Quantitative Study of Individual Emotional States in Social Networks

Jie Tang, Yuan Zhang, Jimeng Sun, Jinghai Rao, Wenjing Yu, Yiran Chen, and ACM Fong

Abstract—Marketing strategies without emotion will not work. Emotion stimulates the mind 3000 times quicker than rational thought. Such emotion invokes either a positive or a negative response and physical expressions. Understanding the underlying dynamics of users' emotions can efficiently help companies formulate marketing strategies and support after-sale services. While prior work has focused mainly on qualitative aspects, in this paper, we present our research on quantitative analysis of how an individual's emotional state can be inferred from her historic emotion log and how this person's emotional state influences (or is influenced by) her friends in the social network. We statistically study the dynamics of individual's emotions and discover several interesting as well as important patterns. Based on this discovery, we propose an approach referred to as MoodCast to learn to infer individuals' emotional states. In both mobile-based social network and online virtual network we verify the effectiveness of our proposed approach.

Index Terms—Social network, Predictive model, Emotion dynamics, Social influence

1 Introduction

Emotion can be defined as a complex series of psychophysical stimulate that arises spontaneously 3000 times faster than rational thought. Such emotion invokes either a positive or a negative response and physical expressions. Years before, researchers believed that our decision process was mainly taken by rational thoughts. The truth, however, turned out to be much different. Rational thought leads customers to be interested but it is emotion that sells. Marketers now realize that human beings are quite emotional in nature and they take their purchase decision emotionally.

It is an emotional world we live in. We may become happy because of watching a great movie, having delicious food, or having completed a difficult task; while we may also feel happy just because our friends feel happy. It is interesting to understand how an individual's emotion is influenced by various factors. Previously, it was very difficult to study the problem due to the lack of availability of data. Recently, with the success of many large-scale online social networks, such as Facebook, MySpace, Ning, and Twitter, many virtual communities are loosely formed and are often based around some common interests such as forums on a wide variety of issues ranging from product reviews to presidential campaigns. In all of these virtual communities, the change of an individual's emotional state can influence others in positive or negative ways and the spread of emotion may further trigger large cascade adoptions of happiness or depression. This propagation of emotion changes has a profound effect on the collective sentiments in social networks. Understanding the dynamics of human emotions in online social networks can provide rich information for applications such as spread of political views and consumer buying patterns. Indeed, there have been qualitative studies on the spread of happiness or sadness. For example, it was found that an individual could propagate one's emotional state through links up to three degrees of separation [1]. It was also found that happy people tend to link up with happy people, and likewise for unhappy people.

In this work, we aim to systematically and quantitatively study how individuals' emotional states evolve in social networks. Given complex social dynamics, we focus on three aspects: (1) *attributes correlation*: how a user's activities and other environmental attributes affect her emotional state; (2) *temporal correlation*: how a user's emotional state at current time is correlated to her emotional state in the recent past; (3) *social correlation*: how a user's emotional state is correlated with emotional states of her social friends.

Employing a mobile social network as the basic for our experiments, we statistically analyze the correlations of emotional states from the three aspects mentioned above. We observed several interesting patterns (factors) that influence the dynamics of users' emotional states. By leveraging these factors, we formulate the problem of emotion prediction in a dynamic continuous factor graph model. Specifically, in this model, the discovered patterns are modeled as different types of factor functions and the user's emotional changes over time are modeled as a Markov chain. Our model is generalizable to other problems by defining different factor functions. Solving

- *J. Tang, Y. Zhang, W. Yu, and Y. Chen are with the Department of Computer Science and Technology, Tsinghua University, Beijing, 100084. China.*
  E-mail: jietang@tsinghua.edu.cn, {fancyzy0526,angilyu,mithich}@gmail.com
- *J. Sun is with IBM TJ Watson Research Center, USA*
  E-mail: jimeng@us.ibm.com.
- *J. Rao is with Nokia Research Center Beijing, China.*
  E-mail: jinghai.rao@nokia.com.
- *ACM Fong is with the School of Computing & Mathematical Sciences, Auckland University of Tech., New Zealand.*
  E-mail: afong@aut.ac.nz.

Manuscript received January 15, 2011; revised January 15, 2011.

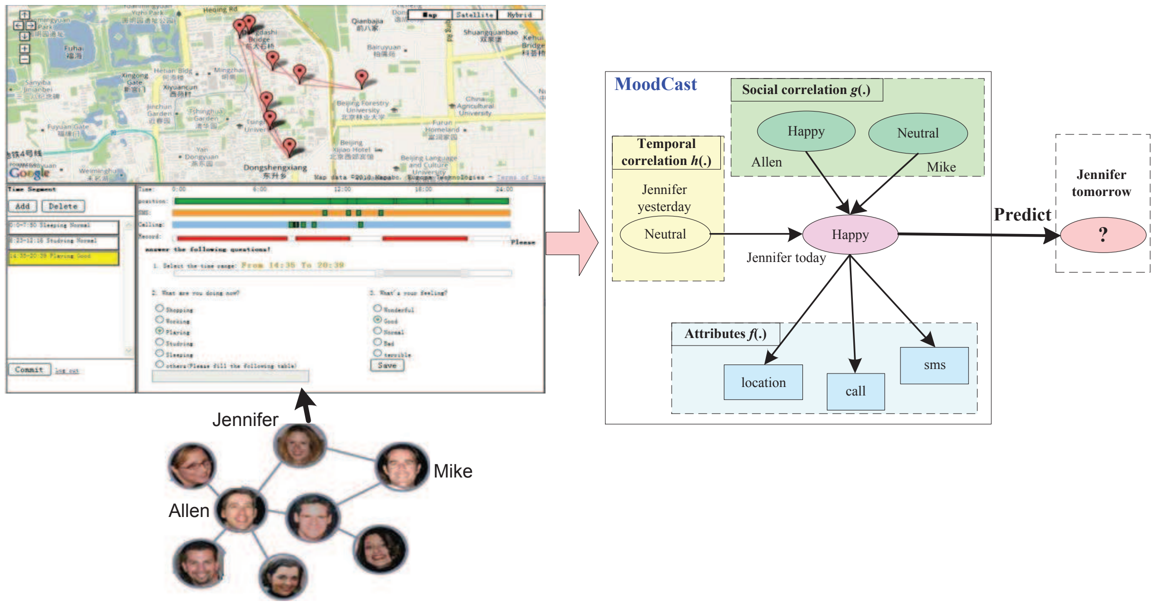


Fig. 1. Problem illustration of emotion prediction in a mobile social network.

such a factor graph model is often intractable. We design an approximate method MoodCast using Metropolis-Hastings sampling to find the maximum likelihood estimation based on the data. We apply MoodCast to predict user's emotion for two real social networks: one mobile social network and one social network derived from LiveJournal.com. Experimental results on both data sets show that MoodCast can clearly improve the prediction accuracy against several baseline methods.

Figure 1 illustrates the process involved in the MoodCast model learning and prediction on the mobile network data set. The bottom-left figure shows Jennifer's social network and the top-left figure shows the personal information for Jennifer. Actually, this is the GUI of our system to collect the mobile social context data. In particular, we collected the mobile context data (e.g., GPS location, call logs, SMS text) of each user, and asked the users to annotate their actions (behaviors) and emotional states when they change. The users can annotate their actions and emotional states in real time through a tool installed on the mobiles or make the annotation through the system afterward. See Section 6 for details of the data set. The middle chart in Figure 1 illustrates the learning model, which incorporates three different types of information including user's historic emotional states, friends' emotional states, and user attributes. The output of the learning phase is a predictive model, a dynamic continuous factor graph model in our case. The final step is to predict the emotional states of each user in the near future.

The rest of the paper is organized as follows: Section 2 formally formulates the problem; Section 3 gives a series of analysis based on the mobile social network and present our observations. Section 4 explains the proposed model. Section 5 presents the algorithm for learning the model. Section 6 gives experimental results that validate the effectiveness and the computational efficiency of our methodology. Finally, Section 7 discusses related work and Section 8 concludes.

2 Problem Definition

In this section, we first give several necessary definitions and then present a formal definition of the problem.

