we describe formulations and algorithms for model parameter estimation in the most common or important scenarios. Note that, it is not known apriori if the collected data complies with periodic and aperiodic observations, as well as the corresponding number of missing updates. Therefore, we obtain the number of missing opinions per time-window (t for periodic and t_ks for aperiodic) using cross-validation. Hence, the time-intervals obtained via grid search refers to the estimate of the unobserved variable: number of events that occured between any two observed events. This cannot be directly estimated from the crawl schedule.

5.1 Estimation from full observartions via regularized least squares (RLS)

Our objective is to estimate the matrix $\mathbf{A}$ from opinions acquired using the asynchronous scenario (Section 4.1). Let $\mathcal{D} = \{x_k^i | k \in S_i, i \in V\}$ be a dataset of all opinions posted by all agents in V. We assume that agent i forms its opinion at time $k \in S_i$ based on previously posted opinions of its neighbors. Then, the loss incurred in predicting all observations by agent i is given by $\sum_{k \in S_i} ||x_k^i - \mathbf{A}_i^T \mathbf{x}_{k-}||^2$. Adding an L_2 regularizer, $\lambda||\mathbf{A_i}||^2$, we can estimate the optimal parameter $\mathbf{A}_i^*$ by solving the following problem:

$$\min_{\mathbf{A_i}} \sum_{k \in S_i} ||x_k^i - \mathbf{A}_i^T \mathbf{x}_{k-}||^2 + \lambda||\mathbf{A}_i||^2 \tag{8}$$
$$\text{s.t.} \quad A_{i,j} = 0 \text{ whenever } (i,j) \notin E \text{ and } i \neq j$$

Here, λ is the user defined regularization parameter and $A_{i,j}$ is the j^{th} entry of vector $\mathbf{A}_i$. By solving $|V|$ such optimization problems (one for each i), we can obtain $\mathbf{A}_i^*$, the optimal value of $\mathbf{A}_i$, for all $i = \{1, \ldots, N\}$, and thus estimate the entire optimal $\mathbf{A}$ matrix $\mathbf{A}^*$.

Let $\tilde{x}_{k-}^i = U_{ij}\mathbf{x}_{k-}, \forall k \in S_i$, where U_{ij} is a $N \times N$ diagonal matrix such that $U_{ij}(j,j) = 1$ if $(i,j) \in E$. Also, let $\mathbf{X}^i = [\tilde{x}_{k-}^i | k \in S_i]^T$ be a $|S_i| \times N$ matrix with rows as $\tilde{x}_{k-}^T$, and $\bar{x}^i = [x_k^i | k \in S_i]^T$ is a $|S_i| \times 1$ column vector. The above problem is same as solving $\mathbf{A}_i^* = \operatorname{argmin}_{\mathbf{A}_i}(\|\bar{x}^i - \mathbf{X}^i\mathbf{A}_i\|^2 + \lambda\|\mathbf{A}_i\|^2)$. It is easy to check that this problem is solved when:

$$\mathbf{A}_i^* = ((\mathbf{X}^i)^T\mathbf{X}^i + \lambda\mathbf{I})^{-1}(\mathbf{X}^i)^T\tilde{x}^i \tag{9}$$

Increasing λ decreases $\|\mathbf{A}^*\|_F$, which can be thought of as a measure of complexity of the model [43]. Here, $||\mathbf{A}^*||_F = \sqrt{\text{Trace}(\mathbf{A}^{*T}\mathbf{A}^*)}$ is the Frobenius norm of $\mathbf{A}^*$.

5.2 Periodic estimation

Following the assumptions laid out earlier in this section, we can write the regularized loss function for learning $\mathbf{A}$, in case of periodic opinion propagation (Section 4.2) as $L(\mathbf{A}) = \sum_{k=1}^{K} \|\mathbf{y}_{k+1} - \mathbf{A}^t\mathbf{y}_k\|^2 + \lambda\|\mathbf{A}^t\|^2$. The best estimate of $\mathbf{A}$ can be obtained by minimizing $L(\mathbf{A})$. Unfortunately, $L(\mathbf{A})$ is not convex in $\mathbf{A}$. Hence, the minimization can get stuck in local minimum. Also, we note that for most prediction tasks, we only need to estimate $\mathbf{M}_t = \mathbf{A}^t$, since we only observe opinions $\mathbf{y}_k$ which are propagated with the constant frequency of t per time window. Let $G^t = (V, E^t)$ be the graph generated by including all t-hop connections in the set of edges E^t. It is clear that $M_t(i,j) = 0$ if $(i,j) \notin E^t$. We can learn the optimal $\mathbf{M}_t^*$ by solving:

$$\min_{\mathbf{M}_t} \sum_{k=1}^{K} \|\mathbf{y}_{k+1} - \mathbf{M}_t\mathbf{y}_k\|^2 + \lambda\|\mathbf{M}_t\|^2 \tag{10}$$
$$\text{s.t.} \quad \mathbf{M}_t(i,j) = 0, \ \text{whenever } (i,j) \notin E^t$$

One way of obtaining $\mathbf{A}^*$ from $\mathbf{M}_t^*$ is to calculate $\mathbf{A}^* = (\mathbf{M}_t^*)^{1/t}$ using a root-finding algorithm [27].

5.3 Aperiodic estimation

Given a set of opinions, $\mathbf{y}_k, \ k = 1, \ldots, K$ and a skip-set $t_k, k = 1, \ldots, K$, analogous to previous discussion, we can write the following optimization problem for learning the weighted adjacency matrix parameter using the squared error as:

$$\min_{A} \sum_{k=1}^{K} ||\mathbf{y}_{k+1} - A^{t_k}\mathbf{y}_k||^2 + \lambda\|A\|_F^2 \tag{11}$$
$$\text{s.t:} \quad A_{i,j} = 0 \text{ whenever } (i,j) \notin E \text{ and } i \neq j$$

The above problem is also a non-convex optimization problem. However, since the feasible set is convex, we can find a local optimum for the above problem using the projected gradient descent method. Let $\mathcal{Y}_K = \{\mathbf{y}_k | k = 1, \cdots, K\}$ be the set of all opinions. Let $f(A; \mathcal{Y}_K, \mathcal{T}_K) = \sum_i ||\mathbf{y}_{k+1} - A^{t_k}\mathbf{y}_k||^2 + \lambda\|A\|_F^2$. The gradient of $f(A)$ w.r.t. A can be written as

$$\begin{aligned}\nabla_A f(A; \mathcal{Y}_K, \mathcal{T}_K) = \sum_i t_k \big[&- 2A^{t_k - 1}\mathbf{y}_k\mathbf{y}_{k+1}^T + \mathbf{y}_k\mathbf{y}_k^T(A^{t_k})^T A^{t_k - 1} \\ &+ A^{t_k - 1}\mathbf{y}_k\mathbf{y}_k^T(A^{t_k})^T\big] + 2\lambda A.\end{aligned} \tag{12}$$

The projected gradient descent algorithm for finding optimal A is described in Algorithm 1. Here, the gradient matrix $\nabla_A f(A; \mathcal{Y}_K, \mathcal{T}_K)$ is evaluated using expression in Equation 12. The projection step $\Pi(A, E)$ ensures that resulting A is projected back to the feasible set, i.e., $A_{ij} = 0$ if $(i, j) \notin E$. While in general the algorithm is not guaranteed to converge to the global optimum, in practice it converges quickly to a local optimum.

ALGORITHM 1: Learning A using projected gradient descent.

Data: $G = (V, E)$.
Input : Opinion-vectors $\mathcal{Y}_K$, skip-sets $\mathcal{T}_K$, initial A_0, convergence threshold ϵ, edge set E
Output : Weighted-adjacency matrix A
initialize: $A \leftarrow \gamma A_0$
while $(||\nabla_A f(A; \mathcal{Y}_K, \mathcal{T}_K)|| \geq \epsilon)$ **do**
  $A \leftarrow A - s\nabla_A f(A; \mathcal{Y}_K, \mathcal{T}_K)$;
  $A \leftarrow \Pi(A, E)$
end
Return A

Note that here we assume the skip set $\mathcal{T}_K$ to be given. In practice, we can restrict each t_k to take values from a set $\{1, \cdots, t_{max}\}$, which can be optimized using cross-validation.

5.4 Parameter estimation with mesoscaled data (MLearn)

Let S be the set of all time instants when the mesoscaled opinions are observable. Let the input graph be $G(V, E)$, the set of communities be $\mathcal{C}$, community-level mesoscaled opinions be $\bar{x}_k^c$ with $c \in \mathcal{C}$ and $k \in S$. Our task in this case is to estimate the edge weight matrix $\boldsymbol{A}$. We cast this problem as the following optimization problem.

$$\min_{\mathbf{A}} \sum_{i \in V} \sum_{k \in S} ||x_k^i - \mathbf{A}_i^T \mathbf{x}_{k-1}||^2 + \lambda ||\mathbf{A}_i||^2$$
$$\text{s.t. } A_{i,j} = 0 \text{ whenever } (i, j) \notin E \text{ and } i \neq j.$$

Here, given any $c \in \mathcal{C}$, $x_k^i = \bar{x}_k^c$ for all $i \in c, k \in S$. We call this framework as MLearn.

