Measuring Temporal Patterns in Dynamic Social Networks

WEI WEI, Carnegie Mellon University
KATHLEEN M. CARLEY, Carnegie Mellon University

Given social networks over time, how can we measure network activities across different timesteps with a limited number of metrics? We propose two classes of dynamic metrics for assessing temporal evolution patterns of agents in terms of persistency and emergence. For each class of dynamic metrics, we implement it using three different temporal aggregation models ranging from the most commonly used Average Aggregation Model to more the complex models such as the Exponential Aggregation Model. We argue that the problem of measuring temporal patterns can be formulated using Recency and Primacy effect, which is a concept used to characterize human cognitive processes. Experimental results show that the way metrics model Recency–Primacy effect is closely related to their abilities to measure temporal patterns. Furthermore, our results indicate that future network agent activities can be predicted based on history information using dynamic metrics. By conducting multiple experiments, we are also able to find an optimal length of history information that is most relevant to future activities. This optimal length is highly consistent within a dataset and can be used as an intrinsic metric to evaluate a dynamic social network.

Categories and Subject Descriptors: H.2.8 [**Information Systems-Database Applications**]: Data Mining

General Terms: Measurement, Algorithms, Performance

Additional Key Words and Phrases: Social network analysis, aggregating method, temporal analysis, Primacy and Recency effects

ACM Reference Format:
Wei Wei and Kathleen M. Carley. 2015. Measuring temporal patterns in dynamic social networks. ACM Trans. Knowl. Discov. Data. 10, 1, Article 9 (July 2015), 27 pages.
DOI: http://dx.doi.org/10.1145/2749465

1. INTRODUCTION

A social network can be modeled mathematically by a graph $G = \{V, E\}$ which consists of a set of agents V and a set of edges E connecting them. A dynamic social network consists of a series of observations of social networks at different timesteps $\{G_1, G_2, \ldots, G_n\}$. A dynamic social network contains not only a set of relationships between agents, but also information on how these relationships change over time. Understanding how temporal factors affect the evolutions of network structures is a

This work is part of the dynamics networks project at the center for Computational Analysis of Social and Organizational Systems (CASOS) of the school of Computer Science (SCS) at Carnegie Mellon University (CMU). This work was supported in part by the Office of Naval Research (ONR) through a MURI N000140811186 on adversarial reasoning, a MINERVA N000141310835 on trust and media, the Defense Threat Reduction Agency (DTRA) HDTRA11010102, and by Center for Computational Analysis of Social and Organization Systems (CASOS). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the ONR, the DTRA, or the US government.

Authors' addresses: W. Wei and K. M. Carley, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA; emails: {weiwei, kathleen.carley}@cs.cmu.edu.

ACM 1556-4681/2015/07-ART9 $15.00
DOI: http://dx.doi.org/10.1145/2749465

persistent research topic in social network analysis. Recently, there have been many works focusing on temporal analysis such as link prediction [Dunlavy et al. 2011], graph evolution [Leskovec et al. 2007; Leskovec et al. 2008], and community evolution [Lin et al. 2009].

One important issue in the study of dynamic social networks is how to measure network activity patterns across different timesteps. For example, we might be interested in the message-sending patterns overtime of a particular agent in the social network. Such analysis can be useful for a number of reasons. (1) Understanding the temporal patterns of agents, a critical step in abnormality detection. In an online social network, a sharp decrease in network activities indicates potential abnormalities such as account intrusion or difficulties accessing service. On the other hand, a sudden increase might indicate, for example, the introduction of a bot-spammer. (2) Temporal activity analysis is a keystone to the problem of network evolution patterns. Future network activities can be predicted based on temporal patterns learned from the existing networks. (3) Repeated temporal patterns suggest certain temporal invariants in the dynamic social network. Those temporal factors are critical to the understanding of the periodic nature of many dynamic social networks.

In this article, we propose two types of dynamic metrics to assess temporal patterns in dynamic social networks – *Persistence* and *Emergence*. Persistence, which aggregates agents' activities over time, is a class of metrics to evaluate the extent to which an agent maintains its level of activity. Emergence, which compares activity in two adjacent timesteps, is a class of metrics to evaluate the suddenness of change in network activities. Each of these dynamic metrics relies on a specific temporal aggregation model, which synthesizes a series of temporal network activities into single numerical representations. Here, the network activities are measured per timestep using static network metrics such as degree centrality, clustering coefficients, or authority centrality. They will be aggregated through one of the temporal aggregation models and form either a Persistence or an Emergence metric. Clearly there can be many different models to aggregate historic data and each has its advantages and disadvantages. In this article, we evaluate three particular temporal aggregation models: average, linear, and exponential. The *Average Model* aggregates temporal information by assigning an equal weight to each timestep. The *Linear Model* assigns linearly increased weights, and the *Exponential Model* assigns exponentially increased ones to each timestep. We will see how those aggregation models affect the performance of the models later in the article.

Since each dynamic metric is an evaluation of temporal patterns, we can use it to approximate activity patterns in the future. We use a regression model to evaluate the performance of each metric by using it to predict future network activities. Experimental results show that the performances of the metrics are closely related to the ways that they replicate Recency–Primacy effects. From a human cognition perspective, Primacy is the extent to which initial information dominates the perception. In contrast, Recency is the extent to which the most current information dominates the perception.

The contributions of this article are threefold. First, we provided two efficient dynamic metrics to effectively measure temporal patterns in social networks. Each metric measures one of two important aspects of temporal patterns. Second, we built a systematic approach to assess the performance of our metrics. The approach is based on the fact that dynamic metrics are both a representation of historical activity patterns and an approximation of future activity. Third, we provide statistical evidences that dynamic social networks have intrinsic temporal invariants. Activity patterns in social networks are highly dependent on historical activity information within a specific time frame.

2. BACKGROUND

2.1. Evolution of Dynamic Social Networks

Social networks are in a constant state of flux [Palinkas et al. 1998; Romney 1988; Sampson 1968; Sanil et al. 1995]. Unsurprisingly, then, studies of network dynamics abound. Early studies in this area suggested that social networks are generated according to random graph models with a fixed probability p that dominates the successful rate that new links added to the networks [Erdős and Rényi 1960]. The fact that this random graph model does not exhibit clustering behaviors, which are generally observed in real social network data, resulted in more realistic network models such as the small world model [Watts and Strogatz 1998] being developed. In a small world network, the distance between two arbitrary agents is proportional to the logarithm of the number of agents in the network. Recent findings on large-scale social networks data [Faloutsos et al. 1999; Newman 2001b, 2004] suggest that the degree distribution of networks may be fairly accurately modeled as a power law [Albert and Barabási 2002; Barabási and Albert 1999; Newman 2005]. A power-law network can be generated by the process called preferential attachment [Newman 2001a], which models agent's ability to attract links as being proportional to the total number of their already established links. However, in real social networks, links grow and decay leading to the need for incremental metrics that handle both positive and negative network changes [Kas et al. 2015]. We design temporal aggregation metrics that account for both tie growth and atrophication.

2.2. Primacy and Recency Effects

Primacy and Recency are often used to characterize the cognitive process of humans [Hovland 1957]. The Primacy effect argues that people remember the very first things better than the later ones [Deese and Kaufman 1957; Murdock Jr, 1962]. The Recency effect, on the other hand, argues that people recall the latest things better than the earlier ones [Miller and Campbell 1959]. These concepts are useful in understanding cognitive processes because they illustrate basic strategies used by humans to measure temporal impacts. We design temporal aggregation metrics that effectively account for primacy and recency.

2.3. Network Metrics

Evaluating the performance of network members plays a critical role in understanding the interactive patterns in a network. Numerous agent level metrics exist that measure the importance of an agent. Commonly used metrics include count-based centrality metrics such as degree centrality [Freeman 1979] and shortest path-based metrics such as betweenness centrality [Anthonisse 1971]. Another way to measure actor performance in a network is to evaluate the impact of node removal or addition. Borgatti defined the positive key player and negative key player problems by evaluating how some network level metrics change if a node is added or deleted [Borgatti 2003]. If deleting a node causes significant structural changes (e.g. making the networks to be disjoint), it will be recognized as a negative network elite. On the other hand, if adding a node makes positive changes to the network structure, it will score highly on the positive elite metrics. Carley used similar ideas, but developed metrics to identify elites by removing them and allowing the network to reconstitute itself [Carley 1990]. Other methods include the use of propagation models to evaluate network elites [Chen et al. 2003]. We note that all of these approaches are static assessments that are applied on a single network timestep. In contrast, for dynamic network data incremental versions of path-based metrics exist such as betweenness and k-betweenness [Kas et al. 2015] and closeness [Kas et al. 2013]. Similar to the metrics proposed by Kas, the metrics we have developed enable the assessment of change over different time periods. However, in contrast to Kas, our approach accounts for burstiness by supporting an averaging procedure.

2.4. Linear Regression Models

Given a random variable $y \in \mathbb{R}$ and a feature vector $X \in \mathbb{R}^d$, the linear regression model assumes that y can be represented as a linear function of X, that is $y = \beta_0 + (\beta_1, \ldots, \beta_d)X^T$. The goal of the regression model is to estimate $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_d)$ based on a set of training data $\{(X_1, y_1), \ldots, (X_n, y_n)\}$. One solution is to use linear least-squares regression [Hastie et al. 2005]. The linear least-squares regression developed a closed form to estimate $\hat{\beta}$ by minimizing the square error. However, in some cases where X is singular, the linear least squares regression failed to work. Lasso regression [Tibshirani 1996] and Ridge regression [Hoerl and Kennard 1970] formulated the regression problem to be an optimization task and eliminated the restrictions of X being nonsingular. To avoid overfitting, lasso regression model adds an L1 norm penalty term in the objective function while ridge regression adds a more strict L2 norm penalty term. There is also a technique called elastic net [Zou and Hastie 2005] that produces a penalty term between L1 and L2 terms. In this article, we choose lasso regression over ridge regression for our prediction model because lasso provided the benefit of allowing the coefficients to become zero. In ridge regression, however, the coefficients can only be either all nonzero or all zeros. This is useful when we have multiple features and we know some of them are not correlated to the target variable. In that case, lasso regression will disable the uncorrelated variables by setting the coefficients to zero.

2.5. Window-Based Aggregation Techniques

Window-based techniques are fairly common in research areas where a sequence of data points needed to be combined into one. There are many reasons to conduct aggregations on a data. For example, when the size of the data beyond the computational ability of computers, aggregation can significantly reduce those calculations but also achieve a good approximation. A window-based aggregation technique uses a window size to determine the amount of data used to generate a single aggregation point. There are many window-based aggregation methods that have been proposed. Bian and Butler [1999] analyzed the most popular window-based aggregation methods (i.e. average, mean, median) and how they affected the mean and standard deviation on a two-dimensional image dataset. However, we find that classic window-based aggregation methods cannot meet the requirements of the temporal pattern problem for the following reasons: (1) Classic window-based aggregation techniques use equal weights on all data to be weighted. This makes all the temporal data to be equally important, which is not reasonable when the time frame of the data gets long. (2) Classic window-based aggregation techniques can only report the magnitude of the data rather than the changing trends of a dynamic network.