A static social network can be represented as $G = (V, E)$, where V is the set of $|V| = N$ users and $E \subset V \times V$ is the set of directed/undirected links between users. Given this, we can define the user's emotional state as follows.

Definition 1. ***Emotion***: *A user v_i's emotional state at time t is represented as y_i^t. Let $\mathcal{Y}$ be the space of the emotional state. We can denote the historic log of all users' emotional states up to time t as $Y = \{y_i^t\}_{i,t}$, where $y_i^t \in \mathcal{Y}$.*

In general, one's emotions is defined with a set of discrete states. For example, in the mobile social network, five different states are defined: "wonderful", "good", "neutral", "bad", "terrible"; while in the LiveJournal social network, the emotional states are defined as a list of moods are defined such as "happy", "sad", "angry", and "tired" [1].

Further, each user is associated with a number of attributes (or actions) at a continuous time-scale thus we have the following definition.

Definition 2. ***Continuous time-evolving attributes***: *The continuous time-evolving attributes in the social network are defined as a set of historic attribute-value logs, i.e., X. Specifically, suppose each user has d attributes. When user v_i changes the value of her j-th attribute at time t, we add a three-dimensional element $(v_i, t, a_j) \rightarrow x_{ij}^t$ into the set X,*

1. http://www.livejournal.com/moodlist.bml

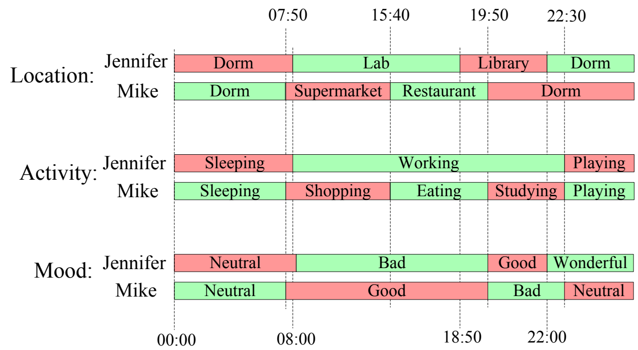


Fig. 2. Example of dynamic attributes.

where x_{ij}^t is the new value of the attribute a_j associated with user v_i at time t. The attribute value maintains until the next change.

Note that as the attribute of a user (e.g., GPS location) may change at any time, it is necessary to define the attribute changes on a continuous time-scale, rather than discrete timestamps. Figure 2 illustrates some examples of continuous time-evolving attributes of two users. The beginning of each color bar indicates the value change of a specific attribute.

Given this, we can define the input of our problem, a dynamic continuous network.

Definition 3. $t-$***Dynamic continuous network****: The dynamic continuous network (from time 0 to time t) is denoted as $G^t = (V, E^t, X^t, Y^t)$, where V is the set of users, $e_{ij}^t \in E^t$ is a link created at time t between user v_i and user v_j, and X^t is the continuous time-evolving attributes of all users in the network, and Y^t represents the set of emotional state changes of all users in the social network.*

We use the superscript t to denote that the dynamic continuous information in the network G^t is up to time t, that is, all edges E, attribute changes X and emotional state changes Y are recorded until time t. Thus the learning task of our problem can be defined as:

Learning task: Given a dynamic continuous network G^t, the goal is to learn a predictive function f to infer emotional states of users at a future time $(t+1)$. Formally, we have

$$f(X^{(t+1)}, V, E^{(t+1)} | G^t) \rightarrow Y^{(t+1)}$$

To learn the emotion predictive model, there are several requirements. First, as the input is a dynamic continuous network, it is necessary to find a function to capture the continuous property. Second, the emotional state of users is related to multiple factors, e.g., network structure, attribute changes, and users' historic emotional state, it is important to find a unified model which is able to incorporate all the information together. Third, the algorithm to learn the predictive model should be efficient. In practice, the scale of the social network might be very large.

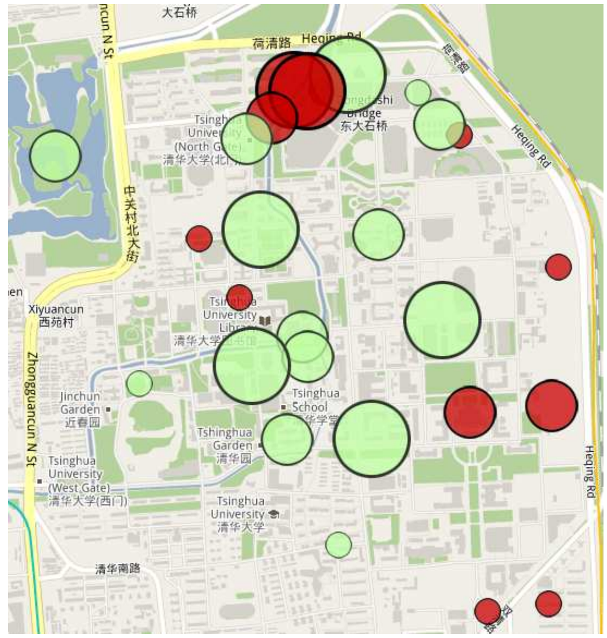


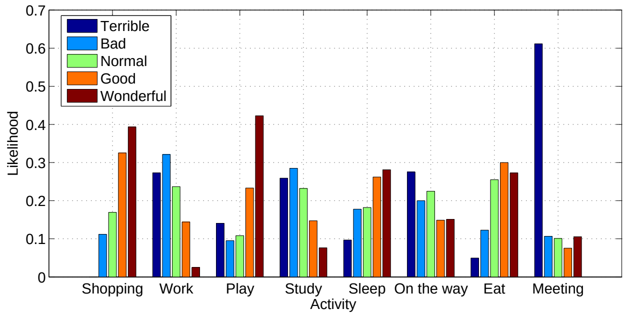
Fig. 4. Location correlation. The user's locations correlate with her current mood. Red circles represent happy places, while blue circles represent unhappy places. The size of the circle indicates the number of the users who feel happy in the corresponding place.

In existing works, Tang et al. [2] proposed a graph model to predict user actions (behaviors) in the dynamic social network and Goyal et al. [3] investigated how to learn the influence probabilities from the history of users' actions. Both models consider user actions, but do not consider users' emotions. Backstrom et al. [4] proposed employing the network structure to improve the performance of classifying users' behaviors in online groups. Christakis and Fowler [5], [1] studied the problem of the spread of happiness in social networks. However, they only qualitatively test the spread of happiness on two small data sets. To the best of our knowledge, no previous work has been done for emotion prediction in the context of dynamic social networks.

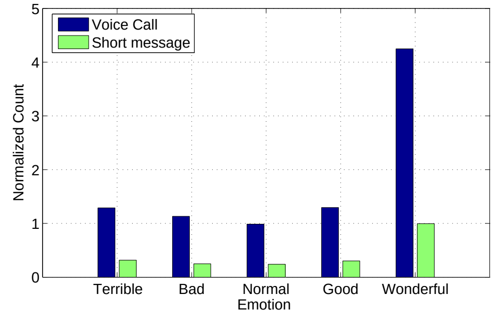
Analyzing users' emotional states can benefit many applications in consumer electronics. For example, an analysis on Twitter shows that the rise and fall of stock market prices is strongly influenced by the public mood [6]. It is also shown that emotional state is almost a crucial factor to influence users' purchase decision in car selling [7].

3 OBSERVATIONS

Before describing our approach for the emotion prediction problem, we first conduct a series of analyses on the mobile social network and present several interesting patterns we have discovered. The data set used here represents 36,000 hours of continuous behaviors and emotional states logged by the mobile phones of 30 users

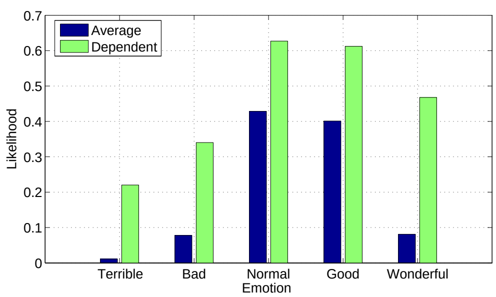


(a) Activity correlation

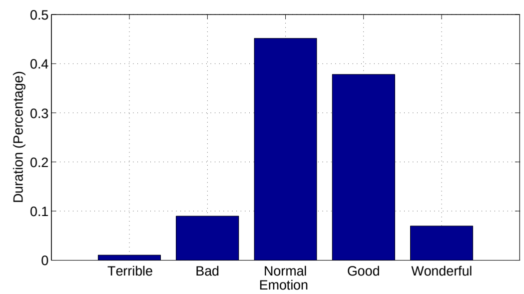


(b) Calling (SMS) correlation

Fig. 3. Attribute correlation. (a) Users' activities correlate with her current mood. y-axis represents the likelihood of a user staying at an emotional state when she is involved into the corresponding activity. (b) Users' calling behaviors correlate with her emotional states.