5.5 Stochastic block model estimation with node types

Let $\mathbf{y}_k,\ k = 1, \ldots, K$ be the opinions acquired in a periodic/aperiodic data collection setting. Here we attempt to learn the edge influence *and* cluster memberships from these temporal data, which are assumed to be generated following the generative model described in Section 3.2.

Recall that the cluster membership probability vector $\boldsymbol{\theta}_i$ for each node $i \in V$ is drawn from $\text{Dir}(\boldsymbol{\alpha})$. ($\boldsymbol{\alpha}$ is the concentration parameter vector for the Dirchlet cluster distribution.) The cluster membership indicator vector is $\boldsymbol{\xi}_{z_i}$ for each node i with cluster-label z_i. The edge influence from i to j is $A_{i,j}$. The block interaction matrix is $\boldsymbol{B}$.

Given opinion data $\mathbf{y}[1 : K]$, our task is to infer all these unknown parameters $\boldsymbol{\theta}[1 : N],\ z[1 : N],\ \boldsymbol{A}$. Note that, only $\boldsymbol{B}$ and $\boldsymbol{\alpha}$ are independent parameters; other variables are all latent. The proposed model is a variant of stochastic block model. However, unlike in existing work [1], our setting does not offer access to edge-influence values directly. Instead, we observe only a stream of temporal data, generated linearly from previous opinions using the hidden edge influences. Hence, the existing inference techniques for stochastic block model cannot be directly applied here.

To estimate the parameters $\boldsymbol{\Lambda} = \{\boldsymbol{\theta}, A, z\}$, first we compute the likelihood of the opinion stream. Combining the opinion model with the other sources of stochasticity, we write the joint model for opinions and graph parameters as:

$$\begin{aligned}&\Pr((\mathbf{y}_k)_{k=1}^K, \mathbf{\Lambda}|\boldsymbol{\alpha}, \boldsymbol{B})\\&\quad\propto \Pr((\mathbf{y}_k)_{k=1}^K|A)\ \Pr(A|(z_v)_{v\in V}; \boldsymbol{B})\ \Pr((z_v)_{v\in V}, \boldsymbol{\theta}|\boldsymbol{\alpha})\\&\quad= \exp\left[\frac{-\sum_{k=1}^K ||\mathbf{y}_{k+1} - A^{t_k}\mathbf{y}_k||^2}{\sigma^2}\right] \prod_{(u,v)\in E} \mathbb{D}(\boldsymbol{\xi}_{z_u}^T \boldsymbol{B}\boldsymbol{\xi}_{z_v}) \prod_{u,c\in V\times[1:C]} \boldsymbol{\theta}_u(c)^{\boldsymbol{\xi}_{z_u}(c)}\,\mathrm{Dir}(\boldsymbol{\theta}|\boldsymbol{\alpha})\end{aligned} \tag{13}$$

There are lot of techniques to solve graphical models and their variants, and we appeal to variatioanl inference (directly adopted using [1]). Choosing different forms for $\mathbb{D}$, e.g., normal, exponential, and pareto distributions, we estimate A_{ij}, which in turn is used to predict the opinion in the next-time stamp (Section 8). In the next section, we describe the dataset construction and metrics used.

Dataset	# Nodes	# Edges	# Messages	Max messages./Node	Min messages./Node
Continents: Europe vs. North America	102	1,020	2,182	52	6
Colleges: IIT[3] Delhi vs. IIT Bombay	102	1,020	1,758	40	3
Occupation: Startup vs. Job	102	1,020	1,439	33	4
Twitter: Delhi elections	548	5,271	20026	102	20
Twitter: Movie	457	4886	14016	236	21
Twitter: Series	947	10253	13203	291	20
Twitter: Fight	848	10118	21526	402	21
Twitter: Bollywood	1031	34952	46845	867	22
Reddit (politics network)	556	94,312	64366	2,571	20

Table 1. Summary of the nine datasets used for experimental validation. The first three correspond to the topics used for in-house controlled experiments on human subjects. The last six correspond to real world datasets obtained from Twitter and Reddit.

6 DATASETS

We use nine diverse datasets to evaluate our algorithm. For each, we require the network topology, and the opinion values of the users over a period of time. The datasets, summarized in Table 1, can be placed in two groups. The first three are generated by us, in-house, through carefully controlled and monitored social influence processes. The last six are derived from Twitter and Reddit forums, provided as-is. The distinction is that in the first three cases, we are able to read opinion values at the time granularity of our choice, so as not to miss any updates; whereas for the last six, we have no such control. The first three cases provide us with valuable insights, as in these cases we were able to capture all visible opinion values, while also minimizing the influence of external sources.

6.1 Controlled social experiments

The set of agents in our controlled experiments consists of a class of 102 students in a course on *Information Retrieval* taught in the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur. The experiments were performed in a laboratory setting, where each student sat in front of a desktop computer and interacted with ten other randomly assigned students (designated social neighbors) through a Web interface (as shown in Figure 2) for a period of three hours. (In order to maintain both connectivity as well as randomness of the social graph with a modest number of nodes, a realistic degree distribution like power-law could not be considered.)

On each topic, the agents refined their opinions continually by communicating with their graph neighbors using the text box. To avoid externalities, participants were not allowed to access the

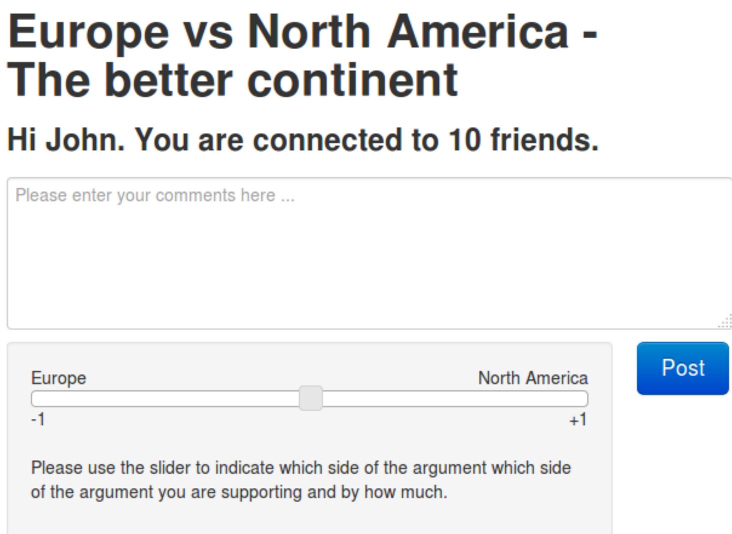


Fig. 2. Web interface for opinion posting for the controlled experiment.

Web, or discuss anything with each other verbally. All communication through the interface was recorded. Social neighbors were kept anonymous, so that the agents did not get biased by the real-life identity of another agent.

In order to collect one dataset, we started by broadcasting to agents a **topic**, posed in the form of a comparison between two entities. The three topics given to the students involved these comparisons:

Continents: The better place to live: Europe vs. North America.
Colleges: The better college to attend: IIT Delhi vs. IIT Bombay.
Occupation: The more preferable occupation: startup vs. a regular job.

These topics were chosen since most agents did not have a strong prior opinion, but had some knowledge about the subjects. This was done to ensure that at least some of the agents would show changes in their opinions during the experiment. Every time an agent posted a message, the interface automatically reported the current opinion value, which was modeled as a real number in the range $[-1, 1]$. The sign represents polarity of the opinion (e.g., if joining a startup is preferred, then the opinion score assigned tends to -1, while the reverse is true for the other choice), and the magnitude represents the degree of conviction. Only the message from an agent, and not his/her current quantitative opinion, was shown to neighboring agents. Agents were asked to make opinion messages self-contained. Every experiment proceeded for one hour, after which the experiment was terminated. At the end of a live experiment, we obtained one dataset, containing all visible timestamped opinions of every agent.

6.2 Twitter datasets

Via hashtags, we chose five controversial Twitter topics (an election, a movie, a TV series, a boxing match, and a celebrity hit-and-run case) and crawled related tweets during a period of intense activity. This provided us with a very good opportunity to measure the performance of our system.

6.2.1 The Network. For each topic, we filtered the candidate set of agents in three steps. We started with around a million tweets. In order to remove corporate accounts, bot accounts, and spammers, we filtered the set of users based on the number of followees, number of followers, and the number of tweets posted by the user. We only preserved those users who had between 100 and

10,000 friends, between 50 and 1,000 followers, and between 200 and 10,000 tweets posted during the account's lifetime. This resulted in a set of a good number of users who are active on Twitter and also enthusiastic about the topic. For these users, we collected, using the Twitter REST API, the user IDs of all their followees, followers, and up to ∼3K most recent tweets. We only collected tweets posted during the week of occurrence of the concerned event. With the information about both the followees and followers of the pre-selected users, we were able to create the complete follower-followee network. Finally, from these users, we selected the largest strongly connected component such that each selected user posted over 20 tweets. The network thus generated is afterwards used to test our system.

