

Recovering time-varying networks of dependencies in social and biological studies

Amr Ahmed and Eric P. Xing¹

Language Technology Institute and Machine Learning Department, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213

Edited by Stephen E. Fienberg, Carnegie Mellon University, Pittsburgh, PA, and approved April 29, 2009 (received for review February 23, 2009)

A plausible representation of the relational information among entities in dynamic systems such as a living cell or a social community is a stochastic network that is topologically rewiring and semantically evolving over time. Although there is a rich literature in modeling static or temporally invariant networks, little has been done toward recovering the network structure when the networks are not observable in a dynamic context. In this article, we present a machine learning method called TESLA, which builds on a temporally smoothed l_1 -regularized logistic regression formalism that can be cast as a standard convex-optimization problem and solved efficiently by using generic solvers scalable to large networks. We report promising results on recovering simulated time-varying networks and on reverse engineering the latent sequence of temporally rewiring political and academic social networks from longitudinal data, and the evolving gene networks over >4,000 genes during the life cycle of *Drosophila melanogaster* from a microarray time course at a resolution limited only by sample frequency.

evolving network | social network | gene network | lasso | Markov random field

In many problems arising in social, biological, and other fields, it is often necessary to analyze populations of entities (e.g., individuals, genes) interconnected by a set of relationships (e.g., friendship, communication, influence) represented as a network. Real-time analysis of network data is important for detecting anomalies, predicting vulnerability, and assessing the potential impact of interventions in various social, biological, or engineering systems. It is not unusual for network data to be large, dynamic, heterogeneous, noisy, and incomplete. Each of these characteristics adds a degree of complexity to the interpretation and analysis of networks.

Classical network analyses mostly assume that the networks in question are fully observable, static, and isotropic. Given the heterogeneity and complexity of network data in many domains, these assumptions are limiting. For example, a majority of current investigations of biological regulatory circuitry focus on networks with invariant topology over a given set of molecules, such as a protein–protein interaction network over all proteins of an organism regardless of the conditions under which individual interactions may take place or a static gene network inferred from microarray data even though the samples may be collected over a time course or multiple conditions. In reality, over the course of a cellular process, such as a cell cycle or an immune response, there may exist multiple underlying “themes” that determine the functionalities of each molecule and their relationships to each other, and such themes are dynamic and stochastic. As a result, the molecular networks at each time point are context dependent and can undergo systematic rewiring, rather than being invariant over time. Indeed, in a seminal study by Luscombe et al. (1), it was shown that the “active regulatory paths” in a gene expression correlation network of *Saccharomyces cerevisiae* exhibit dramatic topological changes and hub transience during a temporal cellular process and in response to diverse stimuli. Similar phenomena are not uncommon in many other domains, such as the evolution of citation networks in scientific communities and the emergence/disintegration of

cryptic liaisons and communities in society. A key technical hurdle that prevents us from an in-depth investigation of the mechanisms underlying this phenomena is the unavailability of serial snapshots of the rewiring network during the unfolding and progression of the process in question. For example, under a dynamic biological system, usually it is technologically impossible to experimentally determine time-specific network topologies for a series of time points based on techniques such as 2-hybrid or ChIP-chip experiments.

Although there is a rich (and growing) literature on modeling observed network data in either static (2) or dynamic scenarios (3–5), with a few exceptions (6, 7), much less has been done toward developing efficient inference techniques for recovering unobserved network topologies from nodal (i.e., entitywise) observations in a dynamic context. Existing methods usually ignore potential network rewiring under different conditions and infer an invariant network (8–10). Recently, a new class of models known as the temporal exponential random graph model (tERGM) has been proposed for modeling networks evolving over discrete time steps (5). It is based on a log-linear graph transition model $P(G|G^{t-1})$ defined on a set of temporal potentials that capture certain characteristics of the graph rewiring dynamics, such as the “edge-stability,” “reciprocity,” and “transitivity” statistics over time-adjacent graphs. Based on tERGM, subsequently, a hidden tERGM model (htERGM) was developed to impose stochastic constraints on latent rewiring graphs so that given a time series of nodal attributes, one can infer time-specific network topology based on the posterior distribution of G^t of every time point (7). However, the MCMC-based algorithm in ref. 7 for posterior inference under htERGM is very inefficient, and up until now, providing an efficient inference algorithms for this model remains an open problem.

We propose TESLA, a machine learning algorithm for recovering the structure of time-varying networks over a fixed set of nodes from time series of nodal attributes. In these networks, an edge between 2 nodes implies a dependency between their nodal states, such as coactivation in a gene regulatory network. TESLA stems from the acronym TESLLOR, which stands for temporally smoothed l_1 -regularized logistic regression. It represents an extension of the well-known lasso-style sparse structure recovery technique and is based on a key assumption that temporally adjacent networks are likely not to be dramatically different from each other in topology and therefore are more likely to share common edges than temporally distant networks. Building on the highly scalable iterative l_1 -regularized logistic regression algorithm for estimating single sparse networks (11), we develop a regression regularization scheme that connects multiple time-specific network inference functions via a first-order edge smoothness function that encour-

Author contributions: E.P.X. designed research; A.A. and E.P.X. performed research; A.A. and E.P.X. contributed new reagents/analytic tools; A.A. and E.P.X. analyzed data; and E.P.X. and A.A. wrote the paper.