3. DYNAMIC NETWORK METRICS

3.1. Persistence

Persistence is a type of dynamic metric used to evaluate the degree of persistent activities over time. Let $\boldsymbol{P_{i,t}}$ denote one instance of a Persistence measure for agent i at time t. Persistence is defined in Equation (1) to be the result of a temporal aggregation function $Agg(\cdot)$ with an input of several activity measures $M_i = (m_{i,1}, \ldots, m_{i,t})$. We will go through the activity measures and temporal aggregation functions later

$$P_{it} = Agg(m_{i,1}, \ldots, m_{i,t}). \tag{1}$$

3.2. Emergence

Emergence is a type of metric used to evaluate the changes in activity over time. We use $\boldsymbol{E_{i,t}}$ to denote the Emergence for agent i at time t. In Equation (2), the Emergence is defined in two different cases. At the very beginning timestep where $t = 1$ or at any timestep when both $P_{i,t}$ and $P_{i,t-1}$ are zero, Emergence is equal to 0, which indicates a neutral change. When t is larger than 1 and $\max\{P_{i,t}, P_{i,t-1}\} \neq 0$, it is defined to be the normalized change of Persistence at the most recent timesteps. Depending on whether the Persistence contains negative values, normalization terms are defined differently. For Persistence that is guaranteed to be positive, $N_{i,t} = \max\{P_{i,t}, P_{i,t-1}\}$. If the positive restriction does not hold, the normalization factor $N_{i,t} = |P_{i,t}| + |P_{i,t-1}|$, which is a weaker bound of the nominator

$$E_{i,t} = \begin{cases} 0 & t = 1 \text{ } or \max\left\{P_{i,t}, P_{i,t-1}\right\} = 0 \\ \dfrac{P_{i,t} - P_{i,t-1}}{N_{i,t}} & t > 1 \end{cases}, \tag{2}$$

where $N_{i,t} = \max\{P_{i,t}, P_{i,t-1}\}$ for Persistence that contains only positive values and $N_{i,t} = |P_{i,t}| + |P_{i,t-1}|$ when Persistence might contain negative values.

To see why the normalization factors make sense, consider Equation (3) as the definition of $E^{+}_{i,t}$ which is equivalent to Equation (2) under the assumption that Persistence is guaranteed to be positive. Here we separated the conditions into three parts and $N_{i,t} = \max\{P_{i,t}, P_{i,t-1}\}$ is substituted for its corresponding value under each condition. Emergence is positive when $P_{i,t} > P_{i,t-1}$ and negative when $P_{i,t} < P_{i,t-1}$. It also follows naturally that if $P_{i,t}$ and $P_{i,t-1}$ are the same or $t = 0$, it yields an Emergence value of 0. Because of the normalization, Emergence always ranges from -1 to 1. When Emergence is close to 1, it suggests a radical increase in Persistence over time. This can be the result of a sudden increase in activity measures from 0 to an arbitrary amount. When the Emergence is close to -1, it suggests that the agent has a radical decrease in its Persistence. If an agent remains at a stable amount of activity overtime, it will have an Emergence of approximately 0, which indicates a neutral emergent pattern,

$$E^{+}_{i,t} = \begin{cases} 1 - \dfrac{P_{i,t-1}}{P_{i,t}} & P_{i,t} > P_{i,t-1} \\ 0 & t = 1 \text{ } or \text{ } P_{i,t} = P_{i,t-1}. \\ \dfrac{P_{i,t}}{P_{i,t-1}} - 1 & P_{i,t} < P_{i,t-1} \end{cases} \tag{3}$$

When Persistence may contain negative values, we have a slight different definition of Emergence called $E^{\pm}_{i,t}$ in Equation (4). Similar to what we have discussed in the positive case, the Emergence is positive when $P_{i,t} > P_{i,t-1}$ and vice versa. It is also obvious to notice that $|P_{i,t} - P_{i,t-1}| \leq |P_{i,t}| + |P_{i,t-1}|$ which makes $-1 \leq E_{i,t} \leq 1$,

$$E^{\pm}_{i,t} = \begin{cases} \dfrac{P_{i,t} - P_{i,t-1}}{|P_{i,t}| + |P_{i,t-1}|} & P_{i,t} > P_{i,t-1} \\ 0 & t = 1 \, or \text{ } P_{i,t} = P_{i,t-1}. \\ \dfrac{-(P_{i,t-1} - P_{i,t})}{|P_{i,t}| + |P_{i,t-1}|} & P_{i,t} < P_{i,t-1} \end{cases} \tag{4}$$

To summarize, Equations (3) and (4) are definitions of Emergence under different assumptions of the data. However, those two definitions can be combined into one using Equation (2), which is the formation definition of Emergence.

One thing which makes Emergence different from Persistence is that there is no magnitude information because of the normalization. We believed that normalization makes Emergence more useful in illustrating the agents' temporal patterns overtime. Consider two agents i and j who have Persistence $P_{i,t} = \beta \cdot a$ and $P_{j,t} = \gamma \cdot a$ at time t and $P_{i,t-1} = \beta \cdot b$ and $P_{j,t-1} = \gamma \cdot b$ on time $t - 1$. When β and γ are significantly different (e.g. β is significantly larger than γ), those two agents exhibit fundamentally different behaviors in terms of Persistence because of the differences in the magnitudes of activities. If we assume that those Persistence are positive and a>b, applying Equation (3) makes the magnitude factor β and γ to disappear. In that case, $E_{i,t} = E_{j,t} = 1 - \frac{b}{a}$. In other words, agent i and j have the same Emergence in spite of their difference in magnitude demonstrated in Persistence.

3.3. Activity Measures

At each network snapshot t, the activity measure $m_{i,t}$ represents the evaluation of activity of node i. Common activity measures include centrality measures, clustering coefficients, and eigenvector-based measures [Bonacich 1972; Watts and Strogatz 1998]. Contrary to Persistence and Emergence, which are dynamic network metrics, the activity measures can only evaluate the activities within a specific timestep. In some situations, they are referred to as *activity metrics*.

Our Persistence and Emergence metrics can be applied to a wide variety of activity metrics. In fact, there are no restrictions on the types of activity metrics that our method can be utilized in conjunction with. If the activity metrics are guaranteed to be non-negative, the corresponding Persistence will also be non-negative. Some examples include degree centrality and betweenness centrality. When calculating Emergence, we can choose the normalization factor as $\max\{P_{i,t}, P_{i,t-1}\}$ to achieve a tighter bound. If the activity metrics are not guaranteed to be nonnegative, the corresponding Persistence may contain negative values. In this case, we have to choose the more relaxed normalization factor $|P_{i,t}| + |P_{i,t-1}|$. Such activity metrics include eigenvector centrality. In some cases, one can use linear transformation techniques (e.g. to divide by its minimal values) to convert those metrics into a nonnegative measures [Wei et al. 2011].

4. TEMPORAL AGGREGATION MODELS

Temporal aggregation models are essential components of the dynamic metrics we defined in Section 3. In this section, we will illustrate three temporal aggregation models and illustrate how they relate to the dynamic metrics.

4.1. Aggregation Functions

The aggregation function agg($\cdot$) takes a variable length vector parameter $\boldsymbol{m}$ as input and returns a single numeric evaluation of the temporal information contained in $\boldsymbol{m}$. The temporal activity of a specific agent i from time 1 to time t is denoted as a vector $\boldsymbol{m_i} = (m_{i,1}, \ldots, m_{i,t})$. We then apply the aggregation function agg($\cdot$) on this vector to assess Persistence and Emergence. Given the fact that the length of temporal history varies in the networks, the aggregation function should be adaptable to variable input length.

For an input vector $\boldsymbol{m} \in \mathbb{R}^d$, we define the aggregation function to be the linear combination of input variables, as is defined in Equation (5)

$$\mathrm{Agg}\,(\mathbf{m}) = \alpha_1 \mathrm{m}_1 + \alpha_2 \mathrm{m}_2 + \cdots + \alpha_{\mathrm{d}-1} \mathrm{m}_{\mathrm{d}-1} + \alpha_{\mathrm{d}} \mathrm{m}_{\mathrm{d}} = \boldsymbol{\alpha} \cdot \mathbf{m} \tag{5}$$

with regularization condition $\sum \alpha_{\mathrm{i}} = 1$.

From Equation (5), we can see that the aggregation function takes all the values in the activity vector into account. A regularized vector $\boldsymbol{\alpha}$, which we call the **strength vector**, is used to tune the activity vector $\boldsymbol{m}$ so that only the information in the desired timesteps are counted toward the value of the aggregation function. The sum of the strength vector over the time range of the activity measure must be equal to 1. This is referred to as the regularization condition of the strength vector. In some cases where the dimension of **m** is very large, a good temporal aggregation model will assign large α_{i} to timesteps which are of the most interest to the analysis of the temporal assessments and assign small α_{i} to the timesteps that are less important. The temporal aggregation models that we will discuss later define mechanisms to generate the values of strength vector α according to the length of the activity measure. Note that the values of strength vector $\boldsymbol{\alpha}$ only depend on the temporal model and the dimension of the input vector d. Once the temporal model and the dimension of the input vector are determined, the strength vector will remain fixed regardless of what the exact values of the input vector **m** are.

A good temporal aggregation model should satisfy the following requirements:

(1) The model should come up with a scalable algorithm to dynamically calculate the result of an aggregation function based on input vector **m**. For applications such as abnormality detection in large datasets, it might require the algorithm to respond in real time.
(2) The aggregation models should be temporally informative and meaningful to Persistence and Emergence metrics. They should achieve good generalizations of temporal trends and provide predictive insight.
(3) A tunable model parameter should be provided to switch between Recency and Primacy effect. When the model parameter favors Recency effect, α_{i} on the most recent timesteps should be large. On the other hand, when the model parameter favors the Primacy effect, α_{i} on the distant past should be increased.

4.2. Average Aggregation Model

The main idea of the average model is to aggregate historical temporal information by simply averaging over it. This simple model is the most widely used temporal aggregation model. We use a window parameter w so that the model will only aggregate temporal information in the latest w timesteps. We define the aggregation function on Average Aggregation Model with window parameter w to be $Agg^{A(w)}(\boldsymbol{m})$, which is defined in Equation (6)

$$\mathrm{Agg}^{\mathrm{A(w)}}(\mathbf{m} \in \mathbb{R}^{\mathrm{d}}) = \begin{cases} \dfrac{\mathrm{m}_1}{d} + \cdots + \dfrac{\mathrm{m}_{\mathrm{d}}}{d} = \dfrac{\sum_{i=1}^{d} \mathrm{m}_{\mathrm{i}}}{d} & d < w \\ \dfrac{\mathrm{m}_{\mathrm{d-w+1}}}{w} + \cdots + \dfrac{\mathrm{m}_{\mathrm{d}}}{w} = \dfrac{\sum_{i=\mathrm{d-w+1}}^{d} \mathrm{m}_{\mathrm{i}}}{w} & d \geq w \end{cases}. \tag{6}$$

In Equation (5), the definition of the aggregation function of Average Aggregation Model is divided into two parts: if the length of the input vector d is smaller than the window size, it shrinks the window size to fit to the actual dimension of the data. If otherwise there is too much information, it truncates the early timesteps and considers the information in the most recent timesteps that fall into the window.