6.2.2 Opinion values. Since tweets are limited to only 140 characters, we accumulated the tweets posted by every user during a single hour, and generated a document. Each document was turned into an opinion score. 'Opinion' here connotes a positive or negative attitude to the particular issue/event, which was detected by subjecting these hourly documents to a sentiment analysis tool specifically designed for Twitter [23]. The method relies on scoring tokens based on their co-occurrence with positive emoticons such as smiley " :) " or negative emoticon or frowny face " : (". Prior work has shown the efficacy of using emoticons [18] for sentiment detection. For example, if in our dataset, we find the word 'love' to co-occur in x tweets containing the smiley " :) " and to co-occur with y tweets containing the frowny face " : (", the sentiment score given to the word 'love' according to the algorithm is equal to $x/(x+y)$. This gives the relative propensity of the token to be used in a positive content. To get a clean set of scored sentiment tokens, we only used tweets that were written in English, and only considered tokens that occurred at least 20 times in our dataset. For every document we finally obtained a single sentiment score in the range $[-1, 1]$. The score represents the relative proportions of words with positive and negative connotations.

6.2.3 Collected Twitter Data. We gathered the following Twitter datasets for testing our proposals. The details of the datasets are given here and also summarized in Table 1.

• **Delhi Elections 2013 (Tw:Politics).** The Delhi Legislative Assembly elections of 2013 was a keenly contested event with three major parties (two old parties, BJP and Congress, and one newly formed party, AAP) winning roughly equal vote share. For testing our system, we used the Twitter search API to collect tweets containing the following hashtags: #BJP, #AAP, #Congress, and #Polls2013. The first three represented the hashtags for the three major parties competing in the elections, while the fourth was the most popular hashtag corresponding to the event. We gathered tweets during the period of 9th to 15th December 2013. This period corresponds to the week following the declaration of results on 8th December 2013. The obtained dataset has around 20K posts and a connected graph having 548 users and 5.2K edges.

• **Release of "Avengers: Age of Ultron" (Tw:Movie).** This superhero movie was released in the first week of May 2015. We considered the following hashtags: #Ultron, #marvel, #avengers, and collected the tweets during the period of 28 April to 5 May 2015. The resulting network has 487 users and 4.8K edges and around 14K tweets. One important aspect of this dataset is that all the collected users have positive opinion in all the posts.

• **Season 6 of "The Game of Thrones" (Tw:Series).** The sixth season of this American thriller-drama was first aired on April 12, 2015. We collected the tweets with hashtags #GOT and #gameofthrones during the period of April 8 to 15, 2015 that resulted in more that 21K posts and a network of 947 users and 10k edges.

• **Boxing match between Floyd Mayweather, Jr. and Manny Pacquiao (Tw:Boxing).** This boxing match was a much-hyped event often billed as "The Fight of the Century". This event took place on May 2, 2015. It triggered a huge discussion in Twitter. We gathered the related tweets from 29 April to 7 May 2015 that led a rich collection of 21K messages and a network of 848 users and 10K edges.

• **Bollywood actor hit-and-run case verdict (Tw:Bollywood).** This controversial event is the final hearing on the hit-and-run incident by Salman Khan, a popular Bollywood actor. This event triggered an intense war-of-words among many users, some openly supporting Khan. We collected the tweets with the related hashtags: #Salman, #HitAndRun etc., during the period of 7 May to 16 May 2015. Finally, we obtained a corpus of 20K tweets with a network having $\sim$ 1K nodes and $\sim$ 46K edges.

6.3 Reddit Politics Data

Reddit is a social post curation website, where users submit content in the form of text posts or links to websites with the content. More than 6% of online adult users use Reddit[4]. Content in Reddit is categorized by areas of interest called 'subreddits'. Reddit boasts over seven thousand active subreddits[5] on topics as varied as music, politics, sports, world news, programming, etc.

We collected data of Reddit users who posted content in the subreddit 'politics' during the period of July 1 to December 31, 2012. We crawled all posts made by Reddit users during the above period in the subreddit politics. We obtained 120K posts made by 31K users.

6.3.1 The Network. The social network in Reddit is not explicit. We applied certain heuristics to recover an approximate user network. We created an undirected network taking 31K users as vertices, and assumed the existence of an edge between two users if there existed two subreddits (other than politics) where both posted during the given time period.

Similar to the case of the Twitter data, we randomly selected approximately 500 users such that the users have made more than 20 submissions during the given period and the network between them formed a single connected component. We ended up selecting a subnet of 556 users for the subsequent experiments.

6.3.2 Opinion Values. Most of the posts made by users of Reddit were in well formed English. Thus, we used the standard linguistic analysis tool LIWC [38] to analyze sentiment scores from them. We computed the sentiment of a post as the difference between the positive emotion score and the negative emotion score, as returned by LIWC. The results were normalized by mapping the range of values obtained to the range $[-1, 1]$ using linear scaling.

7 EVALUATION METRICS

In this paper, we adopt a data driven approach to opinion modeling. To this end, we assume that we have access to actual opinions (ground truth) expressed by people interacting on the social network (see Section 6). We estimate edge-influences (Section 5) under diverse settings, which are then used to predict future opinions. We evaluate the utility of our proposal by measuring the deviation of the predicted opinion from actual opinion. If $\mathbf{y}_k \in \mathbb{R}^{|V|\times 1}$ is the opinion vector expressed by users at timepoints $k = 1, \ldots, K$, the predicted opinion vector

$$\hat{\boldsymbol{y}}_{k+1} = \begin{cases} \hat{\boldsymbol{A}}\mathbf{y}_k & \text{if models is FLM} \\ \hat{\boldsymbol{A}}\mathbf{y}_k^t & \text{if models is PLM} \\ \hat{\boldsymbol{A}}\mathbf{y}_k^{t_k} & \text{if models is ALM} \end{cases}$$

[4]http://pewinternet.org/Reports/2013/reddit.aspx
[5]http://www.reddit.com/about/, as on June 7, 2014.

7.1 Normalized error

For real opinions, a natural measure of error is the squared error of the predicted opinion with respect to the observed opinion. Thus, error, $\boldsymbol{e}_k = |\mathbf{y}_k - \hat{\mathbf{y}}_k|$. Hence, the root mean square error (RMSE) for all nodes at time $k+1$ is:

$$E = \sqrt{\frac{\boldsymbol{e}_k^T \boldsymbol{e}_k}{N}}$$

However, this error metric is sensitive to the scale of the input data. Hence, we use the *normalized error* metric:

$$\text{NRMSE} = \frac{E}{(y_{max} - y_{min})} \tag{14}$$

where, $y_{max} = \max(x_k^i), \forall(i)_{i=1}^N \;\&\; \forall(k)_{k=1}^K$, and $y_{min} = \min(x_k^i), \forall(i)_{i=1}^N \;\&\; \forall(k)_{k=1}^K$, are the maximum and minimum values of all observed opinions, respectively.

7.2 Quantized error

Another metric which captures the polarity of the opinions is the quantized error. We define this as the fraction of times, the polarity of the predicted opinion matches the observed one. Thus, the quantized error at time instant $k+1$ is given by:

$$QError = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\left[y_{k+1}^i \hat{y}_{k+1}^i < 0\right] \tag{15}$$

where $\mathbf{1}(\cdot)$ is the indicator function. The product $y_{k+1}^i \hat{y}_{k+1}^i$ is positive only if y_{k+1}^i and $\hat{y}_{k+1}^i$ have the same sign.

7.3 Relative improvement factor

Apart from the above two metrics, we also use improvement factor (IF) metric as a performance indicator for opinion model with node classification. Formally, this is defined as,

$$\text{IF} = \frac{\text{NRMSE}_{\text{node-classification}} - \text{NRMSE}_{\text{individual-edge-weighting}}}{y_{max} - y_{min}}$$

7.4 Δ_{NRMSE} and Δ_{QError}: Metrics used in mesoscaling

To evaluate the utility of our mesoscaling model estimators, we first compute the errors (normalized RMSE and quantized error) for the community-level opinions and then report improvement of these metrics w.r.t. the same in the individual node levels. More formally, to obtain the error at time $k+1$ we compute the following:

$$\text{Comm-NRMSE} = \sqrt{\frac{1}{|C|} \sum_{c \in C} \frac{1}{|c|^2} \left[\sum_{i \in c} (y_{k+1}^i - \hat{y}_{k+1}^i)\right]^2}$$

$$\text{Comm-QError} = \frac{1}{|C|} \sum_{c \in C} \frac{1}{|c|} \sum_{i \in c} \left[\mathbf{1}(y_{k+1}^i \hat{y}_{k+1}^i < 0)\right]$$

Then we report,

$$\Delta_{\text{Metric}} = -\frac{1}{|C|} \sum_{c \in C} \frac{\text{Comm-Metric}[c] - \text{Node-Metric}[c]}{\text{Node-Metric}[c]}$$

Here, Metric is either NMRSE or QError, Node-Metrics are NRMSE or QError computed over the specified nodes, and Comm-Metrics are NRMSE or QError computed at the community level.

