A SHRINKAGE APPROACH TO TRACKING DYNAMIC NETWORKS

Kevin S. Xu [*], *Mark Kliger* [†], *and Alfred O. Hero III* [*]

[*] University of Michigan, Ann Arbor, MI 48109 USA
[†] Medasense Biometrics Ltd., PO Box 633, Ofakim, 87516 Israel
[*] {xukevin,hero}@umich.edu, [†] mark@medasense.com

ABSTRACT

The analysis of network data is of interest to many disciplines, ranging from sociology to computer science. Recent interest has shifted from static networks to dynamic networks, which evolve over time. A fundamental problem in the analysis of dynamic networks is tracking long-term trends, which are obscured by short-term variations. In this paper, we propose a method for minimum mean-squared error tracking of dynamic networks using a recursive shrinkage estimation framework that accounts for the spatial correlation in the network. Unlike model-based tracking methods such as the Kalman filter, the proposed method does not require knowledge about the network dynamics. We demonstrate that the proposed method is able to track dynamic networks effectively through experiments on simulated and real networks.

Index Terms— Dynamic network, time-varying network, tracking, shrinkage, prediction

1. INTRODUCTION

Many developments have been made in recent years in the study of networks, including social, information, and biological networks among others. Networks provide a natural structure for modeling and representing relational data. Until recently, networks have been analyzed as static entities. However, many real networks are dynamic; that is, they evolve over time. In particular, the affinities between nodes in a network can change, resulting in changes in the adjacency matrix and the corresponding network topology. The change is often due to a combination of short-term variation, which is inherently unpredictable, and long-term trends or drifts, which are often obscured by the short-term variation.

In this paper, we consider the problem of tracking dynamic networks. Specifically, we are interested in estimating the aforementioned long-term trends in dynamic networks from periodic observations of the affinities. Dynamic networks present two significant challenges that prevent us from using traditional signal processing methods such as Kalman filtering, namely very high dimensionality and lack of models for the temporal dynamics. Existing studies on tracking dynamic networks have typically applied simplistic methods such as the sliding window or exponentially weighted moving averages [1, 2] with ad-hoc selection of parameters.

We propose a method for tracking dynamic networks using shrinkage estimation. We treat the adjacency matrices of time-evolving networks as the outputs of a dynamic system, with states corresponding to long-term trends in the affinities. We recursively estimate these states using a convex combination of the current observation and the previous state estimate. The interpretation of this update as a shrinkage estimate allows us to optimize the weight in the convex combination to minimize the mean-squared error (MSE) of the estimate in terms of the Frobenius norm. The proposed method does not require an explicit model to be specified for the state evolution; instead, it uses the spatial correlation in the network to estimate all necessary quantities. Through experiments on simulated and real networks, we find that the proposed method performs well in practice and is able to adapt quickly to changing environments.

This work was partially supported by the National Science Foundation grant CCF 0830490. Kevin Xu was partially supported by an award from the Natural Sciences and Engineering Research Council of Canada.

2. PROBLEM FORMULATION

The objective of this paper is to accurately track dynamic networks. We assume that the network is observed sequentially at discrete time steps. The observation at each time step is a weighted graph snapshot represented by adjacency matrix $W^t = \left[w_{ij}^t\right]$, where w_{ij}^t denotes the edge weight between nodes i and j at time step t. The edge weights correspond to the affinities between nodes; higher weights are indicative of stronger connections[1]. We assume that the graph is undirected, so that $w_{ij}^t = w_{ji}^t$; however, the methods considered in this paper generalize in a straightforward manner to directed graphs.

We define a network state matrix $\Psi^t = \left[\psi_{ij}^t\right]$ at each time step. Ψ^t can be viewed as a trend matrix that reflects long-term variations in the affinities. It is an unobservable quantity in practical applications. We posit the linear observation model

$$W^t = \Psi^t + N^t, \tag{1}$$

where N^t is a symmetric matrix of zero-mean random variables, not necessarily independent or identically distributed, that accounts for short-term variations in the affinities.