One thing worth noticing is that the window is always aligned with the most recent timestep. The most recent information is never lost. If the window size w is set to a small number such as 1, the Average Aggregation Model will degenerate to the model that only considers the most recent timestep, which is an extreme practice of Recency effects. On the other hand, if we increase the window size to infinity, it will consider every single timesteps in the history. This enables the model to represent a stronger Primacy effect as it takes more past information into consideration.

We fit the aggregation function into Equations (1) and (2) to get the Persistence and Emergence for Average Aggregation Model in Equations (7) and (8), respectively:

$$P_{i,t}^{A(w)} = \frac{\sum_{\mathrm{k=t-max\{t,w\}+1}}^{\mathrm{t}} \mathrm{m_{i,k}}}{\mathrm{max\{t,\ w\}}}, \tag{7}$$

$$E_{i,t}^{A(w)} = \begin{cases} 0 & t = 1 \; or \; P_{i,t-1}^{A(w)} = P_{i,t}^{A(w)} \\ \dfrac{P_{i,t}^{A(w)} - P_{i,t-1}^{A(w)}}{N_{i,t}^{A(w)}} & t > 1 \end{cases}, \tag{8}$$

where $N_{i,t}^{A(w)} = \max\{P_{i,t-1}^{A(w)}, P_{i,t}^{A(w)}\}$ if Persistence is non-negative and $N_{i,t}^{A(w)} = |P_{i,t-1}^{A(w)}| + |P_{i,t}^{A(w)}|$ if otherwise.

One of the issues in calculating the metrics is computational complexity. Fortunately, both the Persistence and Emergence in Average Aggregation Model can be calculated recursively based on values at the previous timestep. Equation (9) illustrates a recursive method to calculate Persistence at time t based on Persistence at time $t - 1$ and the activity measure at time t. The basic idea here is that we first get the sum of the previous activity measures by multiplying $P_{i,t-1}^{A(w)}$ with its window size. If $1 < t < w$, the window size is $t - 1$. If $t \geq w$, the window size is w. After we get the sum, depending on how many terms are already in the sum, we might need to subtract the oldest activity measure from the sum. If $t \geq w$, we need to subtract the sum by $m_{i,t-w}$ because this is the element that is discarded as the window moves one step forward. If $1 < t < w$, we do not need to subtract the sum by anything because the new window will not discard any elements. If $t = 1$, then the calculation of Persistence degenerates into the base case. Only $O(1)$ modifications are needed to calculate Persistence at the current timestep based on the result at the previous timestep. The calculation of Emergence can also be done in $O(1)$ as a byproduct of the Persistence calculation which is illustrated in Equation (8),

$$P_{i,t}^{A(w)} = \begin{cases} m_{i,1} & t = 1 \\ \dfrac{P_{i,t-1}^{A(w)} \cdot (t-1) + m_{i,t}}{t} & 1 < t < w \\ \dfrac{P_{i,t-1}^{A(w)} \cdot w - m_{i,t-w} + m_{i,t}}{w} & t \geq w \end{cases}. \tag{9}$$

Algorithm 1 is the pseudocode for calculating Persistence and Emergence recursively in Average Aggregation Model. The algorithm takes four input variables: the current timestep t, Persistence at time $t - 1$, P_{t-1}, activity measure at time t, m_t and the window size w. The algorithm returns Persistence and Emergence at time t. Based on the analysis of recursive calculation, the algorithm scales in $O(1)$ in terms of running time when activity metrics are calculated in advance.

4.3. Linear Aggregation Model

One serious drawback of the Average Aggregation Model is the lack of differences between timesteps within a window. In the Linear Aggregation Model, we assign different strengths to different values of input vector $\boldsymbol{m}$ based on how close it is to the most recent timestep.

In the Linear Aggregation Model, the strengths increase linearly from the distant timesteps to the recent timesteps. For the Linear Aggregation Model with window size

ALGORITHM 1: Calculate Persistence and Emergence on Average Aggregation Model

```
Input: $t$, $P_{t-1}$, $m_t$, $w$
Output: $E_t$, $P_t$
if $t = 1$
    $p_t = m_t$
else
    if $1 < t < w$ then
        $P_t = \frac{P_{t-1} \cdot (t-1) + m_t}{t}$
    else
        $P_t = \frac{P_{t-1} \cdot w - m_{t-w} + m_t}{w}$
    end
end
if $t = 1$ or $P_{t-1} = P_t$ then
    $E_t = 0$
else
    if $P$ non-negative = True then
        $E_t = (P_t - P_{t-1})/\max\{P_t, P_{t-1}\}$
    else
        $E_t = (P_t - P_{t-1})/(|P_t| + |P_{t-1}|)$
    end
end
Return $E_t$, $P_t$
```

w, we can get the strength vector $\boldsymbol{\alpha} \in \mathbb{R}^w = (\mu, 2\mu \ldots, (w-1)\mu, w\mu)$. Using the regularization condition in definition (5), we can solve $\mu = \frac{2}{(1+w)w}$. Substituting it back to the strength vector we find that $\alpha = (1, 2, \ldots, w-1, w) \cdot \frac{2}{(1+w)w}$. Using this strength vector, we are left with Equation (10), which is the formal definition of the aggregation function of the Linear Aggregation Model. Similar to the aggregation function of the Average Aggregation Model, we divide the definition into two cases: If the length of input vector d is less than the window size, then the window is shrunk into the actual dimension of the data. Otherwise, the aggregation function will consider the input information that falls into the window. The window is always aligned with the current timestep and anything outside the window will be discarded. The model will represent more Recency effects when the window size is small. When the window size is large, the model will put more weight on historical information and will better represent more Primacy effects,

$$\mathrm{Agg}^{\mathrm{L(w)}}(\mathbf{m} \in \mathbb{R}^{\mathrm{d}}) = \begin{cases} \dfrac{2}{(1+\mathrm{d})\mathrm{d}} \displaystyle\sum_{i=1}^{d} i \cdot m_i & d < w \\ \dfrac{2}{(1+\mathrm{w})\mathrm{w}} \displaystyle\sum_{i=d-w+1}^{d} i \cdot m_i & d \geq w \end{cases}. \tag{10}$$

Equations (11) and (12) are the definitions of Persistence and Emergence of Linear Aggregation Model by substituting the aggregation function. Unfortunately, there is no recursive method to calculate the dynamic metrics in the Linear Aggregation Model. The calculation of both Persistence and Emergence must be made directly from the definition,

$$P_{i,t}^{L(w)} = \frac{2}{(1+\max\{\mathrm{d},\mathrm{t}\})\max\{\mathrm{d},\mathrm{t}\}} \sum_{i=t-\max\{d,w\}+1}^{t} i \cdot m_i, \tag{11}$$

$$E_{i,t}^{L(w)} = \begin{cases} 0 & t = 1\, or\ P_{i,t-1}^{L(w)} = P_{i,t}^{L(w)} \\ \dfrac{P_{i,t}^{L(w)} - P_{i,t-1}^{L(w)}}{N_{i,t}^{L(w)}} & t > 1 \end{cases}, \tag{12}$$

where $N_{i,t}^{L(w)} = \max\{P_{i,t-1}^{L(w)}, P_{i,t}^{L(w)}\}$ if Persistence is non-negative and $N_{i,t}^{L(w)} = |P_{i,t-1}^{L(w)}| + |P_{i,t}^{L(w)}|$ if otherwise.

Algorithm 2 is the pseudocode to calculate the dynamic metrics on Linear Aggregation Model. The computational complexity of this algorithm is $O(w)$, which is proportional to the window size.

ALGORITHM 2: Calculate Persistence and Emergence on Linear Aggregation Model

```
Input: $t, m = (m_1, \ldots \boldsymbol{m}_d), w$
Output: $E_t, P_t$
$P_t = 0$
For $i = t - max\{d, w\} + 1$ to $t$ do
    $P_t = P_t + i \cdot m_i \cdot \frac{2}{(1 + \max\{\mathrm{d, t}\})\max\{\mathrm{d, t}\}}$
end
if $t = 1$ or $P_{t-1} = P_t$ then
    $E_t = 0$
else
    if $P$ .nonnegative = True then
        $E_t = (P_t - P_{t-1})/\max\{P_t, P_{t-1}\}$
    else
        $E_t = (P_t - P_{t-1})/(|P_t| + |P_{t-1}|)$
    end
end
Return $E_t, P_t$
```

4.4. Exponential Aggregation Model

The Exponential Aggregation Model assigns exponentially increasing weights to the input vector. Similar to the Linear Aggregation Model, the information at the timesteps that are closer to the current timestep will be weighted more strongly and thus have larger values in the strength vector. Instead of increasing the strength vector linearly, the Exponential Aggregation Model has an exponentially increased strength vector. Because of this, the importance of temporal information increases rapidly overtime.

The aggregation function of Exponential Aggregation Model is defined in Equation (13). There is a parameter β in the model that ranges from 0 to 1 inclusive, which we call the *transmission parameter*. The transmission parameter controls the main exponential component of the aggregation function

$$\mathrm{Agg}^{\mathrm{E}(\beta\in[0,1])}(\mathbf{m} \in \mathbb{R}^{\mathrm{d}}) = \beta^{d-1} m_1 + \sum_{j=2}^{t} (1-\beta)\beta^{d-j} m_j. \tag{13}$$

Equation (14) is the corresponding strength vector of the aggregation function. One can easily validate that the sum of the strength vector is always equal to 1, which meets the regularization condition in Equation (5). The strength on the first timestep is slightly different from the consequent ones. However, all the values in the strength vector are dominated by β^{d-j}, which is very large at recent timesteps but decays exponentially at earlier timesteps when β ranges from 0 to 1. If we change the transmission parameter to 0, the aggregation function will only consider the information in the most recent timestep, that is $\alpha^{E(\beta)}(j = 1) = 1, \alpha^{E(\beta)}(j \neq 1) = 0$. This means no history information is transmitted into the dynamic metrics, and is an extreme practice of the Recency effect. However, if we increase the transmission parameter, we see that some of the strength of the first timestep is moved to other timesteps. This represents a blend of the Recency and Primacy effects. If the transmission parameter is set to 1, then only the information in the first time period will have impacts to the dynamic metrics, while all other timesteps will have a strength vector of 0. This represents a pure Primacy effect where only the first information matters,

$$\alpha^{\mathrm{E}(\beta)}(\mathrm{j}) = \begin{cases} \beta^{d-1} & if\ j = 1 \\ (1-\beta)\beta^{d-j} & otherwise \end{cases}. \tag{14}$$

Equation (15) is the definition of Persistence under the Exponential Aggregation Model. The Persistence on the first timestep is just the activity measure itself. On the subsequent timesteps, it is defined recursively based on Persistence at time $t-1$ and the activity measure at time t. The transmission parameter β plays an important role in controlling the exponential components. One can easily validate that the Persistence defined in Equation (15) has the same aggregation function defined in Equation (13) as well as the same strength vector defined in Equation (14),

$$P_{i,t}^{E(w)} = \begin{cases} m_{i,t} & if\ t = 1 \\ \beta P_{i,t-1}^{E(w)} + (1-\beta)m_{i,t} & otherwise \end{cases}. \tag{15}$$

Equation (16) is the Emergence of Exponential Aggregation Model. Similar to the Emergence definitions in other aggregation models, the Emergence is defined based on the Persistence on t and $t-1$,

$$E_{i,t}^{E(w)} = \begin{cases} 0 & t = 1\ or\ P_{i,t-1}^{E(w)} = P_{i,t}^{E(w)} \\ \dfrac{P_{i,t}^{E(w)} - P_{i,t-1}^{E(w)}}{N_{i,t}^{E(w)}} & t > 1 \end{cases}, \tag{16}$$

where $N_{i,t}^{E(w)} = \max\{P_{i,t-1}^{E(w)}, P_{i,t}^{E(w)}\}$ if Persistence is nonnegative and $N_{i,t}^{E(w)} = |P_{i,t-1}^{E(w)}| + |P_{i,t}^{E(w)}|$ if otherwise.