8 RESULTS

In this section, we establish that our proposed models are superior to competitive prior approaches. We report experimental results across all nine datasets — three datasets generated from in-house debate games and six datasets obtained by crawling Twitter and Reddit. For the first three datasets, the background process is modeled using the asynchronous full observation system, as all the opinions of every participant are captured in the dataset. For the other six datasets, we model the background process using periodic, aperiodic, and SBLM variants of our models. To better understand the performance of our models with respect to the existing state-of-the-art techniques, we consider four baseline opinion propagations models: Voter model [9], Biased Voter model [10], Flocking model [10, 24], and DeGroot's model [16]. To the best of our knowledge, this is the first work reporting a data driven comparison of opinion exchange models using real-world datasets.

8.1 Baselines

We compare our results with four popular state-of-the-art baseline models.
Voter Model [9]: In this strategy, at each step, first a node within the network is selected at random; next one of its neighbors is choosen uniformly at random (including itself) and then, the original adopts the chosen node's opinion as its own.
Biased Voter Model [10]: The Biased Voter Model introduces a bias over the Voter model, where the bias being that a user is most influenced by the neighbor whose opinion is closest to its opinion.
Flocking Model [24]: In the flocking model, a node i with opinion x_i first selects the set of neighbors j having opinions x_j so that $|x_i - x_j| < \epsilon$ and then updates its own opinion by averaging the opinions of the selected neighbors.
DeGroot's Model [16]: DeGroot's model assumes that a node within the network updates its opinion by taking a weighted average of its neighbors' opinion. In particular, this proposal assumes that the array of weights form a row-stochastic influence matrix with $w_{i,j} \geq 0$, and opinions in the range $[0, 1]$ (which stochastic updates preserve).

8.2 Performance comparison

For each approach, we learn the parameters that is best able to explain the data. Note that although we know how regularly we are sending crawling request to a search API, we do not know how regularly the tweets are missed in the collected data thus obtained. So effectively, it is not known apriori if a periodic or aperiodic strategy best complies with the collected data. Consequently, we consider all the models driven by different data collection mechanism (FLM/ALM/PLM) as well as influence types (Edge-weight based LM/SBLM) in depth.

- FLM (Full (observations) Linear Model) — Here we consider that all the opinions of each user is known, and the updates come asynchronously.
- PLM (Periodic Linear Model) — Here the opinions are always updated after every (say) t time steps. Note that, the collected opinion stream only contains the timstamps of the messages crawled. The actual number of missing updates t are not known. So we estimate t using cross-validation.
- ALM (Aperiodic Linear Model) — Here the length of time interval between subsequent opinion observations varies across time. Note that, with the variation of time intervals, ALM and PLM provide different performances. Since, the actual number of missing updates t_k are not known, we obtain them using cross validation.
- SBLM (Stochastic Block approach to Linear Models) — Here we consider the node labels to model the edge weights. The base edge weights are captured using both ALM and PLM.

However, we only report the results with base edge weights being picked up using ALM alone.

Tables 2 and 3 show a comparative analysis of the opinion prediction error (Normalized RMSE and Quantized error) of four baseline algorithms along with different variants of our algorithm. In particular, Table 2 shows the results for the datasets crawled from Twitter, while Table 3 describes the performance for the datasets gathered from the in-house games. The upper half of each table reports the normalized mean square error that is the actual opinion prediction error, while the rest report the quantized error that is the error in prediction of opinion polarity. We observe that across all these datasets, the overall performance of our schemes is substantially better than all the baselines.

8.2.1 Performance Analysis - Normalized RMSE (NRMSE).

The top half of each of Tables 2 and 3 shows a comparative view of actual opinion-prediction error.

Voter Model and Biased Voter Model: Performance of Voter model is particularly poor. It relies on random opinion updates, thus evidently loses information of actual heterogenous dynamics. Moreover, such version of Voter model keeps the set of opinions in a graph invariant throughout the process. This intrinsic property of Voter model prevents the opinion-values from not growing in a larger space which thereby goes against the spirit of continuous opinion-model. Biased Voter Model attempts to overcome these limitations by introducing node weights. However, the performance of Biased Voter model is worse than ALM or PLM. A closer scrutiny reveals that, biased voter model parameterizes the node weights; but, due to uniform edge weights, it is unable to capture the actual influence dynamics.

Flocking Model: Note that the NRMSE for flocking is substantially lower than Voter model (often Biased Voter model too) in most cases. Recall that this model updates the opinion of a node by averaging those of her neighbours, that are very close to her. Such a selective averaging strategy makes it functionally similar with the linear averaging models. As a result the performance of this model is better than Voter model and her variants.

DeGroot Model: The performance of DeGroot model is fairly competitive for Reddit and Twitter. This is mainly because it incorporates different edge weights that capture the actual dynamics of information-flow from one node to another, which is heavily neglected in the other three baselines. The relatively better performances of flocking and DeGroot model also reflect an inherent linearity in the dynamics that justifies our choice of a more generic linear model.

Linear Propagation Models: ALM and PLM perform significantly better than all the baselines in almost all the cases. A possible explanation can be that it captures the effect of intermittent observations i.e. the phenomenon of periodic/aperiodic observations, which none of the baseline algorithms take care of. Our model is also not limited to positive entries and row-stochasticity, which are the major features of DeGroot model. Being the most generic linear model it captures the negative influence, opinion fluctuation etc. and allows formation of any generic linear combinations of opinions rather than their convex combinations. This is evident from the few real-life examples in Figure 4, where panels (b) and (c) show that the opinion of a user (C) may not follow as a convex combination of the opinions of the users (B, A) she follows.

The performance of SBLM is significantly better than ALM and PLM. This is because, in most real scenarios, the edge-weights depend on the attributes of the connecting nodes. In fact, SBLM offers a perfect accuracy in Tw:Movie, Tw:Series, Tw:Boxing, and Tw:Bollywood. SBLM correctly captures that and enhances the performance. We give the details of the results for SBLM in Section 8.2.7.

Normalized RMSE (%)							
Dataset	SBLM	ALM	PLM	DeGroot	Voter	B-Voter	Flocking
Tw:Politics	2.01	9.59	9.83	10.20	22.98	17.49	9.49
Tw:Movie	2.08	5.71	6.54	7.33	24.29	11.32	16.31
Tw:Series	1.04	3.12	3.74	9.58	27.34	13.22	12.30
Tw:Boxing	1.06	3.77	5.14	7.50	16.26	8.28	16.21
Tw:Bollywood	3.00	2.64	2.58	8.78	28.89	21.20	20.20
Reddit	5.86	6.81	7.03	6.00	15.60	7.51	8.24
Quantized error (%)							
Tw:Politics	0.12	2.55	2.92	2.96	6.21	7.23	8.23
Tw:Movie	0.00	0.53	0.88	1.10	2.93	2.30	1.10
Tw:Series	0.00	2.11	2.74	3.10	4.20	3.20	2.90
Tw:Boxing	0.00	0.84	2.53	5.26	8.20	3.71	4.32
Tw:Bollywood	0.00	1.07	1.16	3.25	6.22	4.67	7.37
Reddit	1.02	0.00	1.07	1.68	2.70	2.16	3.20

Table 2. Opinion prediction performance for periodic and aperiodic observation scenarios for all crawled datasets for 90% training set. The first half of the table dissects forecasting error in terms of NMSE and the second half shows QError. In each cell. The cells with light orange (blue) color indicates the best (second best) predictor. The cells with grey color indicate the best performer among the four state-of-the-art baselines.

Normalized RMSE (%)					
Dataset	AsLM	BiasedVoter	Voter	DeGroot	Flocking
Continents	10.42	31.46	35.51	23.94	32.89
Colleges	12.80	22.77	28.69	59.28	32.06
Occupation	10.36	23.06	30.32	33.28	31.64
Quantized error (%)					
Continents	0	1.96	2.94	1.96	5.88
Colleges	0	2.94	3.92	2.94	4.90
Occupation	0	2.94	6.86	0.98	7.84

Table 3. Opinion prediction performance in case of *asynchronous full* observations for all the in-house games. The top half of the table shows prediction error in terms of Normalized RMSE and the bottom half gives quantized error. The cells with light orange (grey) color indicates the best (second best) predictor.