Under the observation model of (1), the objective of tracking dynamic networks becomes simply estimation of the state matrix Ψ^t at each time step. In this paper, we measure estimation error by the squared Frobenius error. The tracking problem at time step t can then be formulated as

$$\min_{\hat{\Psi}^t} \mathrm{E}\left[\left\|\hat{\Psi}^t - \Psi^t\right\|_F^2 \,\middle|\, W^t, W^{t-1}, \ldots, W^0\right].$$

The matrices W^t, Ψ^t, and N^t can be equivalently represented as the vector quantities $\mathbf{w}^t$, $\boldsymbol{\psi}^t$, and $\mathbf{n}^t$, respectively. This is the typical representation in the tracking literature. The vector quantity $\mathbf{w}^t$, for example, can be obtained from the matrix W^t simply by stacking the columns of the lower triangular portion of W^t. The vectors have length $p = n(n-1)/2$, assuming the graph is undirected and has no self-edges.

[1] Edge weights can also be negative, indicating disaffinity between nodes.

3. EXISTING TRACKING METHODS

3.1. Model-free methods

Two of the simplest and most popular methods for model-free (non-parametric) tracking are the sliding window (SW) and exponentially weighted (EW) moving averages, which are described by the following equations:

$$\hat{\boldsymbol{\psi}}_{SW}^{t} = \frac{1}{l} \sum_{s=t-l+1}^{t} \mathbf{w}^{s}$$

$$\hat{\boldsymbol{\psi}}_{EW}^{t} = (1-\alpha) \sum_{s=0}^{t} (\alpha)^{s} \mathbf{w}^{t-s},$$

where $(\alpha)^s$ indicates α taken to the exponent s. Both methods have the desirable property of not requiring the entire measurement history $\{\mathbf{w}^s\}_{s=0}^{t}$. SW requires only the most recent l measurements, where l corresponds to the window length, and EW can be implemented recursively according to

$$\hat{\boldsymbol{\psi}}_{EW}^{t} = \alpha \hat{\boldsymbol{\psi}}_{EW}^{t-1} + (1-\alpha)\mathbf{w}^{t}. \tag{2}$$

The parameters l and α in SW and EW, respectively, control the amount of weight applied to historical measurements. l is also subject to constraints, namely memory and processing time, while α, commonly referred to as the *forgetting factor*, is a parameter that can chosen freely by the user. The choice of parameter is crucial to the accuracy of these methods; however, existing studies on SW and EW tracking of dynamic networks [1, 2] have typically chosen it in an ad-hoc fashion.

A more principled approach [3] for EW is to choose α to minimize the average one-step prediction error

$$e_{pred} = \frac{1}{t} \sum_{s=0}^{t-1} \left\| \hat{\boldsymbol{\psi}}_{EW}^{s} - \mathbf{w}^{s+1} \right\|_2^2 \tag{3}$$

over a training period. We refer to this method as *global training*. Another method examined in [3] is to choose α adaptively by minimizing prediction error over local windows of length l. In other words, the summation in (3) is taken from $t-l$ to $t-1$ rather than from 0 to $t-1$, and the forgetting factor becomes a time-varying quantity α^t. We refer to this method as *local training*.

A drawback of EW is that it preserves historical affinities indefinitely, which densifies the network topology as t increases. Cortes et al. [1] addressed this problem by thresholding extremely low edge weights in the EW estimate to 0, which removes the edges from the network so that the topology does not get progressively denser.

3.2. Model-based methods

Unlike model-free methods, model-based (parametric) methods require knowledge of a model that accurately describes the dynamics of the network. The standard minimum mean-squared error (MMSE) estimator for tracking states in linear time-varying systems is the Kalman filter [4]. We assume a random walk model for the state evolution over time, where

$$\boldsymbol{\psi}^{t} = \boldsymbol{\psi}^{t-1} + \mathbf{m}^{t} \tag{4}$$

with $\mathbf{m}^t \sim \mathcal{N}(0, R^t)$. We also assume that $\mathbf{n}^t \sim \mathcal{N}(0, Q^t)$. Finally, we assume that the initial state is distributed according to $\mathcal{N}(\boldsymbol{\mu}^0, C^0)$. $\mathbf{m}^t$ and $\mathbf{n}^t$ are assumed to be mutually independent for all t and independent of the initial state. The Kalman filter for the dynamic model of (1) and (4) gives the following state estimate:

$$\hat{\boldsymbol{\psi}}_{K}^{t} = \hat{\boldsymbol{\psi}}_{K}^{t-1} + K^{t}\left(\mathbf{w}^{t} - \hat{\boldsymbol{\psi}}_{K}^{t-1}\right) \tag{5}$$

where K^t is the optimal Kalman gain matrix [4]. Since the observation at each time step is an $n \times n$ matrix, or equivalently, a vector of length p, the Kalman filter requires the inversion of a $p \times p$ matrix in order to calculate the optimal Kalman gain, which presents a computational problem for many applications involving large dynamic networks containing thousands of nodes.