(a) Temporal correlation



(b) Time duration

Fig. 6. (a) The user's previous emotion influence on his current mood. *Average* is the likelihood of a user with emotion y and *Dependent* is the likelihood given that the user feels the same at previous time. (b) Average time duration for different emotional states.

Fig. 5. Case study: emotion-based friend groups for user "Michiel". The red circle includes friends with a higher $happy\_ratio$ and the blue one includes friends with a lower $happy\_ratio$.

from May to July, 2010. (Cf. Section 6 for more details of the data set.)

In the analysis, we focus on the following aspects.

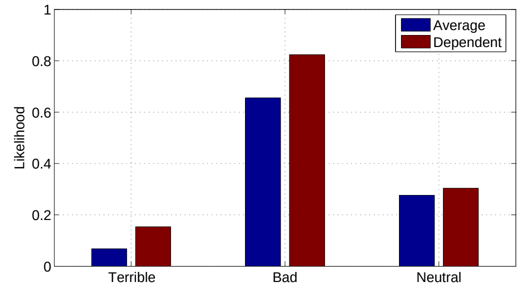
- *Attributes correlation:* probability that a user's emotional state is influenced by different environmental attributes and activities of that user.
- *Temporal correlation:* probability that a user's current emotion and her previous emotion have the same state.
- *Social correlation:* Probability that a user and her friend have the same emotional state.

3.1 Observation on Attributes Correlation

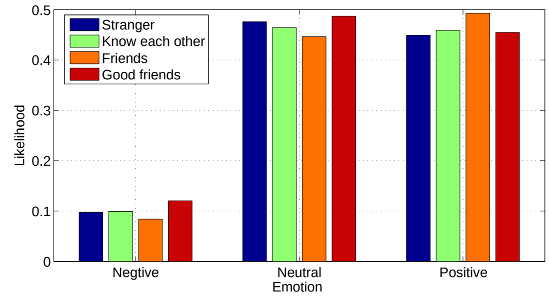
A user's environmental attributes may determine her emotional state at different time. For example, in the mobile social network, one's emotional state can be reflected by her activity, location, calling log, SMS text, etc. We analyze the correlation between each attribute and the user's emotional state.

Figure 3(a) shows the correlation between users' activities and their emotional states. It can be seen that when playing and shopping, a person's emotional state more likely stays at "Positive" (Good and Wonderful). While when meeting, the person's emotional state more likely stays at "Negative" (Bad and Terrible). We also find that people's emotional state tends to be negative when they are on the way to some places. This is possibly due to the traffic jam of Beijing. Figure 4 shows the correlation of user's location with her emotional state. We see that users' emotional states implicitly form happy places (plotted by red circles), where most users' emotion stay at "Positive" and unhappy places (plotted with blue circles), where most users' emotion stay at "Negative".

Furthermore, we have another two interesting observations from the users' calling records and SMS text.



(a) Social correlation



(b) Social correlation

Fig. 7. (a) Friends' emotion influence on one's current mood. *Average* is the likelihood of a user with emotion y and *Dependent* is the likelihood given that the user's friends feel the same. (b) Emotion influence of different types of friends on one's current emotional state. There are four types of social friends (with different social strength): Good Friends, Friends, Know each other, and Stranger.

Figure 3(b) shows the correlation between the user's calling records and her current mood. It seems that 64% of the users are inclined to share their happiness, while 36% of the users tend to talk more with their friends when they feel unhappy. On average, the former group makes 6.3 calls when they are happy, while the latter makes only 5.0 calls. Meanwhile, when they are unhappy, the averages call made by two groups are 0.3 and 3.1.

Another observation is that a user's friends also form implicit groups (e.g., "happy friends" and "unhappy friends"). To demonstrate this, we selected a few active users (who made the largest numbers of calls in our data set) and conducted a statistical analysis on all contacts whom the users called in the two months. To avoid bias, we removed those contacts who communicated with the current user in the two months less than 10 times. Then, for each pair of the selected active user and a contact she called, we calculate a score of $happy\_ratio$:

$$happy\_ratio_{uc} = \frac{\#call\_c\_when\_happy}{\#call\_out\_to\_c\_when\_unhappy} \quad (1)$$

where $\#call\_c\_when\_happy$ is the number of calls user u made to user c when her state stays at "Positive" and $\#call\_c\_when\_unhappy$ is the number of calls user u made to user c when her state stays at "Negative". If the $happy\_ratio_{uc}$ between user u and c is greater than 1.0, we say user c belongs to the happy friends group (positive emotion) of user u; otherwise she belongs to the group (negative emotion) of unhappy friends. A striking finding is that users' friends can be clearly classified into happy group and unhappy group and the overlap between the two groups is very small. A typical example is presented in Figure 5.

3.2 Observation on Temporal Correlation

A user's emotional state at current time is highly correlated to her emotion in the recent past. Figure 6(a) confirms this temporal correlation of the emotional state on the mobile social network (Mobile) data set. It can be easily seen that there is a strong dependency between one's current emotional state and her emotional state in the recent past. For example, when the user feels happy (good or wonderful), she has a much higher probability to continue to be happy than the others. Figure 6(b) shows the average time duration for each emotional state. In general, positive emotions (Good and Wonderful) has a longer duration than negative emotions (Bad and Terrible).

3.3 Observation on Social Correlation

A user's emotional state may be also influenced by her friends. A user is more likely to be happy if her friends feel happy as well. The influence of social emotion has also been previously studied. For example, Christakis and Fowler [5], [1] have studied a special case of emotion influence. They found that within a social network, happiness spreads among people up to three degrees of separation, which means when you feel happy, everyone within three degrees to you has a higher likelihood to feel happy too. Following this thread, we study whether the emotional state also has such an influence pattern in the mobile social network.

Figure 7(a) shows how a user's emotional state is influenced by her friends' emotions. We see that on average the user has a higher likelihood to feel the same with her friends. We also see an interesting phenomenon that negative emotions (bad and terrible) seem to be more infectious in the network. When a user feels unhappy (negative), the likelihood of her friends that also feel unhappy almost doubles than happiness (18% vs 9%). We further made an analysis for social correlation based on different types of social ties. In particular, we asked all the mobile users to manually label each of their social links with "Good friends", "Friends", "Know each other only". Then we conducted the social correlation analysis (similar to Figure 7(a), but for different types of social ties). Figure 7(b) shows the analysis result. We have found that people seem to be more likely to share their happiness within the friend circle, but only share their unhappiness with their best friends.

Summary To very briefly summarize, we have the following intuitions:

1) The environmental attributes and activities of a person can affect her emotional state.
2) One's emotional state is highly time-dependent.
3) One's emotional state is influenced by her friends' emotions.

By leveraging these aspects, we formulate the problem of emotion prediction in a dynamic continuous factor graph model and propose a method referred to as MoodCast for inferring users' emotions in social networks.

4 Our Approach

In this paper, we formalize the problem of emotion prediction in a dynamic continuous factor graph model and propose an approach referred to as MoodCast to learn the model for inferring individuals' emotional states. Our basic idea is to define the correlations (as shown in Figure 1) using different types of factor functions. An objective function is defined based on the joint probability of the factor functions, thus the problem of emotion model learning is cast as learning model parameters that maximizes the joint probability of emotional states based on the input continuous dynamic network.

According to our observations, we could define three kinds of factor functions:

- **Attribute correlation factor function** $\{f(x_{ij}^t, y_i^t)\}_j$. It denotes the attribute value associated with each user v_i at time t.
- **Temporal correlation factor function** $h(y_i^{t'}, y_i^t)$,$t' < t$. It represents the dependency of one's emotional state at time t on her emotional state at the recent past time t'.
- **Social correlation factor function** $g(y_i^t, y_j^{t'})$,$t' < t$. It represents the influence of user v_j's emotion at time t' on user v_i's emotion at time t.