Algorithm 3 is the pseudocode to calculate Persistence and Emergence on Exponential Aggregation Model. The Persistence and Emergence in Exponential Aggregation Model can be calculated recursively. The computational complexity is $O(1)$.

One thing worth mentioning is the possibility of converting the transmission parameter to an equivalent window parameter in the Exponential Aggregation Model. In our experiment section, we will compare the results across different aggregation models. For the average and liner model, it is not difficult to compare them because both of them work on the same parameter, which is the window parameter. For the exponential model that uses the transmission parameter, however, it is hard to compare it against other models under the same scale. One way to solve this problem is to use an equivalent variable that can be calculated from transmission parameter to represent the window size parameter. In Equation (17), we define the *Window Parameter* of the transmission parameter β, $\lambda(\beta, T)$ to be the shortest distance from a specific timestep to the current timestep T so that the sum of the strength vector from the current timestep to this specific timestep is larger than 0.99. Since the sum of all

ALGORITHM 3: Calculate Persistence and Emergence on Exponential Aggregation Model

```
Input: $t, m_t, \beta, P_{t-1}$
Output: $E_t, P_t$
If $t = 1$ then
    $P_t = m_t$
Else
    $P_t = \beta P_{t-1} + (1 - \beta) m_t$
End
if $t = 1$ or $P_{t-1} = P_t$ then
    $E_t = 0$
else
    if $P$ .nonnegative = True then
        $E_t = (P_t - P_{t-1})/\max\{P_t, P_{t-1}\}$
    else
        $E_t = (P_t - P_{t-1})/(|P_t| + |P_{t-1}|)$
    end
end
Return $E_t, P_t$
```

the elements in the strength vector is always 1, this ensures that the majority of the timesteps that will have impacts to the aggregation function will be selected into the window parameter,

$$\lambda(\beta, \mathrm{T}) = \left\{ k \mid \sum_{j=T-k}^{T} \alpha^{\mathrm{E}(\beta)}(\mathrm{j}) \geq 0.99 \geq \sum_{j=T-k+1}^{T} \alpha^{\mathrm{E}(\beta)}(\mathrm{j}) \right\}. \tag{17}$$

4.5. Relationship to Persistence/Emergence and Activity Measures

Before continuing, we here review the relationships between the aggregation models, Persistence/Emergence, and activity measures. Persistence and Emergence are high-level ideas that we construct to measure the temporal patterns of dynamic social networks. These two metrics are built on a specific aggregation model. We have proposed three aggregation models. Each of them can be used in the Persistence/Emergence evaluations. These three aggregation models represent three different ways of handling temporal information. One can construct his/her own aggregation models and apply Persistence/Emergence onto it.

The aggregation models and activity measures carry different information that is important to the evaluation of temporal patterns. Centrality metrics such as betweenness and degree are such metrics that can represent structural information of a network. The Aggregation models are meant to carry the temporal information of activity measures and aggregate them in a meaningful way. The Persistence and Emergence metrics are used to reorganize the temporal and structural information generated by aggregation model and activity measure respectively and to generate only the most important two pieces of information that are relevant to the problem of assessing temporal patterns.

5. DATASET

We use four different datasets to validate the dynamic metrics: the Newcomb fraternity dataset, the Enron email dataset, the DBLP (i.e. DBLP computer science bibliography hosted at http://dblp.uni-trier.de) co-authorship dataset, and the "Arab Spring" twitter dataset. Each dataset has several snapshots and a set of agents who have a series of network activities overtime. In the Enron, DBLP and Arab Spring dataset, there are a significant amount of agents who have very little activities overtime. We applied a data

Table I.

Dataset	Nodes	Snapshots
Newcomb	17	15
Enron	946	34
DBLP	1981	35
Arab Spring	4825	18

processing procedure to eliminate those agents in order to provide enough training data to the regression model on each timestep. Table I is a summary of these datasets after the preprocessing procedures.

The Newcomb fraternity dataset was collected by Theodore Newcomb [Newcomb 1961] at the University of Michigan. The dataset recorded the changes of preference of 17 transfer students living in the same apartment who did not know each other when they moved in. The strengths of the links in the network represent the preferences between two students. In the original data, 1 represents highest preference (i.e., this is the actor's "best friend") while 16 represents the lowest preference. We processed the data so that high preference is consistent with high link strength by subtracting 16 by the original link values in the network. After that, 15 will represent highest friendship relationship while 0 will represent lowest friendship. The dataset consists of 16 full snapshots of network with each representing a survey conducted at the end of a specific week.

The Enron email dataset [Klimt and Yang 2004] contains a set of emails released by Enron Corporation. The dataset contains users that were mostly senior managers in the company. We combined all emails that belong to the inbox in the dataset and build a social network between senders and receivers with each snapshot representing a single month. We filter out users who contributed less than 1% of the total emails to network in any snapshot. We also deleted the first seven snapshots of the data in which we see little network activity. The final dataset contains 946 users with 34 snapshots.

The DBLP dataset [Zaiane et al. 2007] is built from co-authorship relationships in DBLP database, which is a computer science bibliography website hosted at Universität Trier, in Germany. We extracted co-authorship information in each paper and built the network within a year. If multiple authors were present in the same paper, a pairwise relationship was recorded as a link in the network. Multiple links were aggregated to be a single link, the strength of which represents the number of co-authorships. We dropped authors who had less than 25 co-authors in any snapshots and deleted the first 30 years of the dataset. After these preprocessing steps, the dataset contained 1981 distinct authors and 49 different snapshots.

The Arab Spring dataset is collected using the Twitter Application Program Interface (API) from December 2011 to March 2013. The twitter API returns results that contain 10% of all tweets sent out at the time we collect the data. We extracted tweets that have a latitude/longitude attribute that fell into the Arab Spring countries and conducted an Latent Dirichlet Allocation (LDA) topic modeling based on all the tweets sent by a user. We then build the user-to-user network by calculating the similarities between two users based on their topic score vectors. We bucket the network by month and generated 16 user-to-user networks.

6. EXPERIMENTAL RESULTS

6.1. Experimental Setup

To compare the performance of dynamic metrics across different models, we need a criterion to assess whether a specific model is better than another. Since the metrics measure the temporal activities in the past, it will have statistical correlations to future activities of an agent. Our criterion to assess temporal aggregation models is thus based on how strong those correlations are.

The general idea of the experiment is to first calculate the dynamic metrics and then use a regression model to find statistical correlations between the dynamic metrics at time t and the true activity metrics at time $t + 1$. We use lasso regression, which is a linear regression model to train the regression model on dynamic metrics and use it to predict future activity measures. We use a linear regression model instead of more complicated models for the following reasons: (1) a simpler and less powerful regression model leaves the complexity of the prediction task to the dynamic metrics, which is the main focus of our article. If the regression model is simple, we can easily distinguish between different dynamic metrics. However, if we choose a complicated regression model, chances are that those dynamic metrics will have virtually little performance differences because the regression model is powerful enough to generate a model that fits the input/output features. (2) For the dynamic metrics, we want them to meet the human expectation of Persistence and Emergence so that when either of those two metrics changes, people will immediately realize what happened to the network. If we use a more complicated model, it will likely to pick up an aggregation model that requires a less intuitive way to interpret Persistence and Emergence, which makes the metrics less useful in detecting network changes. (3) The lasso regression model is implemented in most statistical software packages. This increases the repeatability of our analysis presented in the article. We want to emphasize that the regression models in the experiments are used for the purpose of comparing different metrics only. They are only used to illustrate the statistical correlations between the metrics and the activity measure and should not be considered to be part of the metrics.

To begin with, we first generate activity metrics on each timestep for each agent in the network. We calculate Persistence and Emergence based on the algorithms we proposed in Section 4. Each aggregation model will have a different way of calculating those metrics. We recorded the running time in the process of calculating these dynamic metrics. After that, we start the training process of the regression models. We build two versions of the experiments. One uses only Persistence and the other uses both Persistence and Emergence. We complete the same experiments on four different datasets illustrated in Section 5. Figure 1 illustrates the general procedure of the experiments. We will not build a regression model using only Emergence because Emergence alone can only indicate the trend and not the magnitude of the activities. Without the magnitude of the activity, the performance of the regression model will be compromised. We implemented lasso regression using the R glmnet package.[Friedman et al. 2010]. The lambda parameter is set to 1e−5.

To illustrate the feature variables and target variables of the regression, consider Figure 2, where data in a dynamic network are organized into a matrix with horizontal axis that represents time and vertical axis which represents the agents in the network. We choose a column in the matrix to be the feature variable of the regression model and choose multiple columns that is one timestep forward to the feature variable to be target variables. The feature variables are chosen to be the dynamic metrics while the target variables are the activity measures. The regression analysis is conducted to find statistical correlations between feature variables and target variables. In Figure 2, green cells are feature variables and blue cells are target variables. In that situation, we are trying to predict information that is two timesteps away using the dynamic metrics. For each experiment, we slide the selection of feature variable throughout the time frame of the dataset to obtain results of different regression experiments under similar parameter settings.

The parameter space varies depending on the aggregation model. For the Average Aggregation Model and Linear Aggregation Model, the window parameter takes value from 1 to approximately the same size as the maximum timestep of the data. For Exponential Aggregation Model, however, we choose transmission parameter to vary

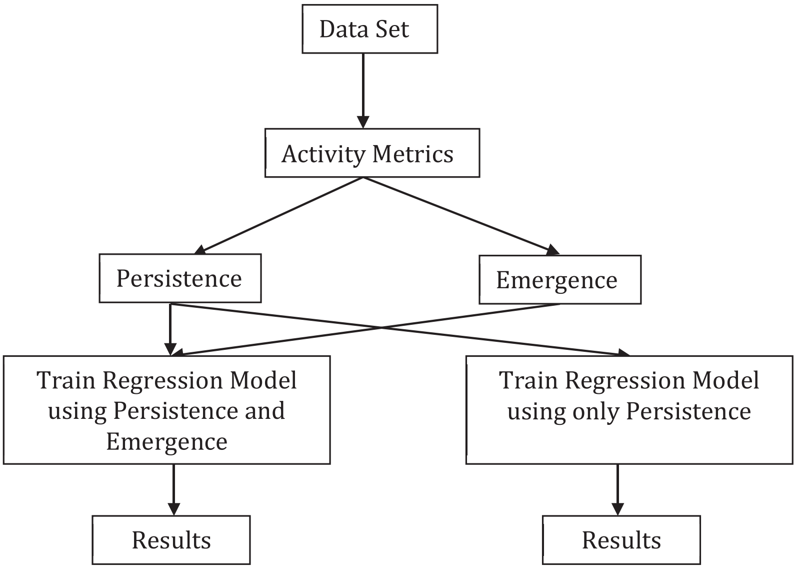
Data Set
Activity Metrics
Persistence
Emergence
Train Regression Model using Persistence and Emergence
Train Regression Model using only Persistence
Results
Results

Fig. 1. Illustration of the experiment procedure.