8.2.2 Performance Analysis - Quantized Error. As we can see from the bottom half of Table 2, quantized error is significantly lower in all the datasets, than the baselines. We also observe that SBLM substantially improves the performance as compared to the aperiodic and periodic counterparts. This is because, SBLM can accurately model the edge influences by incorporating possible node attributes. For the three social games (Table 3), the performances of all the algorithms are quite good. Our model gives a 100% accuracy in these three games. This is because the active participation of the users in the experiments lead to a rich dataset of opinions with a nice dynamical

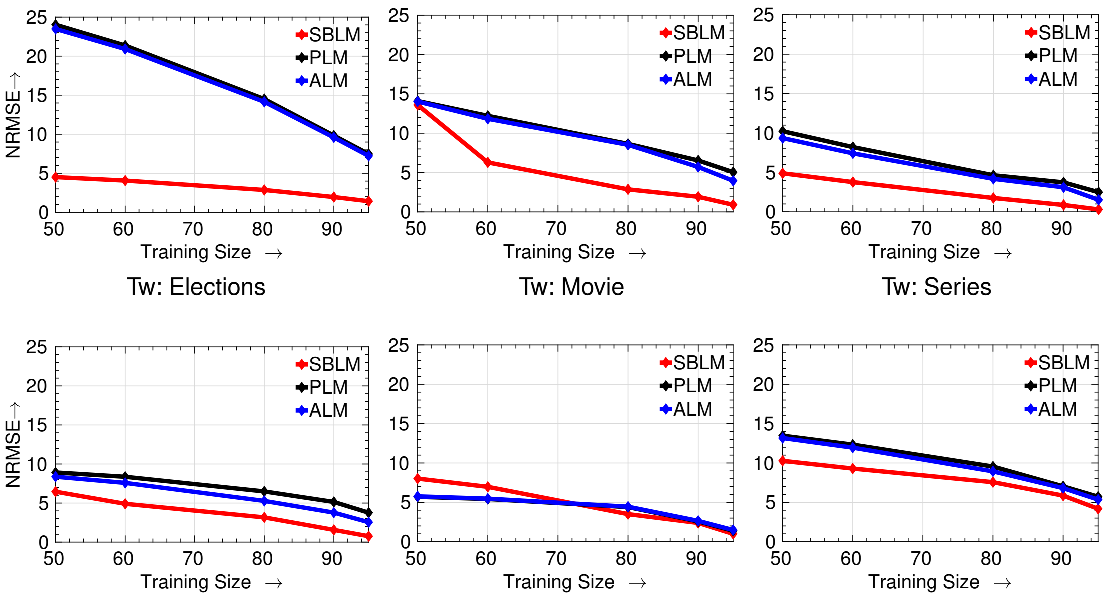
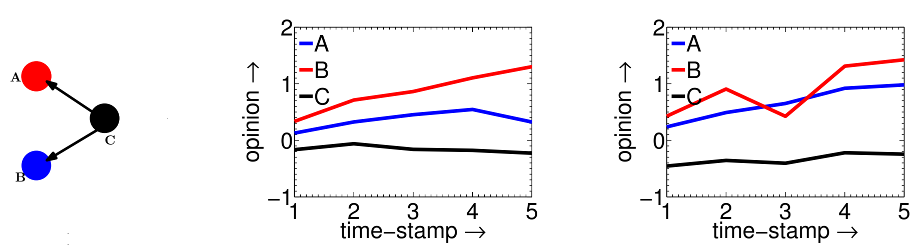


Fig. 3. Performance variation in terms of NMRSE with training size. As the training set increases, the performance of all the algorithms becomes better. SBLM is observed to be the most stable model amongst all. Due to a smaller number of parameters, SBLM can be trained with less training samples than what is necessary for training other paradigms.



(a) A subgraph (b) Tw:Politics (c) Tw:Series

Fig. 4. Two real life examples of opinion flow in a subgraph with three nodes. Panel (a) shows a subgraph structure with three nodes where C follows A and B. Panels (b, Tw:Politics) and (c, Tw:Series) show two examples taken from real data, that depict how opinions of A, B and C evolve with time. We observe that the opinion of C changes as a nonconvex combination of those of A and B. These examples motivate the necessity of a possible departure from DeGroot model which assumes the row stochasticity of the underlying weighted adjacency matrix.

flow without any intermittent observations. Hence, all the algorithms are able to capture the dynamics of the process with high prediction accuracy.

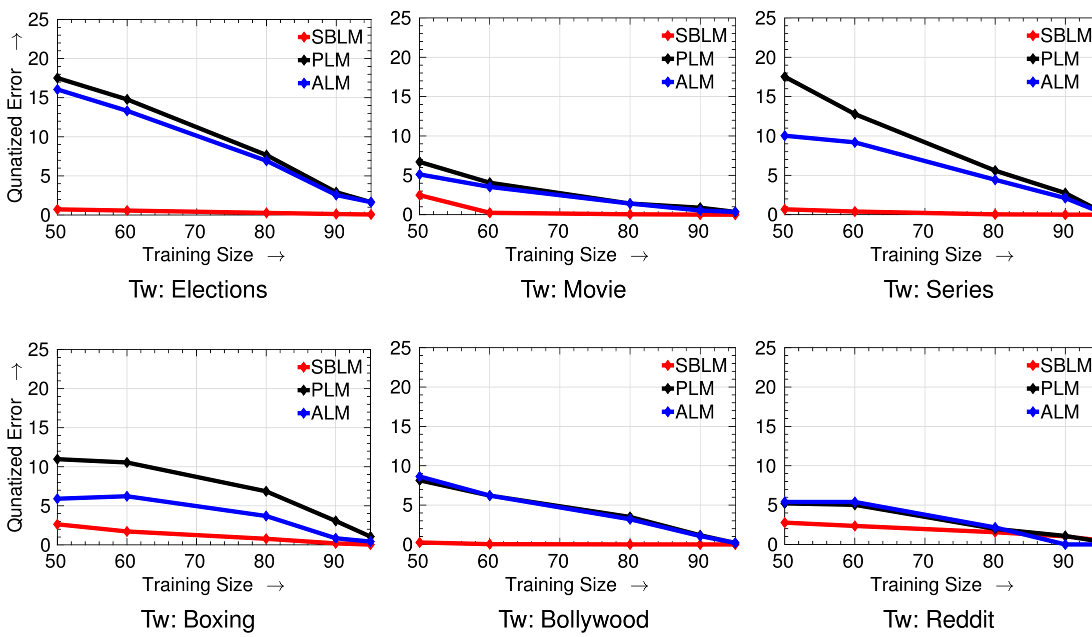


Fig. 5. Performance variation in terms of Quantized error with training size. As the training set increases, the performance of all the algorithms becomes better. We observe that all the algorithms show more or less stable performances with variation with training size. This is because, the variablity of polarity is far less than that of actual opinion. Consequently, all the algorithms can be trained with smaller number of samples and the performance stabilizes after a certain training-size.

8.2.3 Stability to training size. From Figures 3 and 5, we observe that as the training set size increases, the performance becomes increasingly better for both PLM and ALM. We also observe that SBLM is most frugal in terms of data requirements, and achieves the lowest errors for a given training set size. This is expected, as SBLM suitably fits the data, by properly learning the edge-weights along with the clusters the nodes belong to. For political topics (Tw:Politics), the improvement rate is very high, in other words, the performance becomes better w.r.t. training-size variation. It is because in political discussion, we observe a good dynamical flow in opinion diffusion, and consequently the predictive performance increases with higher training size. Furthermore, the distribution of opinion in messages drastically changes from before to after the election. As a result, the learned edge-influences are not so accurate while training on a smaller portion of data. As the sample-size increases, the training becomes more and more robust and the predictive performance improves. However, we find that for all other datasets, performance variation across training size is mostly stable.