The authors declare no conflict of interest.

This article is a PNAS Direct Submission.

Freely available online through the PNAS open access option.

¹To whom correspondence should be addressed. E-mail: epxing@cs.cmu.edu.

This article contains supporting information online at www.pnas.org/cgi/content/full/0901910106/DCSupplemental.

ages edge retention between time-adjacent networks. An important property of this idea is that it fully integrates all available samples of the entire time series in a single inference procedure that recovers the rewiring patterns between nodes over a time series of arbitrary resolution—from a network for every single time point, to one network for every K time points where K is very small. Importantly, TESLA can be cast as a convex optimization problem for which a globally optimal solution exists and can be efficiently computed for networks with thousands of nodes. Recently, Zhou et al. (6) proposed a nonparametric local kernel-weighting technique for learning smoothly evolving graphical Gaussian models by using the method in ref. 12 and showed interesting asymptotic consistency results of their estimator. In contrast, here, we address the case of discrete time-evolving graphs modeled as evolving Markov random fields rather than graphical Gaussian models, and our method is a more direct formalism that uses the smooth-evolution property as an explicit penalty in the loss function that can be traded off with data fitness, rather than an implicit condition needed to guarantee consistency. Our ongoing theoretical analysis of TESLA suggests that it can detect blockiness and jumps in graph evolution, and possibly enjoys graph-structure consistency rather than parameter-estimation consistency as shown in ref. 6 for their method.

We applied TESLA to the recovery of a time-varying social network in the U.S. Senate based on 2 years of voting records, an evolving author-keyword network from the proceedings of a machine learning conference over 13 years, and a sequence of 23 epoch-specific networks over 4,028 genes during the life cycle of *Drosophila melanogaster* based on a microarray time course. TESLA represents a practical and scalable method for recovering large-scale time-varying networks underlying sociocultural and biological processes at arbitrary temporal resolutions.

Problem Formulation

For concreteness, consider the problem of estimating a sequence of time-varying dependency networks underlying a fixed set of genes with discrete states (e.g., active or inactive) over a dynamic biological process. Denote by $G^t = (V, E^t)$ the graph structure at time t with node (i.e., gene) set V of size $|V| = p$ and edge set E^t . Let $\mathcal{D} = \{\mathbf{X}^t\}_{t=1}^T$ represents the set of samples of node states associated with all of the nodes of the graph over time. For generality, the time stamp t can be also understood as denoting the onset of a time epoch of length N_t . When $N_t = 1$, we have a single snapshot of node states at time t ; whereas when $N_t > 1$, we assume that we have multiple *iid* observations of all node states corresponding to graph G^t within epoch t ; hence, $\mathbf{X}^t = \{\mathbf{X}_1^t, \dots, \mathbf{X}_{N_t}^t\}$, $\mathbf{X}_d^t \in \{0, 1\}^p$. It should be apparent that estimating a time-specific network G^{t^*} naively based on $\{\mathbf{X}^{t^*}\}$ for every t^* is an ill-defined problem, because this corresponds to the near-extreme case of “large p (dimension size) small n (sample size) problem” in statistical inference, where, in principle, one wants $n = N_{t^*} = 1$ to obtain truly “time-specific” estimation. Even with some relaxation on time specificity, N_t would still be very small to reflect changes of G^t with acceptable resolution.

A legitimate assumption one could exploit to overcome this problem, as we shall do, is that the graphs $\{G^t\}$ are not *iid*, but follow a time-evolution process, so that all samples in \mathcal{D} reflect something about the graph G^{t^*} for every $t^* \in \{1, \dots, T\}$. Consider the following model:

$$P(\mathbf{x}^t | \Theta^t) = \exp \left(\sum_{i \in V} \theta_{ii}^t x_i^t + \sum_{(i,j) \in E^t} \theta_{ij}^t x_i^t x_j^t - A(\Theta^t) \right), \quad [1]$$

where the joint probability distribution of the random variables at each time t is given by a pairwise Markov random field (MRF). In Eq. 1, the parameters $\{\theta_{ij}^t\}_{(i,j) \in E^t}$ capture the correlation (or the dependency strength) between variables X_i^t and X_j^t , and $A(\Theta^t)$ is the log partition function of the distribution at time t . Given a set

of samples $\{\mathbf{x}_{1:N_t}^t\}$ drawn from $P(\mathbf{x}^t | \Theta^t)$ at every time step $t = 1, \dots, T$, our goal is to estimate the structure of the graph at every time, i.e., to estimate $\{\hat{E}^t\}_{t=1}^T$. Note that $\{\hat{E}^t\}_{t=1}^T$ can be directly obtained from $\{\hat{\Theta}^t\}_{t=1}^T$ through a sign function, with $\text{sign}(\theta) = 0$ if $\theta = 0$, $\text{sign}(\theta) = 1$ if $\theta > 0$, and $\text{sign}(\theta) = -1$ if $\theta < 0$. We propose the following estimator for $\{\hat{\Theta}^t\}_{t=1}^T$, which we'd like to call TESLA, obtained by solving a temporally smoothed l_1 -regularized logistic regression problem for reasons we shall explain in the next section:

$$\hat{\theta}_i^1, \dots, \hat{\theta}_i^T = \arg \min_{\theta_i^1, \dots, \theta_i^T} \sum_{t=1}^T \ell(\mathbf{x}^t; \theta_i^t) + \lambda_1 \sum_{t=1}^T \|\theta_i^t\|_1 + \lambda_2 \sum_{t=2}^T \|\theta_i^t - \theta_i^{t-1}\|_1, \quad \forall i, \quad [2]$$

where

$$\begin{aligned} \ell(\mathbf{x}^t; \theta_i^t) &= \sum_{d=1}^{N_t} \log P(x_{d,i}^t | \mathbf{x}_{d,-i}^t, \theta_i^t) \\ &= \sum_{d=1}^{N_t} (\log(1 + \exp(\mathbf{x}_{d,-i}^t \theta_i^t)) - \mathbf{x}_{d,-i}^t \theta_i^t x_{d,i}^t), \end{aligned}$$

where θ_i^t denotes a $(p-1)$ -dimensional column vector corresponding to the i th column of parameter matrix Θ^t of the MRF excluding θ_{ii}^t , i.e., the pairwise correlation coefficients from all other nodes in V to node i at time t ; $\mathbf{x}_{d,-i}^t$ denotes the observed states of all nodes except node i in the d th sample in time epoch t ; $\ell(\cdot)$ represents the log conditional likelihood of state $x_{d,i}^t$ given $\mathbf{x}_{d,-i}^t$ under a logistic regression model; and $\|\cdot\|_1$ represents the l_1 norm of a vector. Note that the TESLA problem in Eq. 2 will be defined and solved for each node i in V , and in the end we recover an estimate of $\{\hat{\Theta}^t\}_{t=1}^T$ via a simple “signed” max (or min) magnitude function: $\hat{\theta}_{ij}^t = \max(|\hat{\theta}_{ij}^t|, |\hat{\theta}_{ji}^t|) \times \text{sign}(\arg\max(|\hat{\theta}_{ij}^t|, |\hat{\theta}_{ji}^t|))$, $\forall i \neq j, t$, and $\hat{\theta}_{ii}^t = \hat{\theta}_{ii}^t, \forall t$. The edge set E can be subsequently recovered from the nonzero entries in Θ^t by using the sign function. As we explain in the sequel, the problem in Eq. 2 has a close connection to the graphical lasso method (13) for (static) sparse graph estimation and the fused lasso method (14) for coupled multivariate regression. Thus, many of the virtues and theoretical insights of these methods, such as computational efficiency, sparsity, consistency, may apply to TESLA.

We are aware of 2 earlier attempts on estimating time-varying graphs. In ref. 7, the time evolution of graph sequence $\{G^t\}$ is modeled by a hidden temporal exponential random graph model, which allows one to explicitly infer a posterior distribution of G^{t^*} conditioning on all samples in \mathcal{D} in much the same spirit of performing posterior decoding of hidden states in a hidden Markov model, albeit now with a computational complexity that is at least $O(2^{p^2})$ per iteration under an MCMC-based inference algorithm. In ref. 6, a time interval-dependent kernel was applied to weight every sample in \mathcal{D} with respect to t^* , which are subsequently treated as reweighted *iid* samples from G^{t^*} for estimating G^{t^*} . It can be shown that if $\{\mathbf{X}^t\}$ are continuous valued and follow a time-varying graphical Gaussian model (GGM), and when the precision matrix Θ of the GGM is assumed to be smoothly evolving, this scheme can consistently recover the (value of) Θ^{t^*} corresponding to every G^{t^*} in the limit (6). However, another important type of consistency known as model-selection consistency, or pattern consistency, which concerns the identification of nonzero elements of Θ^{t^*} , and hence directly leads to recovery of the topology E^{t^*} of graph G^{t^*} , is not available. It is noteworthy that in ref. 6, the smooth evolution assumption is used as a precondition to guarantee value consistency

of the Θ^{t*} estimator, rather than being treated as an explicitly tunable constraint that can be used to accommodate real data. Therefore, in cases where more turbulent dynamics (e.g., sudden jumps) drives graph evolution, this method may run into difficulties. The TESLA model presented above represents a more direct approach that allows more flexible tradeoff among smoothness of graph evolution, sparsity of graph estimation, and likelihood of the fit. Although we focus on discrete-valued networks in this article, our technique can be readily applied to continuous-valued networks. In the following section, we explain in detail the rational behind the TESLA model, and an algorithm that efficiently solves the estimation problem.

Temporally Smoothed Sparse Graphical Regression

It is reasonable to expect that the time-varying networks underlying biological or social processes represent a sparse structure that evolves smoothly over time. The TESLA objective defined in Eq. 2 is essentially a temporally smoothed sparse graphical regression problem aiming at reconstructing such networks. It builds on 3 key ideas that are extensively studied in recent literature for structure recovery from high-dimensional data: (i) structural estimation via graphical regression; (ii) sparsity and consistency via l_1 shrinkage of parameters; (iii) structure coupling via parameter fusion. Below, we briefly discuss each of these ideas and its relevance to our problem.