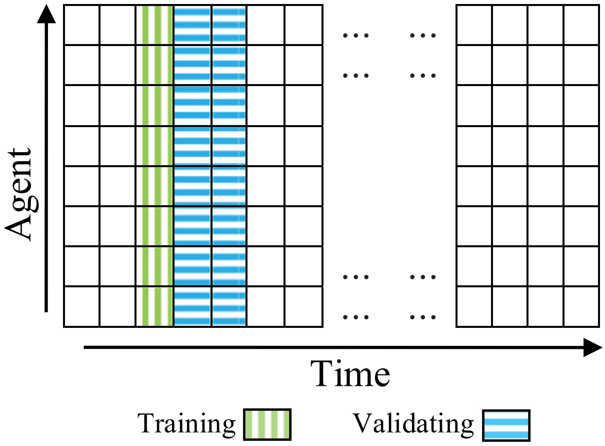

Fig. 2. Illustration of each set of experiments.

evenly: 0.01, 0.02,..., 0.99, 1.00. In some cases, where we want to compare the models under the same parameterization, we use the equivalent window parameter defined in Section 4.4 to replace the transmission parameter in Exponential Aggregation Model. We will also fix the length of the feature variable to be 1 while changing the length of the target variable to see how many future timesteps can we predict based on the dynamic metrics. We call the length of the target variable *validation size*.

6.2. Activity Measures

We choose four different activity measures in our experiment: degree centrality, closeness centrality, betweenness centrality, and clustering coefficient. Each of these metrics will evaluate different aspects of the network. Degree centrality measures the number of connections of each agent in the network. Closeness centrality measures the average

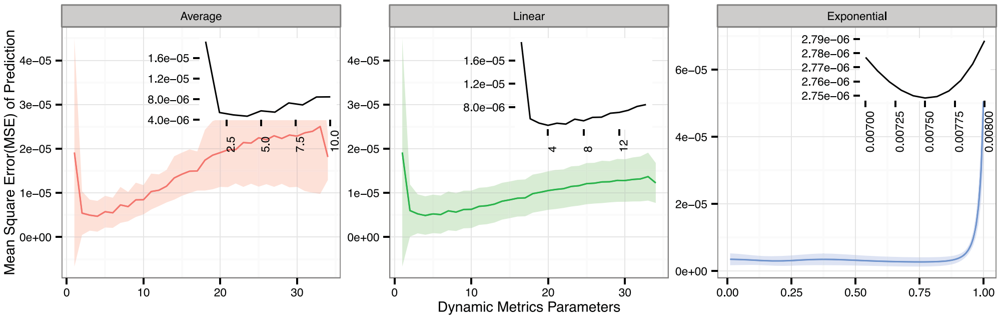

Fig. 3. Comparison of aggregation model performance in DBLP dataset when validation size = 1.

efforts to reach all other agents in the network through shortest paths. Betweenness centrality measures the importance of an agent by counting what fraction of shortest paths goes through this particular agent. Finally, clustering coefficient measures to what extent agents in a network tend to cluster with each other.

6.3. Measuring the Temporal Patterns to Predict Future Activities

To see the performance of dynamic metrics on predicting future activity metrics, first consider the results on the DBLP dataset illustrated in Figure 3. Here, we use degree centrality as activity measure and have both Persistence and Emergence enabled in the regression model to predict activity measures in the next timestep (i.e. validation size = 1). Each plot represents a different temporal aggregation model. The horizontal axis is the parameter of each model. For Average and Linear model, the parameter is a window size ranges from 1 to the temporal size to the dataset. For the Exponential model, the parameter is the transmission parameter ranging from 0.01 to 1. The vertical axis is the means square error (MSE) of prediction of the regression model. Each point in the graph represents the mean of a set of experiments, which have the same aggregation model and model parameter but with different start point and end point of the feature and target variable. The shaded area is the confidence interval of MSEs of different regression analysis.

Figure 3 shows a non-linear pattern of prediction error. Despite fluctuations, all of the models have a higher prediction error when parameters become large. This is because when the parameters are large, the aggregation functions assign more weights to the historical data and the result becomes less relevant to the future activity measures. By changing the parameters, the prediction accuracies change significantly. The sub plots in each of the main plot are a close-up view of the region where the minimal MSE occurs. We see that there exists a specific parameterization that results in a minimal MSE of the prediction in each aggregation model. We have observed this phenomenon in all of the experiments that we conducted. This may suggest that there exist an optimal parameter that we can choose to minimize the prediction error for each aggregation model. The optimal parameter represents an amount of history activity measures that are of most relevance to the task of measuring temporal patterns. Using too much history information will be a detriment to the sensitivity of the analysis, while too little information will make the metrics insufficient to represent the true dynamic pattern. We can also see that the Exponential model has a much smoother curve of MSE over parameters than that of Average and Linear models. This is because the window parameters can change only in the range of integers while the transmission parameter can change in real values. This will lead to the differences in the expressiveness of Recency and Primacy effects, which will affect the sensitivity of the analysis.

Table II. Result of the Lasso Regression Model

Coefficient Persistence	Coefficient Emergence	Intercept
9.98e−01	1.11e+00	1.26e−05

Table III. Visual Comparison of Predicted and Actual Activity Measures

Agent Num.	Persistence	Emergence	Predicted Act.	Actual Act.	Diff
1	0.5695	−0.1455	0.4065	0.4049	0.0016
2	1.0000	0.0000	0.9982	1.0000	0.0018
3	0.6538	0.1056	0.7702	0.7599	0.0103
4	0.2963	0.0510	0.3526	0.3679	0.0153
5	0.1297	0.0177	0.1492	0.1613	0.0121
6	0.0644	0.0042	0.0690	0.0777	0.0087
7	0.0329	0.0102	0.0442	0.0397	0.0046
8	0.0187	0.0048	0.0240	0.0193	0.0047
9	0.0117	−0.0008	0.0108	0.0105	0.0003
10	0.0063	−0.0019	0.0042	0.0067	0.0025

Table II illustrates the results of the lasso regression. Since lasso is a linear regression model, the result of the fitting on a two-dimensional input will generate three estimations. For this specific experiment, we see that both Persistence and Emergence have a fairly large coefficient while the intercept is relatively small. Emergence has slightly higher coefficients than Persistence.

In Table III, we illustrate the prediction results of the first 10 agents in the network in one of the experiments. In this specific experiment, we choose the exponential aggregation model and set transmission parameter to be 0.75. We calculated both Persistence and Emergence on the 15th timestep and ran the regression analysis to predict the activity measure on the 16th timestep. We can see that the dynamic metrics does a good job of generalizing the temporal patterns and even a linear regression model can achieve a good accuracy in the prediction task. We can see that the predicted activity measures and actual activity measures are fairly close and the differences are quite small for all of the agents. Similar experimental results are observed on all other datasets and timesteps.

The analysis in Figure 3 suggests an interesting way to look at the experimental results, which is to compare the performance of the models when parameters minimize the prediction error. Figure 4 and Figure 5 present the analysis using this technique. Since we have four different datasets and each of them can have four different activity measures, we have 16 sets of different sets of Persistence and Emergence measures. For each measure set, we conduct experiments using both Persistence and Emergence (marked with "∗") and only Persistence (marked without "∗"), which makes the total number of experiments to be 32. In each plot, the vertical axis represents the MSE of prediction and the horizontal axis represents the validating size of the experiments. For each validating size, we conducted many sets of regression experiments by varying the model parameters. The one we report in the graph is the one that minimizes the mean of MSE. Different colors and shapes of the points represent different aggregation models.

From Figure 4, we can see that the Exponential model generally achieves the best prediction accuracy, followed by Linear and Average model. There are two reasons which lead to this. First, the Exponential model is more expressive in representing the Primacy and Recency effects than the other two models. The exponential model uses a real-valued parameter to control the level of Primacy and Recency effects, while other models can only use a discrete window-size parameter. The real-valued parameter gives the dynamic metrics better chance to capture the optimal values for regression

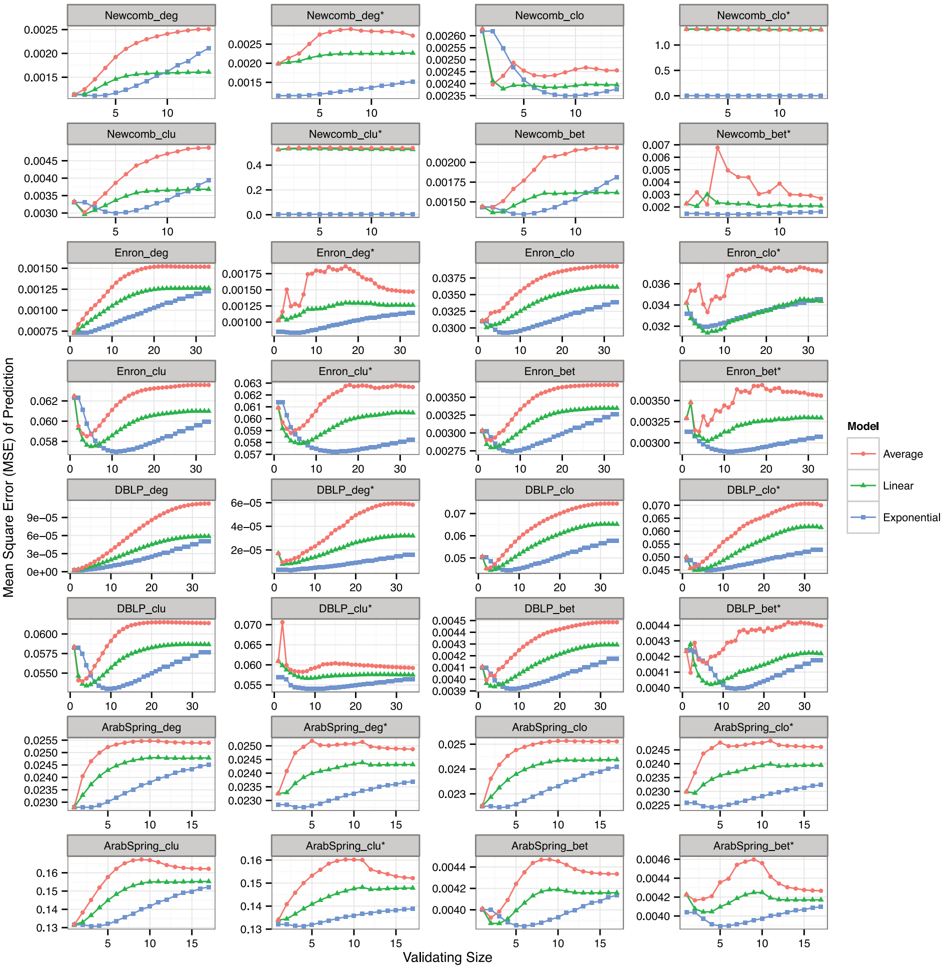

Fig. 4. Comparison of prediction performance across validation size.

analysis. Second, the Exponential model features a more reasonable strength vector than that of Linear and Average model. The Exponential component makes the aggregation function focus on the most recent timesteps, which are important to the task of prediction. Although the Linear model also has a strength vector that increases with time, it increases too gently to capture the actual evolutionary pattern of the agents. The Average model, which did not distinguish between the recent and past information at all, has the worst prediction accuracy and the worst ability to measure temporal patterns.