8.2.4 Effect of mesoscaling . To understand the effect of mesoscaled data acquisition, we first construct communities over the social network using the method described in [19]. This algorithm allows to set the number of communities apriori, thus helping in further analysis. Also, communities obtained using this method contain crucial signals in social network scenarios [12, 13]. In the existing datasets, we averaged these opinions over communities to obtain the community-level sentiments. Then, we build our model MLearn over various training-set size and the number of

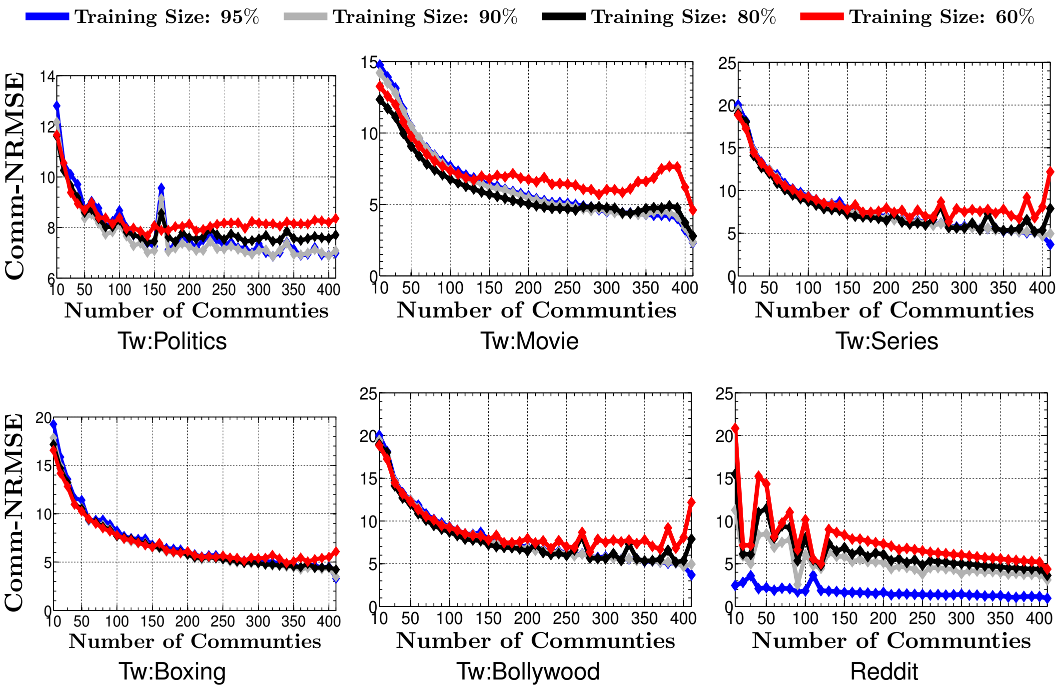


Fig. 6. Effect of mesoscaling. Actual opinion prediction performance (NMRSE) of community-level opinions. As the no. of community increases, the performance becomes better because of the increasing experssive power of the model that enables it to capture more and more granular signals.

communities. Figures 6 to 9 report the results for the effect of mesoscaling on the overall performance.

8.2.5 Overall Performance variation. As the number of communities increases, the overall performance in terms of Comm-NRMSE gets better (figures 6 and 7). This is expected, since as the number of communities increases, the model becomes more expressive and is able to capture the granular signals. Reddit show irregular pattern in its performance, since the communities are formed arbitrarily due to threshold based artificial graph construction. We also observe that when the number of specified communities is small, we don't observe much variation of performance with training size, since the community-level averaged opinions have similar distibution as time grows.

8.2.6 Performance improvement due to mesoscaling. In order to establish the utility of mesoscaling, we averaged individual NRMSE (derived from SBLM) of all members belonging to a community and obtained corresponding Comm-NRMSE and then compared it with Comm-NRMSE derived through mesoscaling. The value of Δ_{NRMSE} (Figure 8) is positive for most communities which firmly establishes the need of considering community level opinions which are smoother and hence less noisy. An interesting observation brought out by Figures 8 and 9 is that the performance improves initially as the number community grows and then decreases. This trend is clearly observed in the three datasets - Tw:Series, Tw:Boxing and Tw:Bollywood. It is indicative towards an optimal granularity of opinion sensing for the topic and the network. For large community sizes,

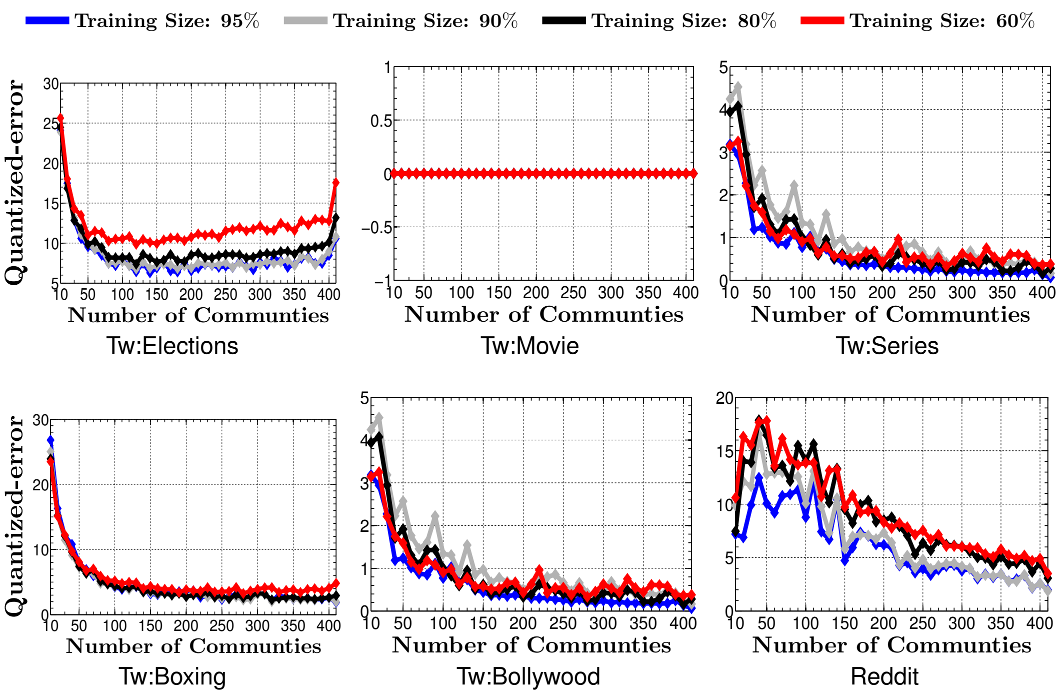


Fig. 7. Effect of mesoscaling. Opinion-polarity prediction performance (Quantized error) of community-level opinions. As the no. of community increases, the performance becomes better.

the community-level opinions are over-compressed, leading to under-fitting of the edge-weight matrix $\mathbf{A}$. On the other hand, extremely granular sensing inflicts more noise, leading to over fitting of $\mathbf{A}$.

The improvement of quantized error (second row of Figure 9) follows a similar trend, only it deteriorates faster specially in those scenarios where there are presence of lot of mild opinions and some dominating opinions. In general, it is a hard task to predict the polarity of users with mild opinions and that is reflected in the poor improvement factor for all the variants of our proposal.

8.2.7 Opinion model with node classification. In the last subsection, we found that the mesoscaling policy is useful in terms of the predictive power of the community-level signals. However, such community construction was based on edge-clustering. No node-properties was taken into account. Here, we investigate the impact of community-construction based on node-properties. In Table 4 we report the improvement factors for different edge-distribution and training sizes across all datasets collected from Twitter. We observe similar trend for Reddit.

In most cases, the overall performance of SBLM strategy is substantially better than its individual edge-weighting counterpart. This is because, in practice, the interaction between two users mostly depends on the nature/personalities of them. Therefore, the redundancy injected for exhaustive edge-set learning is reduced after incorporating the node properties into edge modeling. In other words, an individual edge-weighting method often leads to the overfitting of the model. By curating an edge influence as a function of node attributes, we reduce the no. parameters from $O(|E|)$ to $O(|V|)$, which in-turn decreases the overfitting tendency.

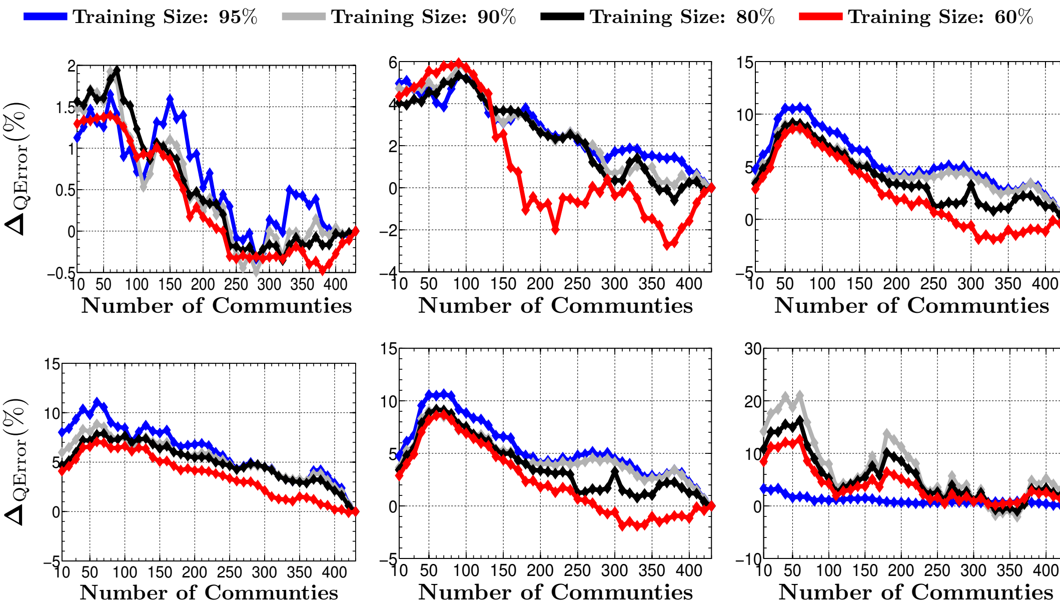


Fig. 8. Variation of performance improvement (in terms of RMSE) due to mesoscaling, with no. of communities and training-set size. As the number of community increases, the improvement first increases and then decreases, suggesting to an optimum level of compression that gives the best improvement.