It is known that the EW method is equivalent to the Kalman filter under the aforementioned random walk model for univariate state and observation [4]. This can be seen by rearranging (2) to obtain

$$\hat{\boldsymbol{\psi}}_{EW}^{t} = \hat{\boldsymbol{\psi}}_{EW}^{t-1} + (1-\alpha)\left(\mathbf{w}^{t} - \hat{\boldsymbol{\psi}}_{EW}^{t-1}\right). \tag{6}$$

Note that (6) has the same form as (5), with the only difference being that the Kalman gain K^t has been replaced with the scalar $(1-\alpha)$. Hence the EW update has the same form as the Kalman filter in the univariate case, but is more restrictive in the multivariate case. Unlike the Kalman filter, however, it is tractable even in large networks.

4. TRACKING BY SHRINKAGE ESTIMATION

We propose to track dynamic networks by shrinkage estimation of the network state matrix at each time step. Since the Kalman filter is computationally infeasible, we take the EW recursion in (2) as the starting point. Define the shrinkage estimate of Ψ^t to be

$$\hat{\Psi}^{t} = \alpha^{t} \hat{\Psi}^{t-1} + \left(1-\alpha^{t}\right) W^{t}. \tag{7}$$

This is a generalization of EW with a time-varying forgetting factor. $\hat{\Psi}^t$ can be viewed as a convex combination of two estimators of Ψ^t. W^t is an unbiased estimator because N^t is zero-mean; however, it has high variance due to the lack of time averaging. On the other hand, $\hat{\Psi}^{t-1}$ is likely biased but has lower variance. Thus the selection of α^t can be considered as a bias-variance trade-off. We choose α^t to minimize MSE, which is the sum of the variance and the square of the bias.

4.1. Derivation of oracle forgetting factor

We derive the optimal choice of the forgetting factor α^t under the linear observation model of (1). We refer to this as the *oracle forgetting factor* because it requires oracle knowledge of the model parameters. Define the risk at time step t by

$$R(\alpha^{t}) = \mathrm{E}\left[\left\| \hat{\Psi}^{t} - \Psi^{t} \right\|_F^2 \,\middle|\, \hat{\Psi}^{t-1} \right], \tag{8}$$

the conditional expectation of the estimation error at time step t given the previous estimate $\hat{\Psi}^{t-1}$. Substituting (1) and (7) into (8), the risk can be expressed as

$$\begin{aligned} R(\alpha^{t}) = \sum_{i=1}^{n} \sum_{j=1}^{n} \Big\{ & \operatorname{var}\left(\alpha^{t} \psi_{ij}^{t} + \left(1-\alpha^{t}\right) n_{ij}^{t} \,\middle|\, \hat{\Psi}^{t-1} \right) \\ & + \mathrm{E}\left[\alpha^{t} \hat{\psi}_{ij}^{t-1} + \left(1-\alpha^{t}\right) n_{ij}^{t} - \alpha^{t} \psi_{ij}^{t} \,\middle|\, \hat{\Psi}^{t-1} \right]^2 \Big\}. \end{aligned} \tag{9}$$

Assuming N^t is independent of Ψ^t and of N^s and Ψ^s for all $s < t$, (9) simplifies to

$$\begin{aligned} R(\alpha^{t}) = \sum_{i=1}^{n} \sum_{j=1}^{n} \Big\{ & \alpha^{t} \left(\hat{\psi}_{ij}^{t-1} - \mathrm{E}\left[\psi_{ij}^{t} \,\middle|\, \hat{\Psi}^{t-1} \right] \right)^2 \\ & + \left(\alpha^{t}\right)^2 \operatorname{var}\left(\psi_{ij}^{t} \,\middle|\, \hat{\Psi}^{t-1} \right) + \left(1-\alpha^{t}\right)^2 \operatorname{var}\left(n_{ij}^{t} \right) \Big\}. \end{aligned} \tag{10}$$