One important observation is that the validation size plays an important role in the prediction task. With small validation sizes, models generally achieve small errors since those activities to predict is not far away from the point when dynamic metrics are generated. With the increase of the validating size, all the models demonstrated certain degree of decrease in the prediction accuracies. This is because there are certain temporal variables that are not incorporated into the network information we have. The regression model achieves better performance in timesteps that are closer to the

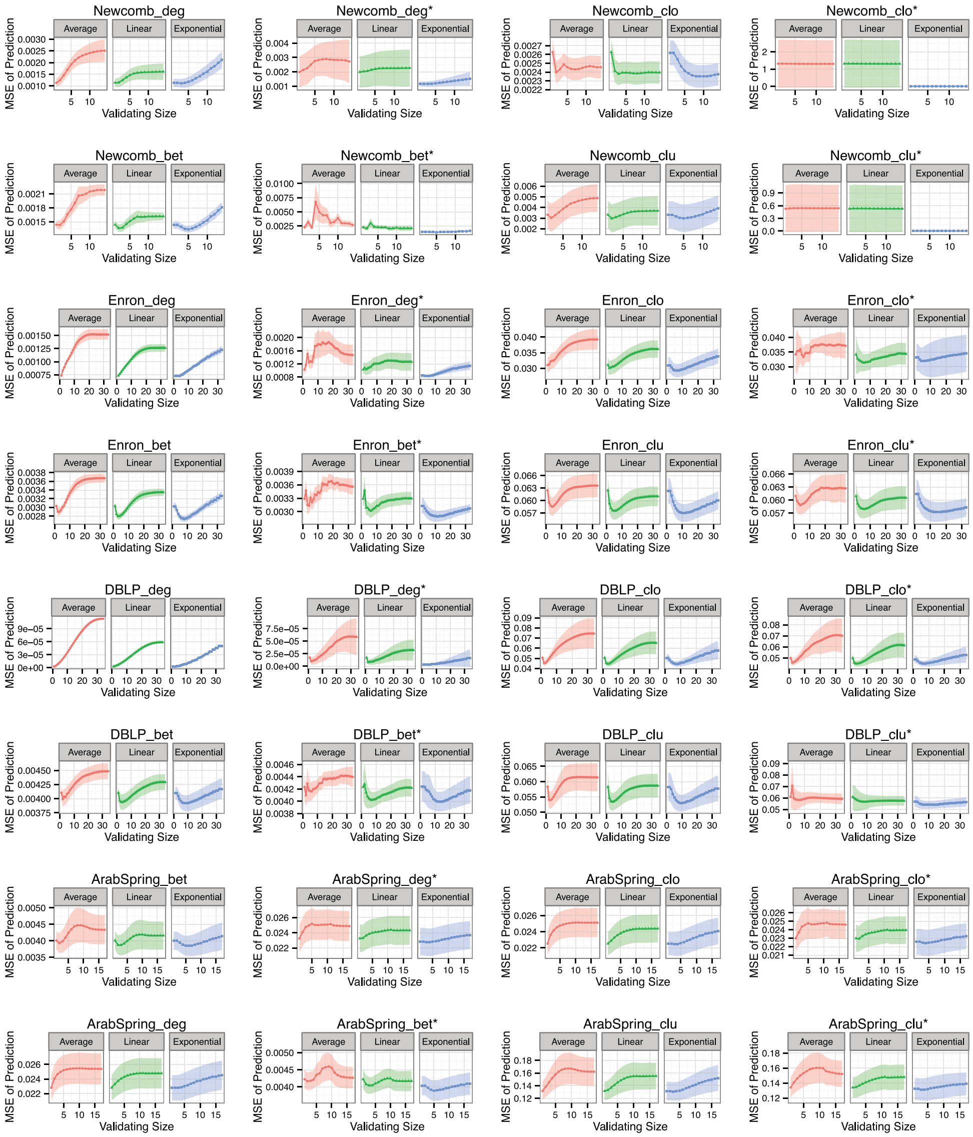

Fig. 5. Comparison of prediction performance across validating size with confidence interval.

feature variable. The larger the validating size is, the worse the performance of the predictions will be.

Another interesting observation is that the error first starts to decrease and then increase with the increase of the validation size. This indicates that the models can do a better job of predicting activity metrics in some distant future than those in the near future. Datasets that have this phenomenon frequently observed such as the Enron dataset and the Arab Spring dataset have complex temporal dependencies in network activity. In those datasets, change in activity measures will see a delay in its effects to the future activities. In other words, activities that are this distance away in the future have the most temporal correlation to the dynamic metrics and thus achieved the best prediction accuracy. We use the term *best validating distance* to indicate the validating size that achieved the best prediction error. For those datasets that have best validating distance equals to 1, dynamic metrics have an immediate impact to the activity measures in the next timestep.

Figure 5 is made using exactly the same piece of information as that in Figure 4 but with each aggregation models plotted separately. That makes it possible to show the confidence intervals of each aggregation models under different dataset/activity measure/experimental settings. We observe that the Exponential model has both a lower prediction error and a tighter confidence interval than other models, which again validated that exponential model performs better than the other two. In addition, we see that the combined use of Persistence and Emergence makes the confidence intervals tighter by comparing experiments with and without "$*$". The two dynamic metrics represent two different aspects of the temporal evolution patterns. Using those two dynamic metrics together achieves a better generalization of the historical information and hence a better prediction of future activities.

Please note that the results presented in Figure 4 and Figure 5 is asynchronous, in that each point in the graph represents different experiments using a different parameter. This is because we report only the results of the model that minimize the prediction error for each validating size. The parameter to reach the optimal prediction error varies from model to model and from one validation size to another. One interesting question to ask is how these three aggregation models behave when they have similar parameters. We use a technique that we introduced in Section 4.4 to convert the transmission parameter into an equivalent window parameter so that we can compare those three models together under the same window parameter.

Figure 6 and Figure 7 are the experimental results to compare different experimental results under the same window parameters. Figure 6 and Figure 7 use the same experimental results as in Figure 4 and Figure 5, but are aggregated over the validation size rather than the parameter space. As a result, those two figures carry different information than that in Figure 4 and Figure 5. In Figure 6, the horizontal axis is changed from validating size to the window parameter. Each point in the graph represents the regression experiment that optimizes the MSE with the same window-size parameter but different validation size. The real-valued transmission parameter of the Exponential model is now replaced by a discrete-valued window size by using technique introduced in Section 4.4. This makes one of the benefits of the Exponential model, which is the better expressiveness by using a continuously changed transmission parameter to disappear. However, even when present the results in that way, the main conclusion we get from previous plots remain the same: the Exponential model performs better than Linear and Average model across different datasets. Figure 7 is a similar plot to Figure 6 but shows the confidence intervals. Again, using two metrics together achieves a better performance in measuring temporal patterns. The reduction in MSE and subsequent tither confidence interval of the Exponential model is even more significant when using both metrics together as opposed to only Persistence.

One important observation from Figure 6 and Figure 7 is that the change of the best model's prediction error over window parameters for each dataset and activity measure has different patterns. For some experiments such as degree centrality on DBLP, the performance remains basically unchanged for different window sizes. For the case of Arab Spring however, all centrality measure exhibit a pattern that the prediction errors first increase and then decrease as the window parameter increases. This illustrates that there are potential long-distance temporal dependencies between the dynamic metrics and the activity measures in the past. We call this distance the

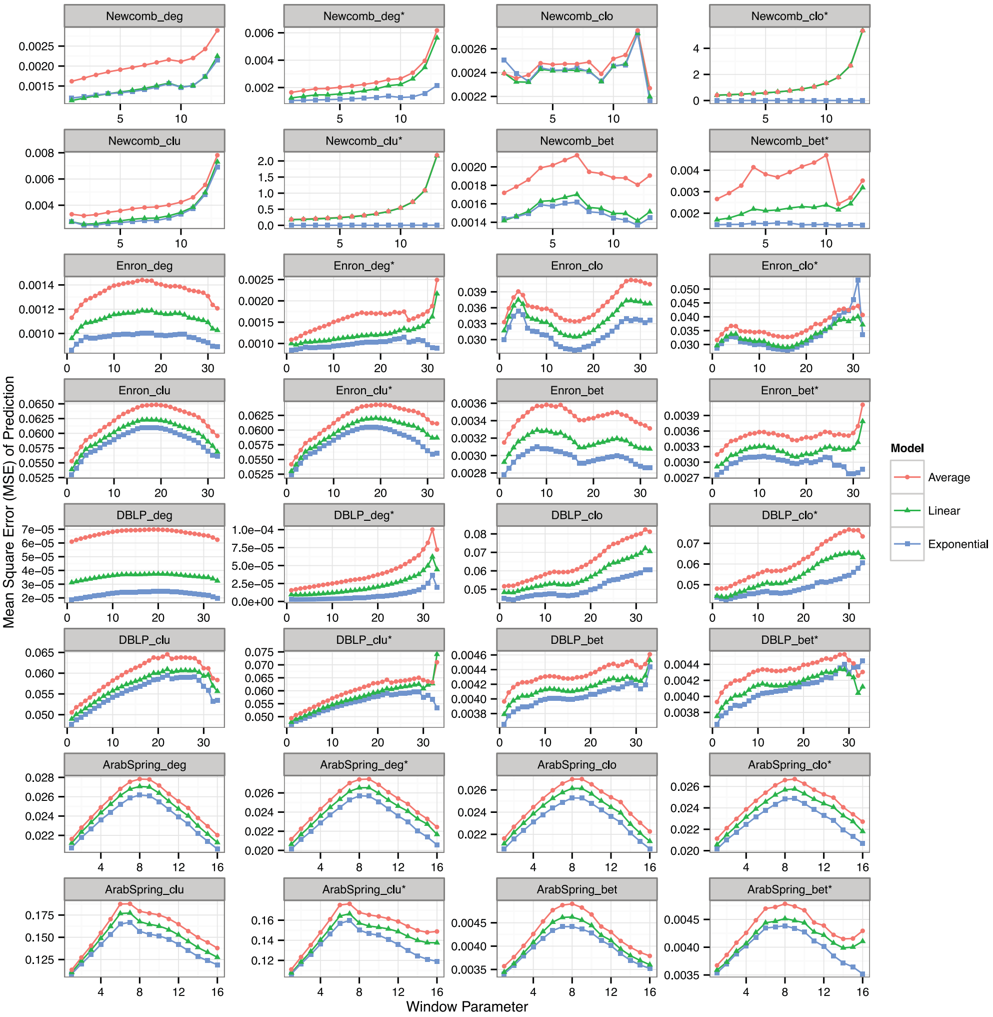

Fig. 6. Comparison of prediction performance across window parameter.

best training distance, which is the difference between two window sizes that achieve the best prediction accuracies. Regression models can achieve good performance when the window size is small but the same prediction error can also be observed again when the window size is large enough. The concept of best training distance is similar to that of the best validating distance since they both demonstrated long distance temporal dependencies between activity measures on different timesteps. However, the best training distance is the distance from the dynamic metrics to the past activity measures, while the best validating distance is the distance from dynamic metrics to the future activity measures. In addition, we observe that those experiments first saw significant increase and then decrease pattern in Figure 6 and Figure 7 are those experiments that have best validating distance large than 1. This indicated that experiments that showed long-distance dependencies in the past also demonstrated similar long-distance dependencies to the future.