The three factors can be instantiated in different ways, reflecting our prior knowledge for different applications. Here, we use the mobile social network as the example to explain how we define the factor functions. Based on the attributes associated with each user in the mobile social network, we define the following attribute factor functions:

Location: The feature represents the location of each user. We use GPS and GSM data to locate the user. The location is usually denoted as the longitude and the latitude value. To reduce noisy data, we only keep locations where the mobile user stays for more than 10 seconds. In addition, we use k-means clustering [8] to cluster all appearing GPS or GSM location points into k small regions. The user's longitude and latitude value is transformed into the index of the region.

SMS text: The feature represents whether or not a word is contained in the Short Message Service (SMS) text message sending to or received from one's friends.

Calling logs: The feature represents that the user makes (or receives) a call to his friend.

Activity: The features represents what the user is doing. There are eight predefined categories for the activities in the annotation system. Besides, the users can also freely add tags to describe their activities. The feature is the index of the selected category.

All the factor functions are time-dependent. For example, when there is a new call at time t, then a factor function is defined. All the attribute factor functions are converted into binary functions. For example, $f(x_{i1}^t = 1, y_i^t = \text{`}positive')$ represents if the user v_i adds an annotation as "positive" at time t and the index of his GPS location is 1, then the feature value is 1, otherwise 0. Finally, for all the records in the historic attribute-value log X^t, we can accumulate all the attribute factor functions and obtain a local entropy for all users:

$$\frac{1}{Z_1}\exp\{\sum_{v_i \in V}\sum_{x_{ik}^t \in X}\alpha_k f_k(x_{ik}^t, y_i^t)\} \tag{2}$$

where α_k is the weight of function f_k and Z_1 is a normalization factor. Such a formulation is also used in Maximum Entropy model [9] and Conditional Random Fields [10]. Directly optimizing Eq. (2) would result in a (local) Maximum Entropy model, which is essentially similar to a SVM or Naive Bayes (NB)-based classification model. However, it cannot take advantage of the social correlation and the temporal correlation. We will use the local classification models as baselines and empirically compare with our MoodCast approach.

For the social correlation factor function, we define it based on pairwise network structure and the continuous-time information. That is, if user v_i and user v_j have a relationship, then we define a friend influence factor function as follows:

$$g(y_i^t, y_j^{t'}) = e^{-\sigma_1(t-t')}\exp\{\beta_{ji}(y_i^t - y_j^{t'})^2\} \tag{3}$$

where t' is the latest past time when friend v_j changed her emotion (i.e., the latest record of emotion change in X^t); $e^{-\sigma_1(t-t')}$ is user-independent time-decay factor; σ_1 is a predefined parameter. In the model, we assume users' emotions at time t are conditionally independent of all the past states given the recent past emotional states of their friends. The same assumption is used in HMM [11], Markov Random field [12], and Kalman Filters [13]. In addition, β_{ji} is the weight of the function, representing the influence degree of v_j on v_i. The user-independent time-decay factor $e^{-\sigma_1(t-t')}$ can be also defined in different ways, such as using a quadratic or logarithmic function. We define it as an exponential function so that the parameter σ_1 can be simply absorbed into the learning process by combining with parameter β_{ji}, thus the above feature function can be rewritten as $g(y_i^t, y_j^{t'}) = \exp\{-\beta_{ji}(t-t')(y_i^t - y_j^{t'})^2\}$.

For the temporal correlation factor function, we try to use it to model the decay of the user emotion based on her past emotional states:

$$h(y_i^{t'}, y_i^t) = e^{-\sigma_2(t-t')}\text{exp}\{\lambda_i(y_i^t - y_i^{t'})^2\} \tag{4}$$

where t' is the latest past time when the user v_i changed his emotional state; similarly, σ_2 is a predefined parameter; λ_i represents how likely user v_i changes her emotion. In reality, some users may easily change their emotional state while the emotional states of some other users may be more stable. Similarly, σ_2 can be also absorbed and we have $h(y_i^{t'}, y_i^t) = \exp\{-\lambda_i(t-t')(y_i^t - y_i^{t'})^2\}$.

Finally, a factor graph model is constructed based on this formulation. Typically, we hope that a model can best recover the emotional states Y, which can be represented by maximizing the conditional likelihood of all users' emotional states given the observation data. More specifically, by combining Eqs. (2)-(4) together, we can define the objective likelihood function as

$$\begin{aligned} p(Y|G^t) = \frac{1}{Z}\text{exp}\{ &\sum_{v_i \in V} \sum_{x_{ik}^t \in X} \alpha_k f_k(x_{ik}^t, y_i^t) \\ &+ \sum_{v_j \in NB(v_i)} \sum_{(y_i^t, y_j^{t'}) \in Y^t} -\beta_{ji}(t-t')(y_i^t - y_j^{t'})^2 \\ &+ \sum_{v_i \in V} \sum_{(y_i^t, y_i^{t'}) \in Y^t} -\lambda_i(t-t')(y_i^t - y_i^{t'})^2\} \end{aligned} \tag{5}$$

where Z is a normalization factor; $NB(v_i)$ denotes user v_i's neighbors in the network; $(y_i^t, y_j^{t'})$ indicates a pair of emotional states between user v_i and v_j recorded in Y^t.

Learning the factor graph model is to estimate a parameter configuration $\theta = (\{\alpha_k\}, \{\beta_{ji}\}, \{\lambda_i\})$ from a given historic attribute-value log X^t, which maximizes the log-likelihood objective function $\mathcal{L}(\theta) = \text{log}p_\theta(Y^t|G^t)$, i.e.,

$$\theta^\star = \arg\max_\theta \ \log\, p(Y^t|G^t\theta) \tag{6}$$

5 Model Learning

It is usually intractable to do exact inference in such a graphical probabilistic model. The intrinsic difficulty is to calculate the normalization factor Z, which sums up all possible configurations of Y that makes the complexity exponential to the number of nodes in the graph. Several methods have been proposed to address this problem. For example, [2] defines each factor as an integral (quadratic) function, thus Z can be calculated by a transformation to a multivariate Gaussian distribution. Some other methods such as Junction Tree [14] and Belief Propagation [15] are used to obtain an approximate solution. In this paper, we use a sampling-based Metropolis-Hastings (MH) algorithm [16], a particular Markov-chain Monte Carlo inference. The advantage of the MH algorithm is that it can derive a global gradient updates for each parameter, thus can obtain better performance. The MH-based learning algorithm mainly consists of two key ingredients: (1) a proposal distribution, which defines how likely a new conditional configuration should be accepted; (2) parameter update according to the training error. In the following, we explain the learning algorithm in details.

Input: number of iterations and learning rate η;
Output: learned parameters $\theta = (\{\alpha_k\}, \{\beta_{ji}\}, \{\lambda_i\})$;

Initialize $\theta = \{\alpha, \beta, \lambda\}$;
repeat
  % sample a new Y' according to $q(Y'|Y)$;
  $Y' \leftarrow q(Y'|Y)$;
  $\tau \sim \min(\frac{p(Y'|G^t,\theta)}{p(Y|G^t,\theta)}, 1)$;
  toss a coin s according to a $Bernoulli(\tau, (1-\tau))$;
  if $(s = 1)$ **then**
    % accept the new configuration Y';
    $Y \leftarrow Y'$;
    if $(Err(Y') < Err(Y)$ & $\Delta\theta F < 0)$ **then**
      $\theta^{new} \leftarrow \theta^{old} + \eta(\Delta\theta F)$;
    end
    else if $(Err(Y') > Err(Y)$ & $\Delta\theta F \geq 0)$ **then**
      $\theta^{new} \leftarrow \theta^{old} - \eta(\Delta\theta F)$;
    end
  end
until *convergence*;

Algorithm 1: The MH-based learning algorithm for MoodCast.