Structural Recovery, Sparsity, and Smoothness. When $T = 1$, the optimization problem in Eq. 2 reduces to a static case, where we estimate a single network based on N_1 iid samples. This problem has been heavily studied in recent literature [see [supporting information \(SI\)](#)]. Our discussion thus starts from this base case.

It is well known that for both continuous-valued networks after a GGM and discrete-valued networks after a MRF, estimation of model parameters (which bear the structural information) can be cast as a graphical regression problem in which, for every node X_i , one defines it as a response over predictors corresponding to the remaining $p - 1$ nodes \mathbf{X}_{-i} via a linear or logistic function. To estimate a discrete network modeled by a MRF, the loss function of the graphical logistic regression is not exactly equivalent to the likelihood function of the original MRF but corresponds to a pseudolikelihood, $\hat{P}(\mathbf{X}|\Theta) = \prod_{i=1}^n P(X_i|\mathbf{X}_{N(i)})$, where $N(i)$ is the Markov blanket of node i , i.e., the neighboring nodes of node i (11). In the binary pairwise MRF, the local likelihood $\hat{P}(X_i|\mathbf{X}_{-i}, \theta_i)$ has a logistic-regression form. Thus, the problem of learning Θ degenerates to solving p logistic regression problems each defined on an individual node with respect to all of the other nodes in the graph. This gives rise to the first term of the TESLA objective in Eq. 2.

Because of presence of noise, and in the common situation where $p \gg n$, it is well known that traditional regression methods do not offer a model selection power and tend to overfit data. The immediate consequence of this pitfall on graph estimation is a dense resulting graph that contains a large amount of spurious edges. The lasso method proposed by Tibshirani (15), which solves an l_1 norm regularized least-square linear regression, has a parsimonious property, and exhibits model-selection consistency (i.e., recovers the set of true nonzero regression coefficients asymptotically) in noisy settings even when $p \gg n$. When applied in the graphical regression context, it becomes what is known as the graph-lasso algorithm. It has been shown that the vanishing lasso coefficient estimates identify asymptotically the neighborhood of every node in the graph (10) and thus lead to a consistent estimation of the structure of a continuous-valued network under a GGM. For discrete-valued graphs, Wainwright et al. (11) have shown that applying the l_1 norm-regularized logistic regression over each node with respect to all other nodes can also lead to consistent neighborhood selection over every node and hence converges to the true graph structure asymptotically, under a MRF. In the spirit of graph-lasso, we refer to this algorithm as graph-LLR, which motivates the second term, denoted as L^{shrink} , in Eq. 2.

If, rather than estimating a single graph, one wants to estimate, say, 2 graphs G^1 and G^2 that are topologically close to each other, some constraints need to be applied to their corresponding parameter matrices Θ^1 and Θ^2 . Under the graph-lasso or graph-LLR scheme, this is equivalent to coupling 2 regression problems $X_i^1 = f(\mathbf{X}_{-i}^1; \theta_i^1)$ and $X_i^2 = f(\mathbf{X}_{-i}^2; \theta_i^2)$ by penalizing the discrepancy between θ_i^1 and θ_i^2 , for every i . A similar problem that focuses on encouraging low variance or blocky structure within a coefficient vector θ (rather than between 2 θ s with elementwise correspondence as considered in this article) has recently received remarkable attention. Several extensions have been proposed over the l_1 penalty to enforce smoothness in addition to sparsity over elements in θ . In the fused-lasso method proposed in ref. 14, a fusion penalty in the form of $L^{\text{smooth}} \equiv \sum_{l=2}^p |\theta_l - \theta_{l-1}|$ was used in addition to the lasso penalty over the regression loss. In this penalty, the ordering of the predictors is used as a proxy for coupling predictors of similar coefficients, and the ordering is supplied by using domain knowledge to the model. This penalty, together with the lasso penalty, has been shown to result in sequences of zero (or nonzero) parameters in θ , thus enforcing both smoothness and sparsity. It is easy to see that a penalty of the form $L^{\text{shrink}} + L^{\text{smooth}}$ can be not only applied to elements within a single θ but also to corresponding elements across different θ s, whereby both sparsity within the regression function and smoothness across θ s can be enforced. The L^{smooth} now represents a total variation (TV) penalty, which motivates the third term in Eq. 2.

To summarize, to estimate a sequence of T time-varying networks, at each time point (or epoch) t , we defined a graph-LLR problem over samples $\mathbf{x}_{1:N}^t$. We used the negative pseudolikelihood based on a logistic regression as a surrogate to the intractable log-likelihood function under a MRF, and used an l_1 norm penalty over each column of Θ^t (the regression coefficient vectors of nodes in G^t) to achieve a shrinkage effect on the neighbors of each node. Then, to incorporate temporal dependencies between time-specific graphs that resulted from the graph-LLR, we furthermore introduced a fusion penalty $L^{\text{smooth}} \equiv \|\Theta^t - \Theta^{t-1}\|_1$ over the difference between every pair of Θ^{t-1} and Θ^t , the regression coefficient vectors corresponding to the same node at 2 consecutive time points (or epochs), to bias our estimate to graphs that evolve in a smooth fashion. It is noteworthy that this term can be interpreted as the “edge-stability” potential in the more general htERGM model proposed earlier (7). But the TESLA formulation decouples the learning problem in ref. 7, which is computationally intractable, into a set of p separate regularized regression problems, 1 for each node, as made explicit in Eq. 2.