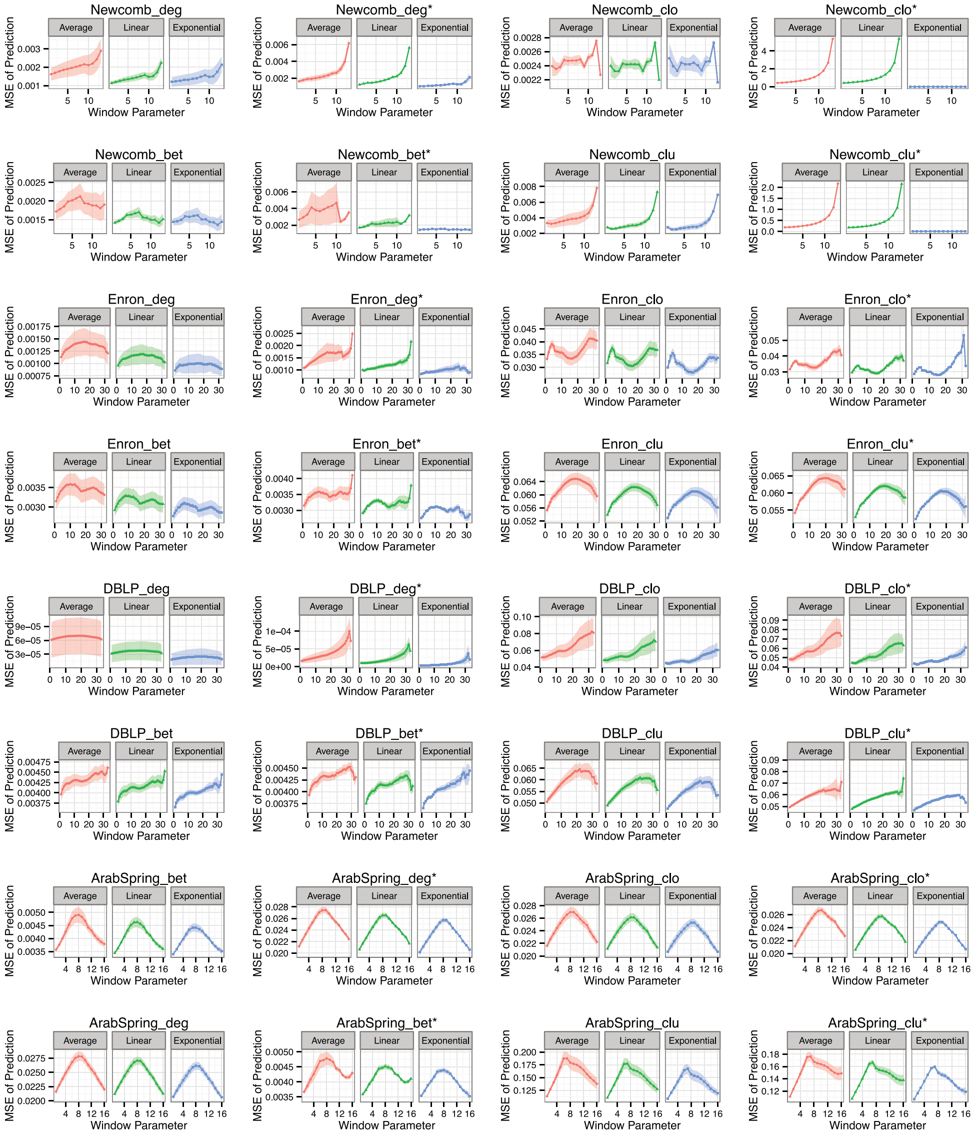

Fig. 7. Comparison of prediction performance across window parameter with confidence interval.

6.4. Measuring Temporal Invariants of Dynamic Social Networks

Results from the previous sections suggested the existent of an inherent long distance temporal dependency exist in the dynamic social network. However, those analyses only show results for a specific parameter/validation size setting. In this section, we want to figure out whether the long-distance temporal dependencies exist regardless of the validating size.

Figure 8 illustrates a comparison of the window parameter across different temporal aggregation models. Here, for each validating size ranging from 1 to the maximum possible value, we record the optimal parameter to achieve the minimal MSE of

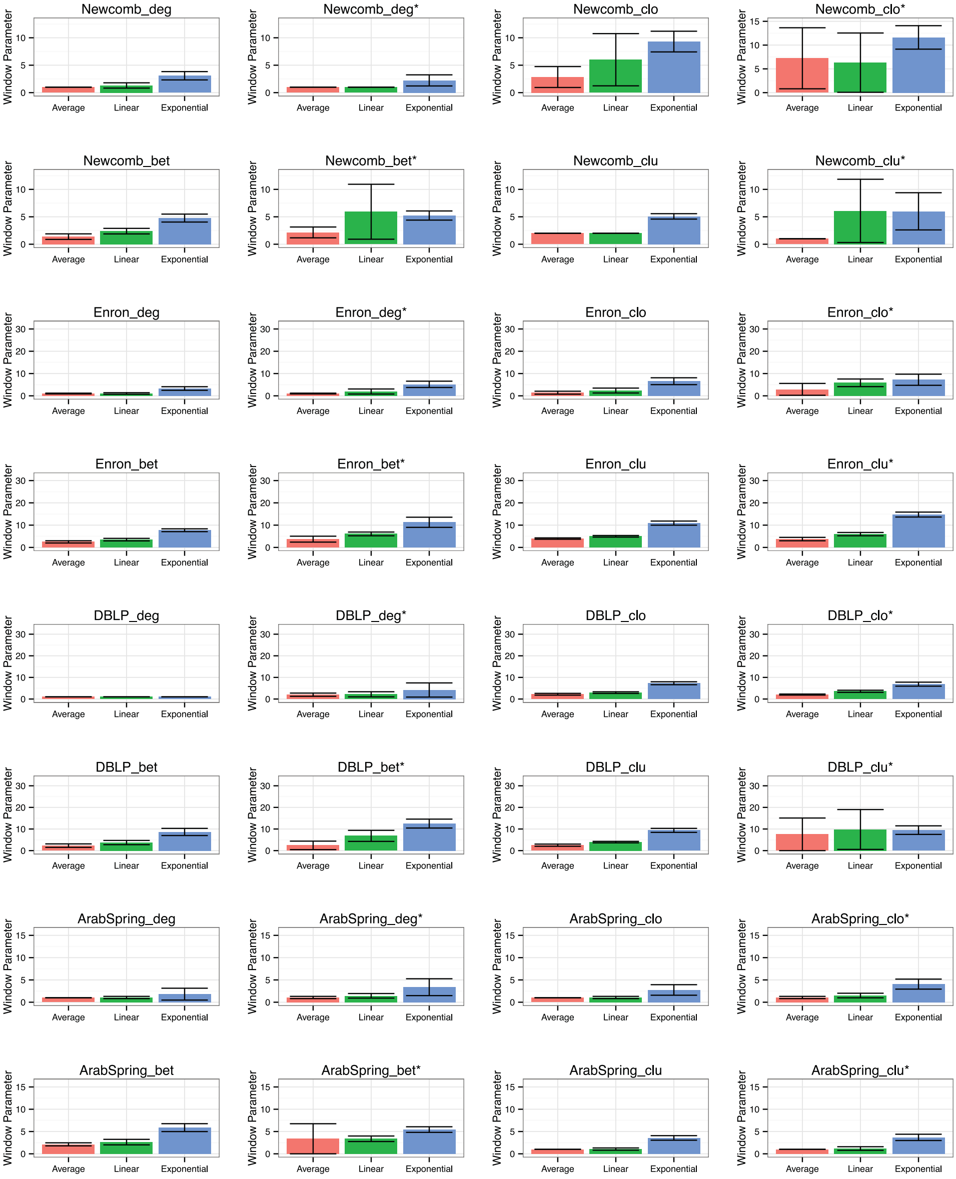

Fig. 8. Mean and confidence intervals of window parameters that achieves optimal performance.

prediction. We then calculate the mean and confidence intervals of those optimal parameters of experiments conducted in each dataset. We call the average window parameter to optimize prediction error the ***reference window***. The mean and standard deviations of reference window is reported in Figure 8 as well. We see that most of the experiments have a relatively stable reference window size with a small standard deviation. This illustrates an internal long-distance dependency within a dataset that is independent of the prediction task. Another observation is that exponential model usually requires more data to achieve the optimal prediction error and always has a small confidence interval. This is due to the flexibility of the model itself and its ability to take more information in the distant past into account. For example, the Newcomb dataset has a reference window of three-to-five weeks, which is a reasonable cycle for a real social network. For Arab Spring dataset, the reference window size is something between three and five months. These analyses suggested the existence of temporal invariants in the dynamic social network.

6.5. Running Time Analysis

One of the key considerations in practice is the running time of the metrics. As earlier theoretical analysis suggests, both Average and Exponential models can be calculated recursively and the execution time is in constant complexity. However, since there are no recursive solutions to the Linear model, its complexity is proportional to the length of the input activity measures. We recorded the execution time of dynamic metrics on each of the three temporal aggregation models. In order to compare models with different running time, we again use the technique introduced in Section 4.4 to convert the transmission parameter of the Exponential model into the window-size parameter. We reported the mean and confidence interval of the running time in each group. The running time only includes the execution time of calculating dynamic metrics, which is illustrated in Algorithms 1, 2, and 3. The running time of regression models, which is used to evaluate our metrics, is not included.

Figure 9 is a comparison of the running time of different models in each of the datasets we used in the experiments. The horizontal axis is the window parameter and the vertical axis is the actual running time. Each point in the graph represents the mean execution time of several sets of experiments with the same aggregation model and parameter. Confidence intervals are also plotted for each window size. We see that the Average and Exponential model basically follow a constant complexity of running time when model parameter changes.

7. FUTURE WORK

This research lays the groundwork for many additional developments. One thing that deserves further exploration is how to make the model parameters adaptable with the network data. In our analysis, the mode parameters are fixed prior to the analysis and stay unchanged later on. Future work should consider building a dynamic model that tweaks the parameters to fit the changes of incoming data adaptively. Such an application would be particularly useful in contexts where network externalities are changing network dynamics. Second, we choose activity measures that are used to characterize network structure. There are many other activity measures that are not related to network structures. Future research should consider what kind of activity metrics could yield the most meaningful results in a dynamic setting. Third, in the definition of Emergence we use two different ways to normalize the Emergence to make it become an indicator that is independent of the magnitude of network changes. More analysis might be necessary to understand the impacts of those normalizations and compare the normalized and non-normalized versions of Emergence. Finally, further empirical work on the temporal dimension of social network data is critical including an evaluation of what additional features, beyond Persistence and Emergence, are necessary in understanding network dynamics.