As summarized in Algorithm 1, in each iteration of the learning algorithm, by the Metropolis-Hastings algorithm, we first sample a new configuration Y' conditioned on Y according to a proposal distribution $q(Y'|Y)$, which is defined over all possible configuration space $\mathcal{Y}$. The algorithm accepts the new configuration with an acceptance ratio τ:

$$\tau \sim \min(\frac{p(Y'|G^t, \theta)}{p(Y|G^t, \theta)}, 1) \tag{7}$$

If the new configuration Y' is accepted, the algorithm continues to update the parameters θ. Basically, there are two cases to update the parameters. In both cases, we first calculate two scores: error Err and un-normalized likelihood difference $\Delta\theta F$. The error is simply the number of mistakenly predicted examples on the training data. That is, based on the currently learned parameters θ, if the model predicts that the emotional state of a user would be positive at time t and the user's emotional state is indeed (annotated as) positive, we say that the model makes a correct prediction; otherwise mistake. The un-normalized likelihood difference is calculated by $\Delta\theta F = \theta F(Y') - \theta F(Y)$, where $F(Y')$ is the exponential component of Eq. (5) (i.e., the formula without the normalization factor and the exponential function). Then in the first case, if the error $Err(Y')$ of the new Y' is lower than Y but the likelihood difference is negative (theoretically should be positive), the algorithm updates the parameters by

$$\theta^{new} \leftarrow \theta^{old} + \eta \Delta\theta F \quad (8)$$

In the second case, if error $Err(Y')$ of the new Y' is larger than Y but likelihood difference is positive (should be negative), the algorithm updates the parameters by

$$\theta^{new} \leftarrow \theta^{old} - \eta \Delta\theta F \quad (9)$$

The proposal distribution $q(Y'|Y)$ in Algorithm 1 can be defined in different ways. For simplicity, we use a random distribution, that is, from the current configuration Y, we randomly change the emotional state of each node and then obtain Y'. Another strategy is to use a heuristic algorithm to sample Y'. For example, we can calculate the un-normalized likelihood difference of a new configuration and in each iteration we try to find a configuration with the largest difference to update the parameters. However, surprisingly, we found that the random solution always clearly outperforms the heuristic-based proposal distribution, either on efficiency or effectiveness.

5.1 Mood Forecasting

Based on the learned parameter θ, we can infer users' future emotional states. Specifically, the prediction problem is cast as finding an configuration of emotional states to maximize the likelihood given the learned parameters and historic data. Formally, the problem is an instance of the Maximum a Posteriori (MAP) problem as follows:

$$\arg\max_{y \in \mathcal{Y}} p(Y = y | G^t, \theta)$$

The prediction algorithm is also based on the Metropolis-Hastings algorithm. The framework of the prediction algorithm is summarized in Algorithm 2. We first initialize a configuration of emotional states Y. Then in each iteration, we sample a new configuration Y', and determine whether we accept the new configuration Y' using the same strategy as that in the model learning algorithm. If accepted, we compare it to the optimal solution we have obtained so far. If Y' is better ($\mathcal{L}(Y') > max$), we record it as the optimal solution.

We find that a "good" initialization of the emotion configuration Y can help improve the performance as well as the speed of convergence. The idea here is that we can initialize the emotional states as the results of maximizing the likelihood with only attribute factor functions Eq. (2). Then we perform the learning and prediction from this start point using the Metropolis-Hastings algorithm.

We can also consider some other algorithms for forecasting users' emotional states, such as Support Vector Machines (SVM), Naive Bayes (NB) and Conditional Random Fields [10]. In existing work, Bollen et al. [6] employs self organizing fuzzy neural network for predicting users' emotional states on Twitter. However, most existing methods do not consider correlations in the social network, thus cannot leverage social networks to help the prediction. In the experimental section, we will compare the performance of our proposed approach with the SVM-based method and the NB-based method.

Input: Learned parameters θ, model and data G^t;
Output: Emotional states Y;

Initialize the emotional states Y;
$output \leftarrow Y$;
$max \leftarrow -\infty$
repeat
  $Y' \leftarrow q(Y'|Y)$;
  $\tau \sim \min(\frac{p(Y'|G^t,\theta)}{p(Y|G^t,\theta)}, 1)$;
  toss a coin s according to a $Bernoulli(\tau, (1-\tau))$;
  if $(s = 1)$ **then**
    $Y \leftarrow Y'$;
    %$\mathcal{L}(Y')$ is a un-normalized log-likelihood;
    if $\mathcal{L}(Y') > max$ **then**
      $max \leftarrow \mathcal{L}(Y')$;
      $output \leftarrow Y$;
    end
  end
until *convergence*;

Algorithm 2: The MH-based prediction algorithm for MoodCast.

TABLE 1
Statistics of the MSN and the LiveJournal data sets.

Data set	Users	Links	Avg. links	Label	Avg. Labels
MSN	30	96	3.2	9,869	329
LiveJournal	469,707	23,318,572	49.6	2,665,166	5.7

6 Experimental Results

In this section, we present experimental results of the proposed MoodCast to evaluate its effectiveness and efficiency.

6.1 Experimental Setup

Data Sets We performed our experiments on two different genres of data sets: one real mobile social network (MSN) and one virtual web-based network (LiveJournal). Statistics of the two data sets are shown in Table 1.

In the MSN data set, we have collected communication (by SMS text and call), calendar, alarm, Wifi signal, GPS location, activity, and mood information from 30 volunteers at a university from May to July, 2010. This data represents over 36,000 hours of continuous data on human behavior and emotional state. In total, there are about 9,869 human labels of emotional states. Users in the MSN data set form a small social network.

The LiveJournal data set was collected from LiveJournal,[2] a social media platform where users share common

2. http://www.livejournal.com

passions and interests. Basically, users on LiveJournal can post what they are doing, how they are feeling, and give comments to posts of their friends. The system also allows users to associate a "mood label" to each of their posts. We suppose that the associated mood label indicates the emotional state of the user when she published the post. Besides the mood labels provided by the website, LiveJournal also allows the users to define new mood labels in their own interest. We conducted an analysis on the 155 most commonly used mood labels and classify them into three categories: positive, negative, and neutral. Table 2 shows the three categories and those typical mood labels in each category.

We collected the LiveJournal data set in the following ways. First we chose the administrator of the computer science community as the seed user and used a crawler to extract her friends. Then for each user, we extracted her mood labels and friends. With the recursively crawling process, we eventually obtained a data set of over 5GB. The derived social network from the data set (as shown in Table 1) consists of 469K users and 23 million friendships between these users. On average, each user has 49.64 friends and publishes 5.64 posts on the website. We preprocessed the LiveJournal data set by (a) removing users (and their posts) who posted less than 10 mood labels (as assigning the mood label to a post is optional, a portion of users did not associate mood labels to their posts); (b) removing posts that are not in English; and (c) converting all words into lower case. After that, the remaining data set contains 117,060 users with 2 million posts. The user attributes are words used in their posts. Specifically, we first created a vocabulary W of words appearing in all posts. Then for each post we create a $|W|$-dimension vector, with each binary-valued element indicates whether the post contains the corresponding word.

In the LiveJournal data set, we found that using all words will result in a large number of features, which introduced a lot of noise and slowed down the learning process. Thus, we used a method for word selection. The basic idea is to measure each word by a score:

$$s(w_i) = \left| \frac{POS_{w_i} - NEG_{w_i}}{N_{w_i}} \right| \cdot \log(N_{w_i}) \qquad (10)$$

where N_{w_i} is the number of posts that contain word w_i; POS_{w_i} and NEG_{w_i} respectively denote the numbers of positive and negative posts that contain w_i. The intuition is that those words with higher discriminative power will have a higher score. Finally, words with scores greater than a certain threshold were selected. In the following subsections, we will analyze how the threshold affects the prediction performance.

Baseline Methods We define four baseline methods for the emotion prediction task.

- *SVM-Simple.* The method only uses user attributes (i.e., user attribute factor functions) as features to train a classification model and then employs the classification model to predict the user mood. For SVM, we use SVM-light [3].
- *SVM-Net.* Besides using user attributes, the method also includes the network information (social correlation and temporal correlation) as features, i.e., the method uses the same features as in our MoodCast approach.
- *Naive Bayes (NB-Simple).* The method uses the same features as that in the SVM method. The only difference is that it uses the Naive Bayes as the classification model.
- *Naive Bayes (NB-Net).* It uses the same features as that in the SVM-Net and uses Naive Bayes as the classification model.