Set the first derivative of (10) to 0 to obtain the minimizer

$$\left(\alpha^t\right)^* = \frac{\sum_{i=1}^n \sum_{j=1}^n \text{var}\left(n_{ij}^t\right)}{\sum_{i=1}^n \sum_{j=1}^n b_{ij}^t} \tag{11}$$

where

$$b_{ij}^t = \left(\hat{\psi}_{ij}^{t-1} - \text{E}\left[\psi_{ij}^t \mid \hat{\Psi}^{t-1}\right]\right)^2 + \text{var}\left(\psi_{ij}^t \mid \hat{\Psi}^{t-1}\right) + \text{var}\left(n_{ij}^t\right).$$

$\left(\alpha^t\right)^*$ is confirmed to be a minimizer because $R''\left(\alpha^t\right)$ is positive everywhere. Under the model assumptions described in Section 3.2, all of the random quantities are Gaussian, so the conditional distribution of Ψ^t given $\hat{\Psi}^{t-1}$ is also Gaussian, and (11) can be computed solely as a function of the model parameters $\boldsymbol{\mu}^0$, C^0, Q^t, and R^t.

4.2. Estimation of oracle forgetting factor

The oracle forgetting factor $\left(\alpha^t\right)^*$ requires oracle knowledge of three quantities: $\text{E}\left[\psi_{ij}^t \mid \hat{\Psi}^{t-1}\right]$, $\text{var}\left(\psi_{ij}^t \mid \hat{\Psi}^{t-1}\right)$, and $\text{var}\left(n_{ij}^t\right)$. In most dynamic network applications, these quantities are unknown, so $\left(\alpha^t\right)^*$ is also unknown. We propose the following strategy for estimating $\left(\alpha^t\right)^*$. First we relate the unobservable quantities ψ_{ij}^t and n_{ij}^t to the observable quantities w_{ij}^t. By the independence of N^t and W^s for all $s < t$, the conditional mean of w_{ij}^t is given by

$$\text{E}\left[w_{ij}^t \mid \hat{\Psi}^{t-1}\right] = \text{E}\left[\psi_{ij}^t \mid \hat{\Psi}^{t-1}\right], \tag{12}$$

and the conditional variance is given by

$$\text{var}\left(w_{ij}^t \mid \hat{\Psi}^{t-1}\right) = \text{var}\left(\psi_{ij}^t \mid \hat{\Psi}^{t-1}\right) + \text{var}\left(n_{ij}^t\right). \tag{13}$$

We assume that the states are evolving slowly over time, so that the contribution of $\text{var}\left(\psi_{ij}^t \mid \hat{\Psi}^{t-1}\right)$ in (13) is negligible.

The unknown quantities are now expressed as conditional moments of observable rather than unobservable quantities, but these moments are still unknown. We can obtain an estimate $\left(\hat{\alpha}^t\right)^*$ of $\left(\alpha^t\right)^*$ by replacing these moments with their sample equivalents. This strategy was employed by [5, 6] in the context of covariance estimation, where sample variances are computed across independent and identically distributed (iid) data replicates. In the network setting, iid replicates are not available, so we pursue an alternate approach to obtain sample moments.

Our approach takes advantage of the spatial correlation of networks and involves replacing the time moments defined in (12) and (13) by spatial sample moments. This strategy was used in [7] for the purpose of clustering, where the underlying assumption was that the state matrix had a block structure according to the cluster memberships. If there is a priori knowledge of the correlation structure, it can also be used. If no a priori information is available, one may compute sample moments using assumptions regarding the network's temporal or spatial nature. For example, in this paper we assume that neighboring edges are correlated and compute spatial sample moments of w_{ij}^t using neighboring edges of i and j. The intuition behind this assumption is that in most networks, including social networks, changes in the states are initiated by the nodes, so they should affect all of the edges attached to that node. Using the sample moments, we calculate $\left(\alpha^t\right)^*$, which allows us to compute the estimated state $\hat{\Psi}^t$.

As mentioned in Section 3.1, thresholding is necessary to prevent the network from getting denser as t increases. In our experiments, we heuristically choose the threshold so that the network does not densify while preserving at least 99% of the total energy of $\hat{\Psi}^t$. This approach is probably not optimal; however, a thorough investigation of thresholding is beyond the scope of this paper.