	80% Sampling					**60% Sampling**				
					Normal					
Dataset	Politics	Movie	Series	Boxing	Bollywood	Politics	Movie	Series	Boxing	Bollywood
$C = 4$	6.0798	-15.1877	-15.9886	-14.4286	-46.3089	9.5455	-13.4338	-14.7631	-13.7680	-45.7740
$C = 7$	7.0634	3.5591	2.2852	2.8453	1.0575	10.5291	5.3129	3.5108	3.5060	1.5924
$C = 10$	7.0529	3.6077	2.2342	2.4405	0.7474	10.5186	5.3615	3.4598	3.1011	1.2822
					Exponential					
$C = 4$	6.0875	-16.4734	-14.8890	-14.5828	-46.1079	9.5532	-14.7196	-13.6634	-13.9221	-45.5730
$C = 7$	6.0989	-11.7864	-13.4794	-13.5742	-43.1158	9.5646	-10.0325	-12.2538	-12.9136	-42.5809
$C = 10$	6.0954	-10.1287	-12.7022	-13.0462	-37.1227	9.5611	-8.3748	-11.4767	-12.3855	-36.5878
					Pareto					
$C = 4$	5.8104	-40.5357	-32.0964	3.0884	-61.3375	9.2761	-38.7819	-30.8709	3.7490	-60.8027
$C = 7$	5.7672	-193.2024	-110.6837	2.4414	-168.6266	2.6163	-191.4466	-109.2810	3.1021	-168.0877
$C = 10$	5.7930	-248.9097	-136.9245	1.7654	-238.5074	1.1124	-247.1539	-135.5218	2.4260	-237.9685

Table 4. Improvement-factor (in %) after node-classifcation, given edge-weight distributions are Normal, Exponential and Pareto.

Table 4 dissects the variation of performance of SBLM w.r.t. the pre-specified number of clusters (C). We clearly observe that a normal distribution fares reasonably better than all other distributions in case of Politics, Movie, Series and Bollywood. Table 4 shows that, for $C = 7$, normal distribution provides a significant performance-boost for these four datasets. E.g. in Tw:Politics, the improvement factor is more than 7%. However, in case of Tw:Boxing dataset, Pareto performs best, resulting in a 3% performance improvement. For Tw:Politics, the performance is most consistent. It performs well even with a small value of prespecified no. of communities, which shows the utility

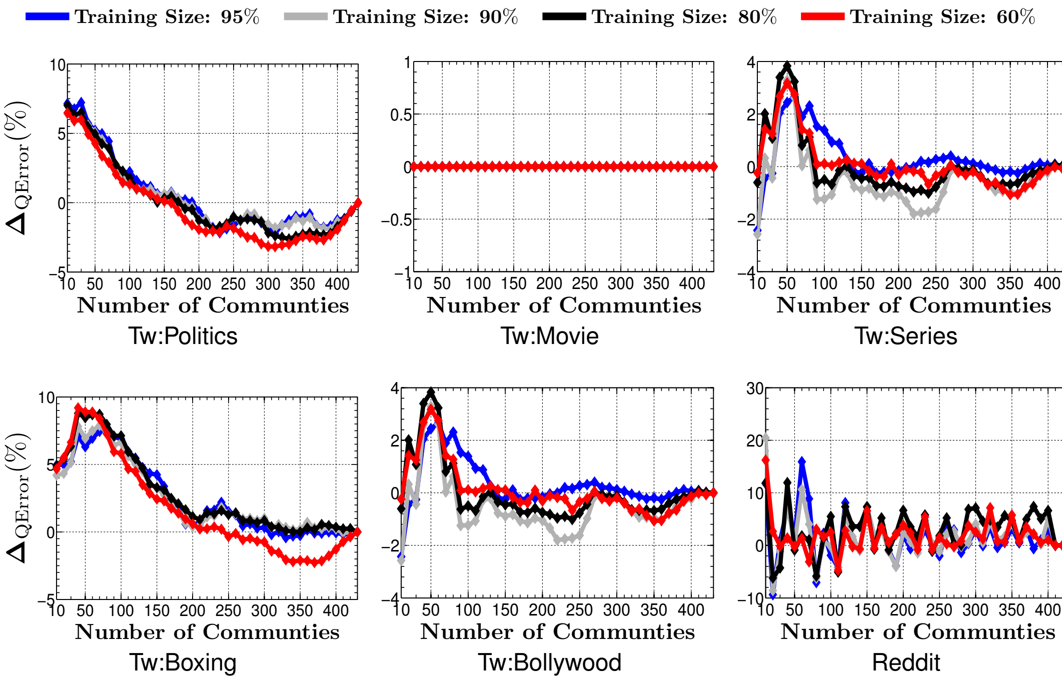


Fig. 9. Variation of performance improvement (in terms of qunatization error) due to mesoscaling, with no. of communities and training-set size. As the number of community increases, the improvement first increases and then decreases, suggesting to an optimum level of compression that gives the best improvement.

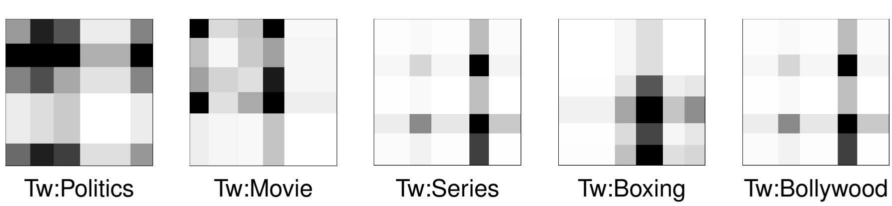


Fig. 10. Community structures on Twitter datasets, due to co-clustering algorithms on opinion models with node-types. For Tw:Politics and Tw:Movie, we observe the community structures have higher entropy than other datasets, that is an indication of strong interactions between the node-clusters.

of our frugal modeling assumption in general.

The overall better performance of a normal distribution is primarily attributed to the existence of substantial negative influences between users. This is a major advantage of the proposed model over DeGroot model, the closest counterpart of its kind. Hence, a normal distribution reflects the proper spectrum of the edges as opposed to exponential and Pareto distributions which restrict the edge weights to be positive. Although Pareto fares quite well in case of Tw:Boxing dataset, it fares very poorly for all other cases.

Quite surprisingly, Table 4 reveals that the relative performance of SBLM is significantly better for training with 60% samples than 80%. Note that, SBLM requires a far less amount of parameters to be trained, as compared to the per-edge models like ALM and PLM. Therefore, as the training size decreases, it shows a slower rate of performance degradation. As a result, the improvement factor becomes high for smaller training size.

It can be observed that, in most of the datasets (Tw:Politics, Tw:Movie, Tw:Boxing, and Tw:Bollywood), on increasing the number of node classes, the performance first improves and then deteriorates[6]. It is because, choice of a few node-clusters may lead to too much compression of important user-aspects which is reflected in a relatively poor performance with $C = 4$ for all apriori edge-distribution. On the other hand, increasing the number of clusters appears to overfit the learned-edge weight that also results in a poor predictive power.

Figure 10 depicts the co-clustering structure obtained on learning SBLM, for normal distribution with $C = 7$. In case of Tw:Politics and Tw:Movie, we observe that the entropy of the revealed community structure is higher than other datasets. It indicates an intense inter-cluster interaction (high B matrix). To some extent, we believe that this is one of the likely reasons responsible for high improvement factor ($> 7\%$ for $C = 7$, Normal edge distribution) in case of Tw:Politics dataset. Moreover, cross interaction between classes often indicates towards overlapping clusters i.e. mixed-membership of a user to various communities. This leaves an open space for modeling mixed-membership stochastic block model while learning edge-influences in context of opinion dynamics. On the other hand, for datasets (Tw:Series, Tw:Boxing and Tw:Bollywood) with block-structures with lower entropy, the corresponding IF values turn out to be low.

8.2.8 Variation across data sets. From Table 2 we observe that the algorithms perform substantially better in Reddit than in all Twitter datasets. Note that in case of Reddit we have collected the evolution of general political opinion whereas in Twitter we concentrated on specific events. Reddit is a forum, where people actually join to form an opinion/impression. Therefore, it is natural that a user in Reddit view others' post, form an opinion and write a well-thought post. Also since the users are more in exploratory mode, a Reddit user can read and scrutinize any other people's comments, which evidently helps her to form an opinion. In our model we have taken a decent estimate whereby two agents are neighbors if they have subscribed to three common subreddits, even then we find that the reach of each agent is a magnitude higher than that of Twitter.