8. CONCLUSIONS AND LIMITATIONS

In this article, we explored the problem of measuring temporal patterns in dynamic social networks. We argued that the patterns of evolution could be modeled in a manner similar to that of the human cognitive process using the principles of Recency

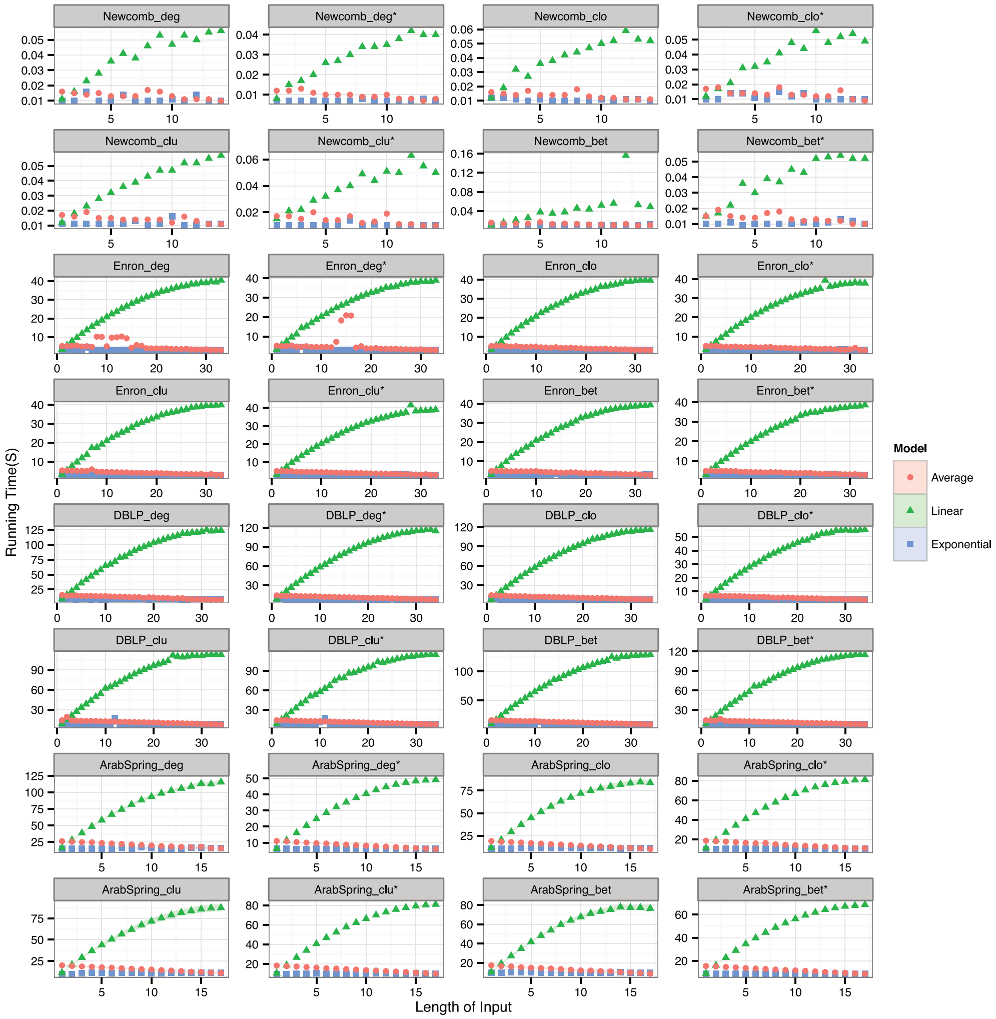

Fig. 9. Comparison of running time under window parameter space.

and Primacy. We built two classes of dynamic metrics and implemented those metrics with three different temporal aggregation models. We found that the extents to which we model Recency and Primacy effects are closely related to the performance of these metrics in terms of the ability to predict future network activities. In particular, the Exponential aggregation model has the best practice of Recency–Primacy effect and achieves the best performance. We also found that there exist intrinsic temporal invariants in dynamic social networks. These temporal invariants remain similar for each agent over time and reflect the temporal dependencies of activities between the recent and the past. We conclude that the activities of agents in the social network are highly predictable given the retrospective information within a fixed temporal length, which patterns can be measured using our dynamic metrics.

There are several limitations to this work. For example, the work presented in the article is based on the consistency assumption of human behavior. Although our experiment results support this consistency, it is still possible that sporadic human behavior might invalidate the basic model assumption in some situation.

There are many applications of this research. First, the dynamic metrics proposed in this research can be used in anomaly detection. Using Persistence and Emergence, we can easily identify agents who show sharp increases in network activities. In online social network websites, those accounts that have abnormal activities might be spammers or robots. The metrics proposed in this research provided a scalable method to detect those abnormalities in large dynamic social networks. Second, the fact that our metrics have statistical correlations to future activities suggests the possibility of estimating agent level activities in dynamic social networks. Although our approach cannot be used to predict the exact link changes, we can use the metrics to predict the activity measures on those link changes. This makes our metrics useful in forecasting the future importance of agents within a network according to the activity measures used without knowing the exact network representations. Understanding the changes in the activity measures such as centralities is key to many problems such as recommendation systems and user impact estimations. Finally, our metrics can be used to study the periodicity of network changes. The temporal invariants can be used to support the studies of periodicity such as the life cycles of fashions, gossips and how they impact the evolution of social networks.

REFERENCES

R. Albert and A. L. Barabási. 2002. Statistical mechanics of complex networks. *Reviews of Modern Physics* 74, 1, 49–52.

J. M. Anthonisse. 1971. *The rush in a directed graph.* Vol. Technical Report BN 9/71. Stichting Mathematisch Centrum, Amsterdam.

A. L. Barabási and R. Albert. 1999. Emergence of scaling in random networks. *Science* 286, 5439, 509–512.

L. Bian and R. Butler. 1999. Comparing effects of aggregation methods on statistical and spatial properties of simulated spatial data. *Photogrammetric Engineering and Remote Sensing* 65, 73–84.

P. Bonacich. 1972. Factoring and weighting approaches to status scores and clique identification. *Journal of Mathematical Sociology* 2, 1, 113–120. DOI:citeulike-article-id:3333118

S. P. Borgatti. 2003. The Key Player Problem. In *Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers*. National Academies Press, 241.

K. M. Carley. 1990. Group stability: A socio-cognitive approach. *Advances in Group Processes* 7, 1–44.

W. Chen, A. Collins, R. Cummings, T. Ke, Z. Liu, D. Rincon, and Y. Yuan. 2011. Influence maximization in social networks when negative opinions may emerge and propagate. In *SDM*. Vol. 11, 379–390.

J. Deese and R. A. Kaufman. 1957. Serial effects in recall of unorganized and sequentially organized verbal material. *Journal of Experimental Psychology* 54, 3, 180–187.

D. M. Dunlavy, T. G. Kolda, and E. Acar. 2011. Temporal link prediction using matrix and tensor factorizations. *ACM Transactions on Knowledge Discovery from Data* 5, 2, 1–27. DOI:10.1145/1921632.1921636

P. Erdős and A. Rényi. 1960. On the evolution of random graphs. *Magyar Tud. Akad. Mat. Kutató Int. Közl* 5, 17–61.

M. Faloutsos, P. Faloutsos, and C. Faloutsos. 1999. On power-law relationships of the internet topology. In *ACM SIGCOMM Computer Communication Review*. Vol. 29, ACM, 251–262.

L. C. Freeman. 1979. Centrality in social networks conceptual clarification. *Social Networks* 1, 3, 215–239.

J. Friedman, T. Hastie, and R. Tibshirani. 2010. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software* 33, 1, 1–22.

T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin. 2005. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer* 27, 2, 83–85.

A. E. Hoerl and R. W. Kennard. 1970. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics* 12, 1, 55–67.

C. I. Hovland. 1957. The order of presentation in persuasion. Yale University Press, Inc.

M. Kas, K. M. Carley, and L. R. Carley. 2013. *Incremental Closeness Centrality for Dynamically Changing Social Networks. Workshop on the Semantic and Dynamic Analysis of Information Networks (SDAIN'13).*

In *Proceedings of the 2013 IEEE / ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, August 25–28, 2013, Niagara Falls, Canada.

M. Kas, K. M. Carley, and L. R. Carley. 2015. An incremental algorithm for updating betweenness centrality and k-betweenness centrality and its performance on realistic dynamic social network data. *Social Network Analysis and Mining (SNAM)*. Springer

D. Kempe, J. Kleinberg, and E. Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM.

B. Klimt and Y. Yang. 2004. The enron corpus: A new dataset for email classification research. *Machine Learning: ECML 2004*, 217–226.

J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins. 2008. *Microscopic evolution of social networks.* Paper presented at the Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, Las Vegas, Nevada, USA.

J. Leskovec, J. Kleinberg, and C. Faloutsos. 2007. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data* 1, 1, 2. DOI: 10.1145/1217299.1217301

Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng. 2009. Analyzing communities and their evolutions in dynamic social networks. *ACM Transactions on Knowledge Discovery from Data* 3, 2, 1–31. DOI: 10.1145/1514888.1514891

N. Miller and D. T. Campbell. 1959. Recency and primacy in persuasion as a function of the timing of speeches and measurements. *The Journal of Abnormal and Social Psychology* 59, 1, 1–9.

B. B. Murdock Jr. 1962. The serial position effect of free recall. *Journal of Experimental Psychology; Journal of Experimental Psychology* 64, 5, 482–488.

T. N. Newcomb. 1961. *The acquaintance process.* New York: Holt, Rinehart and Winston.

M. E. J. Newman. 2001a. Clustering and preferential attachment in growing networks. *Physical Review E* 64, 2, 025102.

M. E. J. Newman. 2001b. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences* 98, 2, 404–409.

M. E. J. Newman. 2004. Coauthorship networks and patterns of scientific collaboration. *Proceedings of the National Academy of Sciences of the United States of America* 101, Suppl 1, 5200–5205.

M. E. J. Newman. 2005. Power laws, Pareto distributions and Zipf's law. *Contemporary Physics* 46, 5, 323–351.

L. A. Palinkas, J. C. Johnson, J. S. Boster, and M. Houseal. 1998. Longitudinal studies of behavior and performance during a winter at the South Pole. *Aviation, Space, and Environmental Medicine* 69, 1, 73–77.

A. K. Romney. 1988. Quantitative models, science and cumulative knowledge. School of Social Sciences, University of California, Irvine.

S. F. Sampson. 1968. *A Novitiate in a Period of Change: An Experimental and Case Study of Social Relationships*. Ph.D dissertation. Cornell University.

A. Sanil, D. Banks, and K. Carley. 1995. Models for evolving fixed node networks: Model fitting and model testing. *Social Networks* 17, 1, 65–81. DOI: 10.1016/0378-8733(94)00249-a.

R. Tibshirani. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* 267–288.

D. J. Watts and S. H. Strogatz. 1998. Collective dynamics of 'small-world' networks. *Nature* 393, 6684, 440–442. DOI: citeulike-article-id:99 DOI: 10.1038/30918.

W. Wei, J. Pfeffer, J. Reminga, and K. M. Carley. 2011. *Handling Weighted, Asymmetric, Self-Looped, and Disconnected Networks in ORA.* Technical Report No. CMU-ISR-11-113. Institute of Software Research, Carnegie Mellon University Pittsburgh, PA.

O. R. Zaiane, J. Chen, and R. Goebel. 2007. DBconnect: Mining research community on DBLP data. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis*. ACM, 74–81.

H. Zou and T. Hastie. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67, 2, 301–320.

Received May 2013; revised November 2014; accepted March 2015