5. EXPERIMENTS

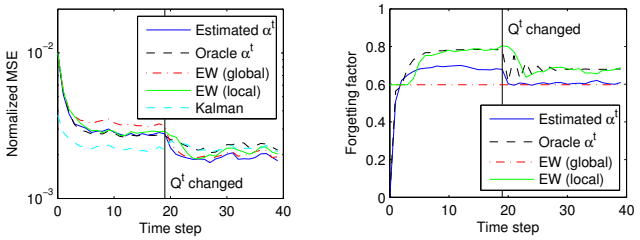
5.1. Simulated network

The first experiment is to track states of a simulated evolving network. The observations are generated according to the linear observation model of (1) and the state evolutions according to the random walk model of (4). The initial state is generated in the following manner. First, we create a base network using the Erdos-Renyi model [8] for unweighted, undirected graphs with $p = 0.2$. The edges in this base network are the only non-zero states. They are Gaussian distributed with mean $\boldsymbol{\mu}^0 = 10$ and covariance matrix C^0 constructed by $c_{ij}^0 = 0.5^{\|i-j\|}$, where $\|i-j\|$ denotes the minimum number of nodes in the base network that need to be traversed to reach j from i. For example, if i and j share a common node, $\|i-j\| = 1$. The structure of C^0 is inspired by banded models for covariance matrices when the variables represent quantities such as time. Here we are assuming a banded model in space over the base network. $\mathbf{m}^t$ is taken to be Gaussian with covariance matrix $0.1C^0$ so that states evolve in a correlated manner.

The experiment is conducted over 40 time steps, with a change in the model at $t = 19$. At each time step, we select some of the non-edges in the base network to be spurious (noise-only) edges. Initially we select 10% of the non-edges to be spurious; at $t = 19$, we drop this to 5%. The observation noise $\mathbf{n}^t$ is Gaussian with covariance matrix $Q^t = I$ initially. At $t = 19$, we change Q^t to $0.5C^0$ so that the observation noise becomes correlated and decreases in power. The change in Q^t allows us to test the ability of the estimators to respond to a change in network dynamics.

We simulated 100 dynamic networks with 20 nodes using the above procedure. The normalized tracking MSE at each time step $\|\hat{\Psi}^t - \Psi^t\|_F^2 / \|\Psi^t\|_F^2$ is plotted in Fig. 1(a) for the proposed method, the globally and locally trained EW methods, and the Kalman filter. The window length is taken to be 3 for the trained EW methods. The Kalman filter and oracle α^t require knowledge of all the model parameters. We do not provide these methods with the modified Q^t so we can observe the loss due to model mismatch. As expected, the Kalman filter dominates the other methods when the model is correctly specified, but its performance does not improve after Q^t is changed, unlike the other methods. As mentioned in Section 3.2, it also presents a computational problem for larger networks, hence why we limit the simulation to 20 nodes. Notice that the estimated α^t initially performs almost as well as the oracle α^t, indicating that the proposed method of sampling neighboring edges does a good job of estimating the unknown moments. After the change in model, the oracle α^t performs worse due to the model mismatch. Also notice that the estimated α^t performs better overall than both the global and local EW methods.

The obtained forgetting factors are shown in Fig. 1(b). Unlike in Fig. 1(a), the oracle α^t is provided with the modified Q^t for comparison purposes. While the mean local EW α^t is closer to the oracle α^t than the mean estimated α^t, local EW does not perform better in terms of MSE because the local EW α^t has a higher standard error of 0.103 compared to the estimated α^t, which has a standard error of 0.075. In addition, notice that the estimated α^t is able to imme-



(a) Normalized MSE (b) Forgetting factor

Fig. 1. Performance comparison for simulated network.

diately detect the change in Q^t, while the local EW α^t shows a lag. This is due to ability of the proposed method to estimate moments by averaging over a spatial window (neighboring nodes) as compared to averaging over a time window, which is the strategy adopted by the locally trained method.

5.2. Reality Mining

The second experiment involves real dynamic network data collected from the MIT Reality Mining project [9] as part of an experiment on inferring social network structures. The data was collected by recording cell phone activity of 94 students and staff at MIT over a year. The phones were equipped with Bluetooth sensors, and each phone recorded the Media Access Control addresses of nearby Bluetooth devices at five-minute intervals. Using this device proximity data, we construct a sequence of adjacency matrices where the affinity between two participants corresponds to the number of intervals where they were in close physical proximity within a time step.