TABLE 2
Examples of mood label classification.

Category	Typical mood label
Positive	fantastic, great, elated, bouncy, jubilant, excited, cheerful, ecstatic
Neutral	normal, awake,calm, determined, working, blah, blank, lazy, thoughtful
Negative	annoyed, aggravated, bad, pain, embarrassed, bored, anxious, crazy, depressed, sad, scared, sick, sore

SVM and NB are usually designed for binary classification. For fair comparison of the proposed MoodCast with the four baselines, we reduced the multiple-classification problem (multiple mood labels) to several binary classification problems, by training a binary classification model for each category of mood labels. For instance, for training a model for the category "happy", we took all data with the "happy" label as positive instances and the other as negative instances. To predict the emotional state of a user, we applied all the trained models to classify the user, and finally took the category which has the highest confidence (distance to the hyperplane in SVM and posterior probability in NB).

Evaluation Measures In all experiments, we evaluated the mood prediction performance in terms of Precision, Recall, and F1-Measure. We also used several case studies as the anecdotal evidence to further demonstrate the effectiveness of our method.

The learning algorithm for MoodCast has been implemented in C++ and compiled with the Visual Studio 2008 compiler. All experiments were conducted on the server with Windows Server 2003, Intel Xeon(TM) CPU 3.20GHz and 4Gb memory.

6.2 Mood Forecasting Performance

On both data sets, we used the historic data (time 0 to $t-1$) of all users as the training data and then to infer users' emotional state at time t. In particular, in the MSN data set, we chose the data in the last 4 days as the test data and the rest as the training data. In the LiveJournal data set, we chose the data in the latest 3 months as the

3. http://svmlight.joachims.org/

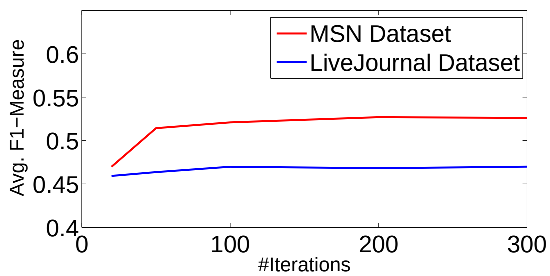


Fig. 8. The influence of sampling iteration times.

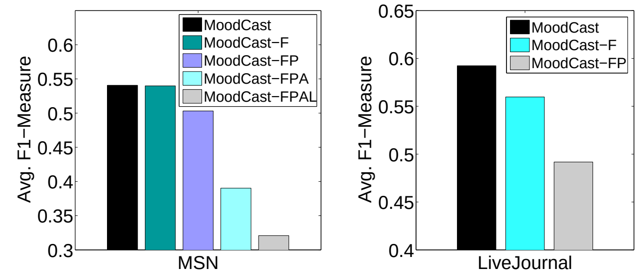


Fig. 9. Contribution of different factor functions. The left is MSN result and the right is LiveJournal result. MoodCast-F stands for ignoring friend influence factor. MoodCast-FP stands ignoring both the friend and time-dependent factor. MoodCast-FPA stands for further ignoring activity attribute and MoodCast-FPAL for further ignoring location attribute.

test set and the rest as the training data set. Table 3 lists the prediction performance of the different approaches on the two data sets with the following observations:

From the results, we see that our method clearly outperforms the baseline methods on both data sets. On average, MoodCast achieves a 8+% improvement compared with both SVM and Naive Bayes methods. Moreover, we see that MoodCast results in a more stable performance while SVM varies greatly on LiveJournal. We analyzed the result by SVM and found that the poor performance of SVM might be due to the sparse values of user attributes. The attributes of LiveJournal data set are keywords in their posts. However, there are many posts that do not contain the discriminative words. For example, a user posted "Survey, everyone read, you will know me better" which contains no useful words for predicting the emotional state. Solely considering the attribute information (as in SVM-Simple and NB-Simple) is difficult to accurately infer users' emotional states.

From Table 3, we can also see that simply combining all the features (social correlation, temporal correlations) together (as in SVM-Net and NB-Net) can improve the prediction performance, but the performance is still not satisfactory. Our method can leverage friends' influence information and users' historic emotion information, thus achieves a better performance.

6.3 Analysis and Discussion

To give a deeper understanding of the results, we performed the following analysis.

Effect of the number of sampling iterations We conducted an experiment to see the effect of the number of the sampling iterations. We use the average F1-Measure of the three classifiers to measure the overall performance. Figure 8 illustrates the experiment result. We see our MH-based learning algorithm converges in less than 100 iterations on both data sets, which suggests that the learning algorithm is very efficient and has a good convergence property.

Factor contribution analysis In MoodCast, we consider multiple different factor functions: friend influence (F), time-dependent (P), and user activity (A) and locations (L). Here we perform an analysis to evaluate the contribution of the different factor functions in our model. We first ranked the individual factors by their predictive power, then removed those factors one by one in the reversing order of their predictive power. In particular, we first removed the social correlation factor function denoted as MoodCast-F, followed by removing the time-dependent factor function denoted as MoodCast-FP, and then evaluated and compared the prediction performances of the different versions of MoodCast. In the MSN data set, we further removed the activity information (MoodCast-FPA) and the location information (MoodCast-FPAL).

Figure 9 shows the average F1-Measure score after ignoring the factor functions. We can observe clear drop on the performance when ignoring some of the factors. This indicates that our method works well by combining the different factor functions and each factor in our method contributes to the performance. The only exception is that when ignoring the social correlation influence factor function on the MSN data set, there is no effect on the prediction performance. This is a bit surprising. Intuitively, in the real mobile network, users may be influenced by friends with a stronger degree than users in the virtual social network. By carefully investigating the data set, we found that in our MSN data set, the friendship network is sparse (averagely each user only has 3.2 friends) and the mobile users seldom contact with each other through mobiles (this might be due to the limited number of participants).

Effect of Word Selection In the LiveJournal data set, we created a $|W|$-dimension vector for each post as its user attributes where W is the vocabulary. In our experiment, we selected a score (Cf. Eq. 10) threshold to decide which words we should put into the vocabulary. We analyzed the effect of different thresholds on the performance. We set different thresholds and compared the average F1-Measure of MoodCast. Figure 10 shows the result. From it we see that the performance is the best when the threshold is about 2. If we chose a small threshold, we will introduce more noise which hurts the performance. Moreover, when the threshold is small, W will be large

TABLE 3
Performance of mood prediction on Mobile Social Network and LiveJournal by different approaches(%).

Classifier	Method	MSN Data set			LiveJournal Data set		
		Precision	Recall	F1-Measure	Precision	Recall	F1-Measure
Positive	MoodCast	68.42	69.23	**68.82**	52.50	73.68	**61.32**
	SVM-Simple	60.88	71.08	65.58	49.56	48.57	49.06
	SVM-Net	59.12	72.70	65.21	50.72	60.29	55.09
	NB-Simple	67.30	56.21	61.25	57.08	43.34	49.27
	NB-Net	71.89	56.59	63.33	59.1	47.38	52.59
Neutral	MoodCast	67.78	76.57	**71.90**	59.61	84.92	**75.44**
	SVM-Simple	67.39	59.73	63.33	67.58	78.69	72.71
	SVM-Net	68.42	55.11	61.05	71.21	78.13	74.51
	NB-Simple	54.14	68.04	60.30	65.95	54.14	59.46
	NB-Net	51.06	71.62	59.62	61.70	61.53	61.61
Negative	MoodCast	30.77	13.95	19.20	45.45	54.98	**49.77**
	SVM-Simple	5.63	4.54	5.03	71.67	37.39	49.14
	SVM-Net	8.18	16.90	11.02	68.78	37.68	48.68
	NB	14.70	28.16	19.32	54.77	36.61	43.89
	NB-Net	17.88	32.08	**22.96**	51.70	41.18	45.84
Average	MoodCast	55.66	53.25	**53.31**	52.52	71.19	**62.17**
	SVM-Simple	44.63	45.12	44.65	62.94	54.83	56.97
	SVM-Net	45.24	48.23	45.76	63.57	58.70	59.42
	NB-Simple	45.38	50.80	46.95	59.26	44.69	50.87
	NB-Net	46.94	53.43	48.63	57.5	50.03	53.35

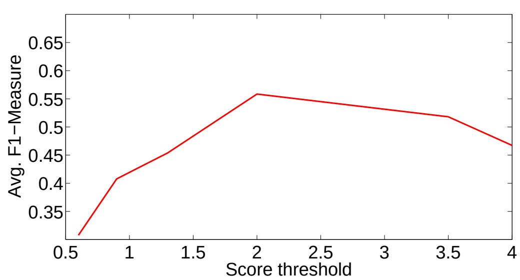


Fig. 10. The effect of word selection. The x-axis stands for the score threshold of creating W and the y-axis stands for the average F1-Measure of three classifiers.

and the computational efficiency will be low. On the other hand, if we chose a large threshold, the vocabulary W will be small and many posts have a 0 attribute vector. The sparse attribute vectors will also result in a performance poor. After considering both effectiveness and efficiency aspects, we chose 1.8 as our threshold (about 40 words) in our experiment.