Convex Optimization Algorithm. The problem in Eq. 1 is a convex optimization problem with nonsmooth constraints. We can instead solve the following equivalent problem by introducing new auxiliary variables, \mathbf{u}_i^t and \mathbf{v}_i^t :

$$\begin{aligned} \min_{\substack{\theta_1^1, \dots, \theta_1^T, \\ \mathbf{u}_1^1, \dots, \mathbf{u}_1^T, \mathbf{v}_1^2, \dots, \mathbf{v}_1^T}} \quad & \sum_{t=1}^T \ell(\mathbf{x}^t; \theta_i^t) + \lambda_1 \sum_{t=1}^T \mathbf{1}^T \mathbf{u}_i^t + \lambda_2 \sum_{t=2}^T \mathbf{1}^T \mathbf{v}_i^t \quad [3] \\ \text{s.t.} \quad & -u_{ij}^t \leq \theta_{ij}^t \leq u_{ij}^t, \quad t = 1, \dots, T, \quad \forall j \in V \setminus i, \\ & -v_{ij}^t \leq \theta_{ij}^t - \theta_{ij}^{t-1} \leq v_{ij}^t, \quad t = 2, \dots, T, \quad \forall j \in V \setminus i, \end{aligned}$$

where $\mathbf{1}$ denotes a vector with all components set to 1, so $\mathbf{1}^T \mathbf{u}_i^t$ is the sum of the components of \mathbf{u}_i^t and likewise for $\mathbf{1}^T \mathbf{v}_i^t$. To see the equivalence of the problem in Eq. 3 with the one in Eq. 2, we note that at the optimal point of Eq. 3, we must have $u_{ij}^t = |\theta_{ij}^t|$ and similarly, $v_{ij}^t = |\theta_{ij}^t - \theta_{ij}^{t-1}|$, in which case the objectives in Eq. 3 and Eq. 2 are equivalent (a similar transformation scheme has been applied to solving the l_1 -regularized logistic regression in ref. 16).

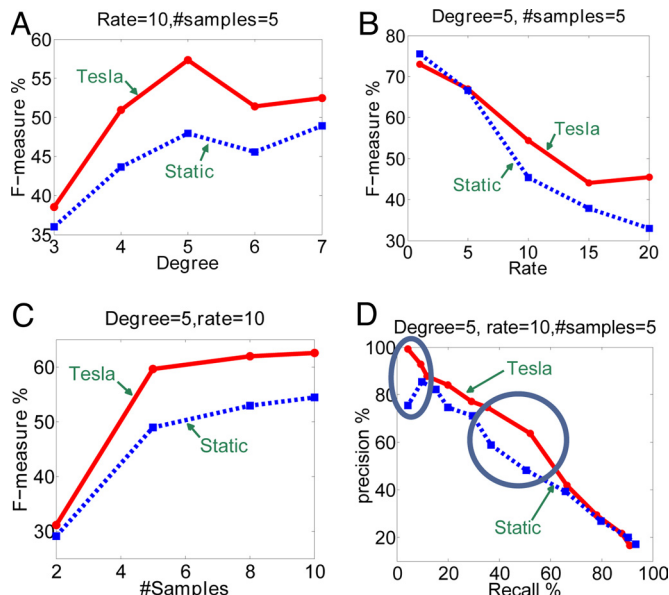


Fig. 1. Simulation results. (A–C) The effect of varying one simulation dimension while fixing the others. (D) The precision-recall curve for a single simulation run (degree = 5, no. of samples = 5, and rate = 10).

Eq. 3 is again a convex optimization problem but now with a smooth objective and linear constraint functions, so it can be solved by standard convex optimization methods, such as the interior point methods (also referred to as barrier methods) (16). In fact, high-quality solvers using interior-point methods are available to directly handle the problem in Eq. 3 efficiently for medium- to large-scale networks (up to a few thousands of nodes and a few hundreds of samples). In this article, we used the CVX optimization package (<http://stanford.edu/boyd/cvx>). In SI, we detail a criteria for selecting the hyperparameters using the BIC score.

Experimental Results

Now we demonstrate TESLA on recovering synthetic time-varying networks and several real-world evolving sociocultural and biological networks from time series of node states. We used a variety of network layout software for visualizing the recovered networks (see SI for more details).

Simulation Results. Based on a simple edge birth–death model defined on a 3-tuple (p, d, r) , we simulated time series of rewiring networks and their corresponding node states and empirically evaluated TESLA on its performance in recovering the time-varying networks from samples of node states. Here, p denotes the number of nodes in the network, d denotes the maximum degree of nodes in the first network, and r determines the number of randomly added/deleted edges to/from the network at each change point. We began by randomly generating edges in the first network

such that each node has a maximum of d edges. The MRF weights of the edges were then sampled uniformly from interval $[-1, -0.5] \cup [0.5, 1]$. We then added and removed r edges every 10 epochs (see SI for more details). We generated a sequence of 100 networks, 1 for each time epoch and then N_t samples of nodal states at each epoch using Gibbs sampling. Performance was evaluated by the F-measure, which is defined as the harmonic mean of precision and recall. We compared TESLA with a static graph-LLR that pools all samples in the time series and estimates a single time-invariant network. The regularization parameters (λ_1, λ_2) for TESLA and λ_1 for the static method were selected from the set $\{0.001, 0.005, \dots, 0.002\}$ for λ_1 and $\{0.1, 0.3, \dots, 2\}$ for λ_2 using the BIC score (see SI).