We divide the data into time steps of one week, resulting in 46 time steps between August 2004 and June 2005. In this data set we do not know the true state values, so we cannot compute the normalized tracking MSE as in the previous experiment. Instead, we compute the normalized prediction MSE $\|\hat{\Psi}^t - W^{t+1}\|_F^2 / \|W^{t+1}\|_F^2$, which is plotted in Fig. 2. We do not know the model parameters for this data set so the oracle α^t and Kalman filter cannot be used. Four important dates, taken from the MIT academic calendar [10], are shown. Note that the error increases significantly around three of these dates, which makes sense because students' proximity networks should vary significantly depending on whether they are attending classes at the time. The EW training methods both perform especially poorly around these dates compared to the estimated α^t, while at other time steps, the performance is similar.

Fig. 3 shows how the forgetting factor α^t varies over time. Notice that the estimated α^t drops around three of the important dates, suggesting that these are change points. On the other hand, the local training method shows a significant lag. Again, this is one of the advantages of the proposed method over local training, namely that it is able to provide effective smoothing yet still detect changes with minimal lag by using spatial correlation.

6. CONCLUSIONS

We examined the problem of tracking dynamic networks and proposed a shrinkage estimator for the time-evolving adjacency matrix. The proposed estimator provides an explicit formula for the optimal choice of forgetting factor. Although the optimal forgetting factor requires oracle knowledge of the unknown dynamics, they can be empirically estimated from sample correlations of activity at nodes

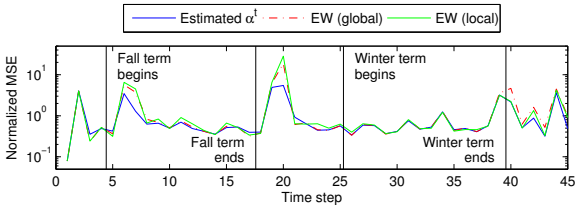


Fig. 2. Normalized prediction MSE for MIT Reality Mining data.

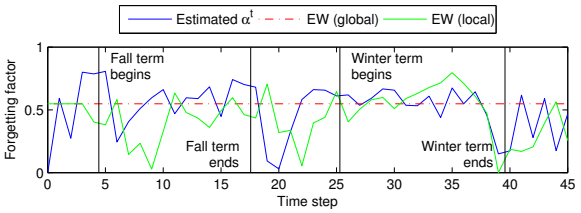


Fig. 3. Forgetting factors for MIT Reality Mining data.

in the network. This allows the estimator to be used for tracking dynamic networks where no dynamic model is known, a situation that is typical in most applications.

7. REFERENCES

[1] C. Cortes, D. Pregibon, and C. Volinsky, "Computational methods for dynamic graphs," *J. Computat. Graphic. Statist.*, vol. 12, no. 4, pp. 950–970, 2003.

[2] H. Tong, S. Papadimitriou, P. S. Yu, and C. Faloutsos, "Proximity tracking on time-evolving bipartite graphs," in *Proc. SIAM Conf. Data Mining*, 2008.

[3] M. Y. Cheng, J. Fan, and V. Spokoiny, "Dynamic nonparametric filtering with application to volatility estimation," in *Recent Advances and Trends in Nonparametric Statistics*. Elsevier, 2003.

[4] A. C. Harvey, *Forecasting, structural time series models and the Kalman filter*, Cambridge University Press, 1989.

[5] O. Ledoit and M. Wolf, "Improved estimation of the covariance matrix of stock returns with an application to portfolio selection," *J. Empirical Finance*, vol. 10, no. 5, pp. 603–621, 2003.

[6] J. Schäfer and K. Strimmer, "A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics," *Statist. Applic. Genetics Molec. Biology*, vol. 4, no. 1, pp. 32, 2005.

[7] K. S. Xu, M. Kliger, and A. O. Hero III, "Evolutionary spectral clustering with adaptive forgetting factor," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2010.

[8] B. Bollobás, *Random graphs*, Cambridge University Press, 2nd edition, 2001.

[9] N. Eagle, A. Pentland, and D. Lazer, "Inferring friendship network structure by using mobile phone data," *Proc. Nat. Acad. Sci.*, vol. 106, no. 36, pp. 15274–15278, 2009.

[10] "MIT academic calendar 2004-2005," `http://web.mit.edu/registrar/www/calendar0405.html`.