6.4 Case Study of Happiness Propagation

Here we present two case studies to demonstrate that our model correctly handles the happiness propagation which is a very common phenomenon in the real world.
Home party On 9 May (Sunday), several users in the mobile social network went to a party at their advisor's home. Most of the users are friends with each other and they all labeled a good or wonderful mood that night. We see that our model can track this happiness event.
Jokes of a chicken and a bartender Another interesting case is from LiveJournal, which happened on 21st June, a user posted a joke called "a chicken and a bartender" and labeled cheerful. Some of her friends such as user b and c are predicted as having a positive mood in the next few days, which is confirmed in the subsequent LiveJournal log. The reason for it can be explained by the fact that they share this joke and post comments to this joke. They are amused by the joke and post comments labeled with positive mood.

6.5 Error Analysis

We conducted an error analysis on the results of our approach. We observed three major types of source of errors.

(1) *Unpredictable emotion*. Our results show that more than 65% of users' emotional states are predictable, although there is still a portion of users' moods are unpredictable, in particular in the real mobile social network. For example, in the MSN data, user "Bo" was unhappy at a time without any hints, while all his friends were happy. We finally found "Bo" had not contacted with his friends for a long time.
(2) *Missing data*. Sometimes the data is missing because the users do not label it. For example, one test case is that user "Yuan" was in a new place and did not label what he was doing. Activity is a very important hint for the user's emotional state. Thus it is difficult to infer his mood around that time.
(3) *Unexpected labeling*. In both LiveJournal and mobile data sets, we observed some unexpected mood labeling. For example, user x posted "That makes me happy" on 05 June 2008 but labeled distress. Similar phenomenon also exists in mobile data. For example, user "mit" was playing but his mood became worse than before; while our statistical study shows that in more than 90% cases, playing will make user happier than before. Such phenomenon could be explained as (i) some special things happen (e.g., the user keeps on losing) and

the historical experience does not help (ii) or the user may wrongly label her mood.

7 Related work

Dynamic Emotion Analysis Some previous work, mostly in the fields of psychology and sociology, has investigated the dynamics of human emotion, including how certain emotions evolve into being, change over time, and affect emotional states of friends. In [5], Fowler et al. studied the dynamic spread of happiness in a social network. They have found that within a social network, happiness spreads among people up to three degrees of separation, which means when you feel happy, everyone within three degrees to you has a higher likelihood to become happy too. Later Cacioppo et al. [17] performed further analysis on the dynamics of loneliness. It has been shown that the spread of loneliness follows similar patterns as happiness. However, most of these works only qualitatively test the spread of emotion on small data sets. Our previous work [18] also studied the problem of emotion prediction. In this work, we aim to provide a more comprehensive analysis to this problem. To the best of our knowledge, no previous work extensively studies what factors are the underlying subtle forces to affect the emotion dynamics.

Dynamic Behavior Analysis Quite a few works have been conducted for social dynamic behavior analysis. Here we list a few examples of them. Tang et al. [19] built a topical factor graph model to measure the influential strength in the social network. And Tan et al. [2] proposed a noise tolerant model for predicting user's actions in online social networks. Goyal et al. [3] investigated how to learn the influence probabilities from the history of users' actions. All these works consider only user actions, but do not consider users' emotions. Backstrom et al. [4] proposed a partitioning on the data that selects for active communities of engaged individuals. Eagle et al. [20] presented a method for measuring human social behavior based on mobile phone data. Also, some works have been conducted in terms of social science, trying to identify the principles underneath human beings' behaviors in a social network. The works of Rosenquist et al. [21] and Fowler et al.[22] are among the representative researches in this field. However, none of these works aims to model and predict the emotional states of users in a social network.

Social Network Analysis Considerable research has been conducted for dynamic social network analysis and social influence. Kossinets et al. in [23] provided statistical analysis on an empirical data set, revealing interesting patterns about how nodes' interaction affect the network structure. Onnela et al. [24] studied the correlation between interaction strength and network structure. And Xing et al. [25] proposed a random field based model to infer the interaction between nodes with the samples of users' emotional states at different time slots.

Winter et al. [26] investigated several models of social influence from an interdisciplinary view. Anagnostopoulos et al. [27] theoretically interpreted social influence as a source of social correlation when the time series of user actions is available. Singla and Richardson [28] studied the correlation between personal behaviors and their interests. Their study finds that people who chat online with each other are more likely to share interests, and the time they spend chatting is positively related to the relation strength. Crandall et al. [29] further investigated the correlation between social similarity and influence. More recently, La Fond and Neville [30] examined the effects of social influence on people's opinion.

Gomez-Rodriguez et al. [31] proposed an effective model to track the flow of information and influence in an online social network. Leskovec et al. [32] studied the problem of positive and negative link prediction in online social networks. Lin et al. [33] studied the problem of tracking popular events in online communities. However, most of these works focus on macro-level analysis (e.g., correlation analysis or influence analysis) or social network formation (e.g., behavior analysis and link prediction), but ignore users' emotional states.

Sentiment Analysis Sentiment analysis deals with the machine analysis of opinion, sentiment, and subjectivity in text [34]. A lot of research has been done on sentiment analysis for online reviews and blogs. Turney [35] detected document sentiments based on pre-specified part-of-speech patterns. TREC 2006-2008 had tracks devoted to opinion retrieval tasks.

In general, approaches for sentiment analysis include unsupervised approach, supervised approach, and language model. The unsupervised approach often first creates a sentiment lexicon in an unsupervised manner, and then determines the degree of positivity (or subjectivity) of a textual unit via some function based on the positive and negative (or simply subjective) indicators, as determined by the lexicon. Early examples of such an approach include Hatzivassiloglou and Wiebe [36], Turney [35] and Yu and Hatzivassiloglou [37]. Another type of approach is classification-based approach. The basic idea is to train a classification model using some manually labeled data and then use the trained classification model to predict the polarity of a document or a sentence. Typical examples of such an approach include Pang and Lee [38], Popescu and Etzioni [39], Goldberg and Zhu [40]. However, existing work has focused on sentiment analysis for particular documents or textual unites. At a general concept level, our problem can be linked to user-level sentiment analysis [41], [42]. However, the approaches proposed in [41], [42] only consider the static social network. To the best of our knowledge, no previous work considers the emotion analysis in the context of dynamic social network.

8 Conclusion

In this paper, we have studied a novel problem of emotion prediction in social networks. We have proposed a method referred to as MoodCast for modeling and predicting emotion dynamics in the social network. MoodCast formalizes the problem into a dynamic continuous factor graph model and defines three types of factor functions to capture the different types of information in the social network. For model learning, it uses a Metropolis-Hastings algorithm to obtain an approximate solution. Experimental results on two different real social networks demonstrate that the proposed approach can effectively model users' emotional states and the prediction performance is better than several baseline methods for emotion prediction.

The general problem of emotion dynamic analysis represents a new and interesting research direction in social network mining. There are many potential future directions of this work. A direct adaptation is to consider indirect influence [5], [1] into the factor graph model. So far, our MoodCast only considers one-degree influence. We can also consider the different levels of the same type of emotion since people may be happy or sad because of different reasons. One possible method is to make the parameters λ dependent on time t. Another interesting issue is to extend the proposed factor graph model so that it can learn a joint model for predicting emotions and actions simultaneously.