Fig. 1 shows that TESLA dominates the static approach in network recovery under a wide range of conditions. As expected, the performance went down as we increased the degree of the networks while fixing the available samples. However, TESLA performs relatively more stably as the rate of changes of the networks is increased because of its ability to adapt λ_2 . In addition, the performance improves as more samples become available. Moreover, the precision-recall curve in Fig. 1D shows that TESLA significantly outperforms the static method in the high-precision and even precision-recall areas as highlighted.

U.S. Senate Network. We analyzed the voting record of 642 bills brought to U.S. Senate of the 109th Congress (2005–2006) (www.senate.gov). Each of the 100 senators can be regarded as a node in a latent evolving social network in the senate, and the votes on every bill represent a time-specific random sample of the binary states of all nodes in this network. Using TESLA, we hope to discover how the relationships between senators change over time. Previously, this dataset was analyzed in ref. 12, where the time stamps of the bills were ignored, and a static network was estimated.

We grouped the voting records into 12 consecutive time epochs where each epoch spans 2 months. Fig. 2 shows the estimated network at epoch 2 (March 2005), epoch 7 (January 2006), and epoch 10 (August 2006). Democratic senators are shown as rectangular nodes, Republican senators are shown as ellipsoid nodes, and Independent senators are shown as diamonds. As we can see, the network is divided into 2 big components, approximately along party lines. We highlight 3 senators that are of particular interest at the boundary between the 2 parties: Senator Jim Jeffords (I), Senator Lincoln Chafee (R), and Senator Bill Nelson (D) and track their neighborhood over time.

Senator Jeffords is an Independent who was originally a liberal-to-moderate Republican. From the inferred evolving network, we can see that his record overall is more connected to Democratic senators, although in approximately January 2006, his voting pattern changed slightly to have more connections with his own party. In fact, it is a known fact that Senator Jeffords continued to vote with Republicans on many major pieces of legislation that happened to appear around this period. Moreover, although Senator Bill Nelson is affiliated with the Democratic party, he is one of the most conservative democratic senators and his views are more correlated with the Republicans, as was discovered by TESLA.

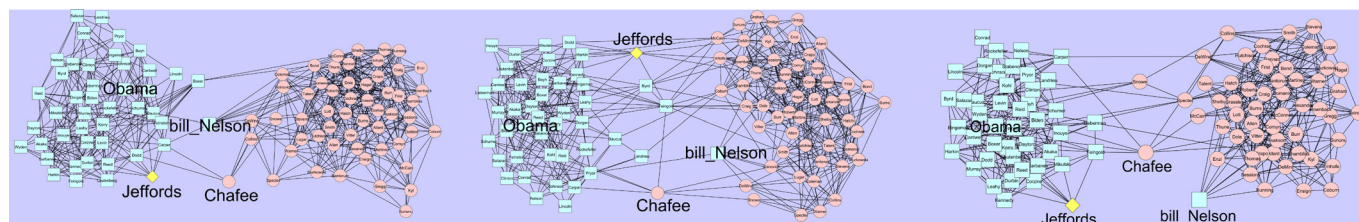


Fig. 2. The dynamic networks of 100 U.S. senators of the 109th Congress. Rectangular, ellipsoid, and diamond nodes denote Democrats, Republicans, and Independents, respectively.