On the other hand Twitter is a popular social-network site and we are looking into the data of particular popular events. Since the underlying graph structure is sparse, an opinion may take time to propagate and may get lost in the process [32]. Thus, the effect of a distant node becomes almost negligible. Also since the event tracked is popular, much of the information may be coming from (outside) Twitter and a user's opinion may get influenced due to that [35]. Therefore, PLM/ALM which assume local influence perform worse in capturing the influence dynamics.

9 CONCLUSION

In this paper, we presented a family of models for opinion propagation in social networks, by estimating edge strengths from the stream of quantitative opinions at the nodes, changing with time. Our model is based on a simple idea that opinion of a user changes as a linear combination of the opinions of her neighbors. Here, the polarities and weights of this linear function reflect the corresponding influence between users. Unlike earlier work, our approach does not favor any particular asymptotic or steady state behavior; rather, it aims to capture fine grained transient opinion dynamics. In order to make our model practically effective, we also consider a wide variety of scenarios involving intermittent observation of opinions at irregular intervals. We also consider

[6]This optimal number of node-classes are near seven (six to eight) for various datasets.

another realistic data collection scenario, where opinions are acquired over communities, rather than polling individual nodes. Such a setting provides a better predictive performance in case of community-level opinion prediction. Finally, in order to offer interpretable and frugal models, we present a variant where edge influence mainly depends on the corresponding node properties, which in turn are learned using a stochastic block model. Such a regularized form of influence reduces overfitting and further boosts the predictive power of our model. Extensive experiments over nine real-world datasets show that our proposal significantly outperforms four state-of-the-art baselines in predicting opinion dynamics of users individually (per user level) as well as collectively (per community level).

Our work opens up many interesting directions for future work. An immediate extension would be to aim for a nonlinear opinion modeling, which should uncover the complexity of the dynamics better than its linear counterpart. Further, it would be of interest to remove our assumption that influence itself is stationary. Specifically, here we assumed that $\boldsymbol{A}$ is constant across time. However, in the context of transient dynamics $\boldsymbol{A}$ itself can vary across time. Therefore, a time varying adjacency matrix could capture opinion signals at more granular level. The structure of $\boldsymbol{A}$ often reveals various properties of the users. Although one way to capture these node properties is addressed in this paper, other approaches, e.g., Chung and Lu's model [2], latent product model, etc., may be useful.

REFERENCES

[1] C. Aicher, A. Z. Jacobs, and A. Clauset. Learning latent block structure in weighted networks. *Journal of Complex Networks*, 2014.

[2] W. Aiello, F. Chung, and L. Lu. A random graph model for power law graphs. *Experimental Mathematics*, 10(1):53–66, 2001.

[3] A. Anagnostopoulos, G. Brova, and E. Terzi. Peer and authority pressure in information-propagation models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 76–91. Springer, 2011.

[4] R. J. Aumann. Agreeing to disagree. *The annals of statistics*, pages 1236–1239, 1976.

[5] R. Axelrod. The dissemination of culture: A model with local convergence and global polarization. *Journal of Conflict Resolution*, 41(2):203–226, 1997.

[6] D. Bindel, J. Kleinberg, and S. Oren. How bad is forming your own opinion? In *FOCS*, 2011.

[7] B. Chazelle. Natural algorithms and influence systems. *Commun. ACM*, 55(12):101–110, Dec. 2012.

[8] F. Chierichetti, J. Kleinberg, and S. Oren. On discrete preferences and coordination. In *EC*, 2013.

[9] P. Clifford and A. Sudbury. A model for spatial conflict. *Biometrika*, 60(3):pp. 581–588, 1973.

[10] A. Das, S. Gollapudi, and K. Munagala. Modeling opinion dynamics in social networks. In *WSDM*, 2014.

[11] A. De, S. Bhattacharya, P. Bhattacharya, N. Ganguly, and S. Chakrabarti. Learning a linear influence model from transient opinion dynamics. In *CIKM*, 2014.

[12] A. De, S. Bhattacharya, S. Sarkar, N. Ganguly, and S. Chakrabarti. Discriminative link prediction using local, community and global signals. In *IEEE TKDE*, 2016.

[13] A. De, N. Ganguly, and S. Chakrabarti. Discriminative link prediction using local links, node features and community structure. In *ICDM*, 2013.

[14] A. De, I. Valera, N. Ganguly, S. Bhattacharya, and M. Gomez Rodriguez. Learning and forecasting opinion dynamics in social networks. In *NIPS*, 2016.

[15] G. Deffuant, D. Neau, F. Amblard, and G. Weisbuch. Mixing beliefs among interacting agents. *Advances in Complex Systems*, 3(01n04):87–98, 2000.

[16] M. H. DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345), 1974.

[17] P. M. DeMarzo, D. Vayanos, and J. Zwiebel. Persuasion bias, social influence, and unidimensional opinions. *The Quarterly Journal of Economics*, 118(3):909–968, 2003.

[18] D. Derks, A. E. R. Bos, and J. V. Grumbkow. Emoticons and Social Interaction on the Internet: the Importance of Social Context. *Computers in Human Behavior*, 23(1):842–849, 2007.

[19] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *KDD*, 2003.

[20] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232, 2015.

[21] A. Goyal, F. Bonchi, and L. V. Lakshmanan. Learning influence probabilities in social networks. In *WSDM*, 2010.
[22] W. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017.
[23] A. Hannak, E. Anderson, L. F. Barrett, S. Lehmann, A. Mislove, and M. Riedewald. Tweetin' in the Rain: Exploring societal-scale effects of weather on mood. In *ICWSM*, 2012.
[24] R. Hegselmann and U. Krause. Opinion dynamics and bounded confidence: Models, analysis and simul ation. *Journal of Artificial Societies and Social Simulation*, 5:1–24, 2002.
[25] M. Henaff, J. Bruna, and Y. LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015.
[26] P. Holme and M. E. Newman. Nonequilibrium phase transition in the coevolution of networks and opinions. *Physical Review E*, 74(5):056108, 2006.
[27] B. Iannazzo. On the newton method for the matrix pth root. *SIAM Journal on Matrix Analysis and Applications*, 28(2):503–523, 2006.
[28] D. Jensen, J. Neville, and B. Gallagher. Why collective inference improves relational classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 593–598. ACM, 2004.
[29] D. Kempe, J. Kleinberg, S. Oren, and A. Slivkins. Selection and influence in cultural dynamics. In *EC*, 2013.
[30] D. Kempe, J. Kleinberg, and E. Tardos. Influential nodes in a diffusion model for social networks. In *Proceedings of the 32Nd International Conference on Automata, Languages and Programming*, ICALP'05, pages 1127–1138, 2005.
[31] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
[32] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a Social Network or a News Media? In *WWW*, 2010.
[33] N. Lanchier. Opinion dynamics with confidence threshold: an alternative to the Axelrod model. 2010.
[34] J. Lorenz. Heterogeneous bounds of confidence: Meet, discuss and find consensus! *Complexity*, 15(4):43–52, 2010.
[35] S. A. Myers, C. Zhu, and J. Leskovec. Information diffusion and external influence in networks. In *KDD*, 2012.
[36] J. Neville and D. Jensen. Relational dependency networks. *Journal of Machine Learning Research*, 8(Mar):653–692, 2007.
[37] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135, Jan. 2008.
[38] J. W. Pennebaker, M. E. Francis, and R. J. Booth. LIWC: Linguistic Inquiry and Word Count. liwc.net, 2007. Accessed on June 03, 2014.
[39] M. Piskorec, N. Antulov-Fantulin, I. Miholic, T. Smuc, and M. Sikic. Modeling peer and external influence in online social networks. *arXiv preprint arXiv:1610.08262*, 2016.
[40] M. Richardson and P. Domingos. Mining the network value of customers,. In *KDD*, 2001.
[41] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer, 2018.
[42] S. Shahrampour, S. Rakhlin, and A. Jadbabaie. Online learning of dynamic parameters in social networks. In *NIPS*. 2013.
[43] J. F. Trevor Hastie, Robert Tibshirani. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001.
[44] B. Wang, M. Liakata, A. Zubiaga, and R. Procter. Tdparse: Multi-target-specific sentiment recognition on twitter. In *EACL*, volume 1, pages 483–493, 2017.
[45] G. Weisbuch, G. Deffuant, F. Amblard, and J.-P. Nadal. Meet, discuss, and segregate! *Complexity*, 7(3):55–63, 2002.
[46] E. Yildiz, A. Ozdaglar, D. Acemoglu, A. Saberi, and A. Scaglione. Binary opinion dynamics with stubborn agents. *ACM Trans. Econ. Comput.*, 1(4), 2013.
[47] M. E. Yildiz, R. Pagliari, A. Ozdaglar, and A. Scaglione. Voting Models in Random Networks.