Acknowledgments

The work is supported by the Natural Science Foundation of China (No. 61073073 , No. 60973102), Chinese National Key Foundation Research (No. 60933013 , No.61035004), National High-tech R&D Program (No. 2009AA01Z138). It is also supported by a research award from Nokia China Research Center.

References

[1] J. Whitfield, "The secret of happiness: grinning on the internet," in *Nature*, 2008.

[2] C. Tan, J. Tang, J. Sun, Q. Lin, and F. Wang, "Social action tracking via noise tolerant time-varying factor graphs," in *KDD'10*, 2010, pp. 1049–1058.

[3] A. Goyal, F. Bonchi, and L. V. Lakshmanan, "Learning influence probabilities in social networks," in *Proceedings of the 3st ACM International Conference on Web Search and Data Mining (WSDM'10)*, 2010.

[4] L. Backstrom, R. Kumar, C. Marlow, J. Novak, and A. Tomkins, "Preferential behavior in online groups," in *WSDM'08*, 2008, pp. 117–128.

[5] J. H. Fowler and N. A. Christakis, "Dynamic spread of happiness in a large social network: longitudinal analysis over 20 years in the framingham heart study," in *British Medical Journal*, 2008.

[6] J. Bollen, H. Mao, and X.-J. Zeng, "Twitter mood predicts the stock market," *Journal of Computational Science*, vol. 2, no. 1, pp. 1–8, 2011.

[7] K. A. Goyal and A. Sadasivam, "A critical analysis of rational & emotional approaches in car selling," *International Journal of Business Research and Management (IJBRM)*, vol. 1, no. 2, pp. 59 – 63, 2010.

[8] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281–297.

[9] A. L. Berger, V. J. D. Pietra, and S. A. D. Pietra, "A maximum entropy approach to natural language processing," *Comput. Linguist.*, vol. 22, no. 1, pp. 39–71, 1996.

[10] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *ICML'01*, 2001, pp. 282–289.

[11] Z. Ghahramani and M. I. Jordan, "Factorial hidden markov models," *Machine Learning*, vol. 29, no. 2-3, pp. 245–273, 1997.

[12] J. M. Hammersley and P. Clifford, "Markov field on finite graphs and lattices," *Unpublished manuscript*, 1971.

[13] S. S. Haykin, *Kalman Filtering and Neural Networks*. New York, NY, USA: John Wiley & Sons, Inc., 2001.

[14] W. Wiegerinck, "Variational approximations between mean field theory and the junction tree algorithm," in *UAI'00*, 2000, pp. 626–633.

[15] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Generalized belief propagation," in *NIPS'01*, 2001, pp. 689–695.

[16] S. Chib and E. Greenberg, "Understanding the metropolischastings algorithm," *American Statistician*, vol. 49, no. 4, p. 327C335, 1995.

[17] J. T. Cacioppo, J. H. Fowler, and N. A. Christakis, "Alone in the crowd: The structure and spread of loneliness in a large social network," *Journal of Personality and Social Psychology*, 2009.

[18] Y. Zhang, J. Tang, J. Sun, Y. Chen, and J. Rao, "Moodcast: Emotion prediction via dynamic continuous factor graph model," in *ICDM'10*, 2010, pp. 1193–1198.

[19] J. Tang, J. Sun, C. Wang, and Z. Yang, "Social influence analysis in large-scale networks," in *KDD'09*, 2009, pp. 807–816.

[20] N. Eagle, A. S. Pentland, and D. Lazer, "Inferring social network structure using mobile phone data," *PNAS*, vol. 106, no. 36, 2009.

[21] J. N. Rosenquist, J. Murabito, J. H. Fowler, , and N. A. Christakis, "The spread of alcohol consumption behavior in a large social network," *Annals of Internal Medicine*, vol. 152, no. 7, pp. 426–433, 2010.

[22] J. H. Fowler and N. A. Christaki, "Cooperative behavior cascades in human social networks," *PNAS*, 2010.

[23] G. Kossinets and D. J. Watts, "Empirical analysis of an evolving social network," *Science*, vol. 311, pp. 88–90, 2006.

[24] J. P. Onnela, J. Saramaki, J. Hyvonen, G. Szabo, D. Lazer, K. Kaski, J. Kertesz, and A. L. Barabasi, "Structure and tie strengths in mobile communication networks," *PNAS*, vol. 104, pp. 7332–7336, 2006.

[25] A. Ahmed and E. P. Xing, "Recovering time-varying networks of dependencies in social and biological studies," *PNAS*, vol. 106, 2009.

[26] W. A. Mason, F. R. Conrey, and E. R. Smith, "Situating social influence processes: Dynamic, multidirectional flows of influence within social networks," 2007.

[27] A. Anagnostopoulos, R. Kumar, and M. Mahdian, "Influence and correlation in social networks," in *KDD'08*, 2008, pp. 7–15.

[28] P. Singla and M. Richardson, "Yes, there is a correlation: - from social networks to personal behavior on the web," in *WWW'08*, 2008, pp. 655–664.

[29] D. Crandall, D. Cosley, D. Huttenlocher, J. Kleinberg, and S. Suri, "Feedback effects between similarity and social influence in online communities," in *KDD'08*, 2008, pp. 160–168.

[30] T. La Fond and J. Neville, "Randomization tests for distinguishing social influence and homophily effects," in *WWW'10*, 2010, pp. 601–610.

[31] M. Gomez-Rodriguez, J. Leskovec, and A. Krause, "Inferring networks of diffusion and influence," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'10)*, 2010, pp. 1019–1028.

[32] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Predicting positive and negative links in online social networks," in *WWW'10*, 2010, pp. 641–650.

[33] C. X. Lin, B. Zhao, Q. Mei, and J. Han, "Pet: A statistical model for popular events tracking in social communities," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'10)*, 2010, pp. 929–938.

[34] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Found. Trends Inf. Retr.*, vol. 2, pp. 1–135, January 2008.

[35] P. D. Turney, "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL'02)*, 2002, pp. 417–424.

[36] V. Hatzivassiloglou and J. M. Wiebe, "Effects of adjective orientation and gradability on sentence subjectivity," in *Proceedings of the 18th conference on Computational linguistics (COLING'00)*, 2000, pp. 299–305.
[37] H. Yu and V. Hatzivassiloglou, "Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences," in *Proceedings of the 2003 conference on Empirical methods in natural language processing (EMNLP'03)*, 2003, pp. 129–136.
[38] B. Pang and L. Lee, "A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts," in *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics (ACL'04)*, 2004.
[39] A.-M. Popescu and O. Etzioni, "Extracting product features and opinions from reviews," in *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT'05)*, 2005, pp. 339–346.
[40] A. B. Goldberg and X. Zhu, "Seeing stars when there aren't many stars: graph-based semi-supervised learning for sentiment categorization," in *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, 2006, pp. 45–52.
[41] C. Tan, L. Lee, J. Tang, L. Jiang, M. Zhou, and P. Li, "User-level sentiment analysis incorporating social networks," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD'11)*, 2011.
[42] R. Agrawal, S. Rajagopalan, R. Srikant, and Y. Xu, "Mining newsgroups using networks arising from social behavior," in *Proceedings of the 12th international conference on World Wide Web (WWW'03)*, 2003, pp. 529–535.

Jie Tang is an associate professor in Tsinghua University. His research interests are social network analysis, data mining, and machine learning.

Yuan Zhang is a undergraduate student in Tsinghua University. His research interests are machine learning and information retrieval.

Jimeng Sun is a researcher from IBM TJ Watson. His research interests include large-scale data mining, social networks and health care analytics.

Jinghai Rao is a researcher from Nokia Research Center. His research interests include semantic web and data mining.

Wenjing Yu is a research assistant from the Department of Computer Science of Tsinghua University. Her research interests are text mining and mobile social network analysis.

Yiran Chen is a undergraduate in Tsinghua University. His research interests are text mining and mobile social network analysis.

A.C.M Fong is a professor in the School of Computing & Mathematical Sciences, Auckland University of Technology. He has published widely in the areas of data mining and communications.