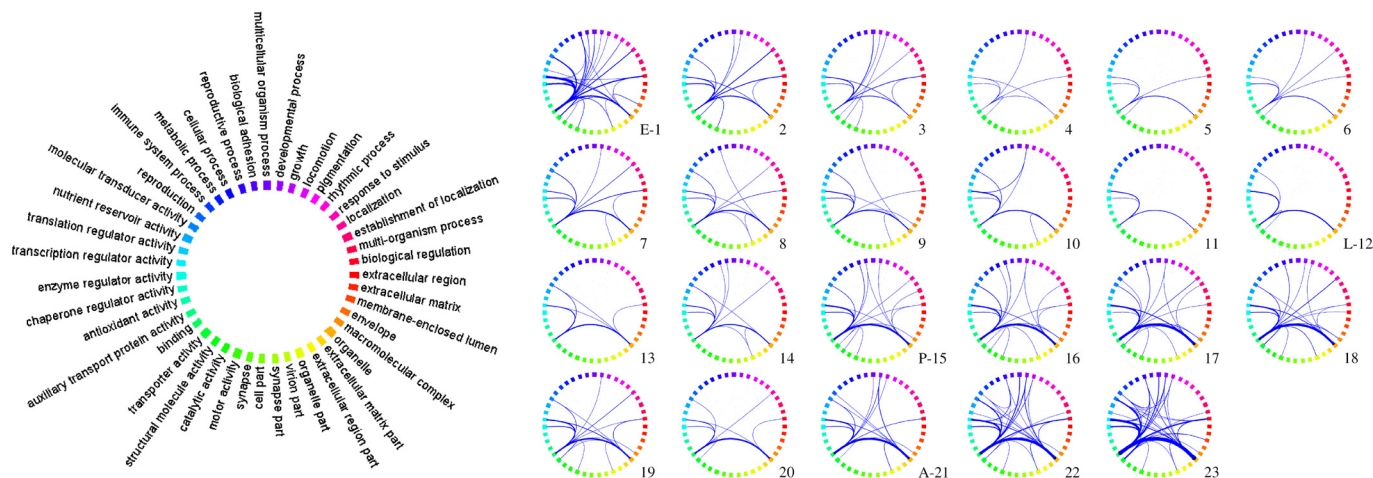


Fig. 3. The temporally rewiring patterns of interactions among gene ontology groups during *D. melanogaster* development. The 4,028 measured genes are grouped according to the 43 ontology groups; the weight of an edge between 2 ontology groups counts the number of connections between genes across these 2 groups. The width of an edge in the visualization is proportional to its weight. For clarity, only edges with weights >5 are displayed. Each ontology group is coded in clockwise order with a color shown in the legend on the upper left corner. Moreover, we highlight the start of each stage by using its first letter.

Another interesting observation can be made on Senator Chafee. He is a Republican; however, as shown in Fig. 2, his views are progressively changing toward the Democrats, as measured by the relative number of connections he has with both parties. In fact, Senator Chafee left the Republican party and became an Independent afterward. Finally, we also highlight Senator Obama (who is now the President of the U.S.) in Fig. 2, and it is clear that he was always connected to only Democratic senators.

Author-Keyword Academic Social Network. The NIPS12 dataset (www.cs.toronto.edu/roweis/data.html) contains the proceedings of the Neural Information Processing Systems (NIPS) conference from the years 1987–1999, which allows us to analyze the dynamics of the academic social network over authors and keywords. As an illustrative case study, we selected the top 36 authors based on publication counts, which resulted in a total of 431 papers; then we selected 73 relevant words from these papers; finally we divided the data into 13 epochs using the time stamp of each paper. Thus, each article in our dataset is a sample from a (time-specific) network with 109 dimensions. To visualize the result, we select 6 keyword nodes: “Gaussian,” “classifier,” “likelihood,” “approximation,” “error,” and “variational” and track their subnetworks containing neighboring authors and keywords up to the first neighbors to avoid overcrowding the figures. See SI for the illustration of the network.

One should observe the smooth transition between the networks from 1987 to 1988 and from 1998 to 1999. Second, the size of the neighborhood of a word is a good indication of the contextual diversity of the usage of this word. For instances, in early years, the word “likelihood” had a limited context in “Gaussian” settings; however, over the years, this behavior changed, and at the years 1998 and 1999, “likelihood” started to appear in different contexts like “speech,” “Bayesian,” “mixture,” etc.; thus, its neighborhood expanded over the years. On the other hand, words like “Gaussian” were always popular in different settings and get more popular over the years, and, as such, the model smoothly expanded its neighborhood as well. It is also interesting to analyze the word “variational,” where in early years it was tightly coupled with the word “Bayesian,” and over the years, its neighborhood was dominated by “Jordan” and “Ghahramani,” who were very active researchers in this area at this time; then, over the year 1999, the neighborhood of the word “variational” expanded to include the word “EM,” as in “variational EM.” Also, note how “Hinton” was always connected to terms related to Boltzmann machines like “distribution” and “field” in the 1980s and “weights” in 1998. Finally, note the

interesting relationship between the word “kernel” and both “Scholkopf” and “Smola.”

Evolving Gene Networks During *Drosophila* Development. The microarray time course from ref. 17 records the expression levels of 4,028 sequence-verified, unique genes measured at 66 time points spanning the embryonic, larval, pupal, and adulthood period during the life cycle of *D. melanogaster*. We detail the preprocessing of these data in SI. Using TESLA, we reconstructed 23 time-varying gene networks, 1 per 3 time points, spanning the embryonic (time points 1–11), larval (time points 12–14), pupal (time points 15–20), and adulthood stages (time points 21–23). It is beyond the scope of this article to fully present and discuss the results of this analysis; instead, to gain a glimpse of the time-specific snapshots and temporal evolution patterns of gene networks in a living organism during its full developmental course, we show in Fig. 3 a display of the transient interactions between functional groups of genes during this process and provide more analyses in SI.

During the life cycle of *D. melanogaster*, many genes are expected to play different roles at different times and interact with different proteins of different functions. According to Flybase (<http://flybase.bio.indiana.edu>), the 4,028 genes in our dataset can be categorized into 3 primary gene ontology (GO) groups: cellular component, molecular function, and biological process. They can be further divided into 43 secondary gene ontology groups. Fig. 3 shows the interactions between these ontology groups that are evolving over time. It is found that, through all of the stages of developmental process, genes belonging to the following GO groups: binding function, transcription regulator activity, and organelle function, are most active. In particular, the group of genes involved in binding functions plays the central role as the hub of the networks of interactions between ontology groups. Genes related to transcriptional regulatory activity and organelles functions exhibit persistent interaction with the group of genes related to binding functions. Other groups of genes that often interact with the binding genes are those related to functions such as developmental process, response to stimulus, and biological regulation.

Large topological changes can be observed from the temporal rewiring patterns between these gene ontology groups. The most diverse interactions between gene ontology groups occur at the beginning of embryonic stage and near the end of adulthood stage. In contrast, near the end of embryonic stage (time point 10), the interactions between genes are largely restricted to those from the following 4 gene ontology groups: transcriptional regulator activity,

enzyme regulator activity, binding, and organelle. However, despite these interesting empirical observations, we would like to emphasize that this dataset has been criticized for missing important developmental genes (e.g., the gaps genes and many pair-rule genes), and is thus perhaps unsuitable for drawing stronger biological conclusions about the mechanisms underlying network evolution. Indeed, our purpose for showing these empirical results is not to suggest any biological findings, but to illustrate a promising utility of TESLA in reverse-engineering time-varying network in large-scale realistic problems.

Discussion and Future Work

In this article, we investigated a class of network structure inference problems that is of practical relevance and importance in network analysis in diverse contexts ranging from biology to social sciences. Rather than treating the networks as observable invariant entities, we view them as latent, dynamically rewiring metastructures behind the observed high-dimensional longitudinal data. We proposed an efficient and highly scalable algorithm called TESLA to infer such time-evolving networks, by solving a set of temporally smoothed l_1 -regularized logistic regression problems via existing convex optimization techniques; and we demonstrated TESLA in several empirical analyses that involve reverse-engineering either a synthetic network, or the real-world time-evolving U.S. Senate network, NIPS author-keyword network, and fruit fly gene network. There are several directions for future work. First, although standard off-the-shelf interior point methods were used successfully in this work, developing a customized solver along the method proposed in ref. 16 is an important future work that could potentially scale up the size of the networks targeted by TESLA by an order of magnitude. On the theoretical side, although it was shown that the solution to the fused lasso problem is asymptotically value-

consistent (14) under the assumption that the model dimension is fixed; it is still an open problem to characterize the solution when the model dimension is allowed to grow and to determine at which rate the sample size should grow with respect to the network size and evolution rate to achieve model-selection consistency in a time-varying network. Although in a recent article Zhou et al. (6) proved that maximizing a regularized likelihood function with weighted samples results in a value-consistent estimation of the covariance matrices associated with a time-varying GGM, their result does not directly apply to TESLA, for which we would like to establish guarantees on model-selection consistency under the more flexible TV-penalized logistic loss for an evolving MRF. Finally, characterizing the uncertainty and statistical significance of structure recovery under finite-sample-size scenarios remains an important issue that has received significant attention in existing work on static structure recovery. For example, Meinshausen et al. (18) proposed a randomization scheme for computing the P values of structure recovery in high-dimensional linear regression. In ref. 19, the concept of false discovery rate was used to characterize the number of nonspurious edges in a learned network. Extending these results to time-varying, high-dimensional problems addressed by TESLA remains an open problem: Theories are needed to calculate the P value of transient links, and appropriate multiple-hypothesis testing may be needed to adjust these P values.

ACKNOWLEDGMENTS. We are indebted to L. Song for help with drawing the biological figures and M. Kolar for preprocessing the Senate dataset. This material is based on work supported by an National Science Foundation (NSF) CAREER Award (to E.P.X.) under Grant DBI-0546594, Defense Advanced Research Projects Agency Award Z931302, Office of Naval Research Award N000140910758, and NSF grant IIS-0713379. E.P.X. is also supported by an Alfred P. Sloan Research Fellowship.

1. Luscombe N, et al. (2004) Genomic analysis of regulatory network dynamics reveals large topological changes. *Nature* 431:308–312.
2. Wasserman S, Robins G (2005) An introduction to random graphs, dependence graphs, and p^* . *Models and Methods in Social Network Analysis* (Cambridge Univ Press, Cambridge, UK).
3. Snijders T (2001) The statistical evaluation of social network dynamics. *Sociological Methodology*, eds Sobel M, Becker M (Blackwell, Boston), pp 361–395.
4. Robins L, Pattison P (2001) Random graph models for temporal processes in social networks. *J Math Social* 25:5–41.
5. Hanneke S, Xing E (2006) Discrete temporal models of social networks. *Workshop on Statistical Network Analysis, the 23rd International Conference on Machine Learning (ICML-SNA)* (Association for Computing Machinery, New York).
6. Zhou S, Lafferty J, Wasserman L (2008) Time varying undirected graphs. *Proceedings of the 21st Annual Conference on Learning Theory, 2008*, <http://colt2008.cs.helsinki.fi/papers/COLT2008.pdf>.
7. Guo F, Hanneke S, Fu W, Xing E (2007) Recovering temporally rewiring networks: A model-based approach. *The 24th International Conference of Machine Learning, 2007* (Association for Computing Machinery, New York).
8. Friedman N, Linial M, Nachman I, Pe'er D (2000) Using Bayesian networks to analyze expression data. *J Comput Biol* 7:601–620.
9. Basso K, et al. (2005) Reverse engineering of regulatory networks in human b cells. *Nat Genet* 37(4):382–390.
10. Meinshausen N, Bühlmann P (2006) High dimensional graphs and variable selection with the lasso. *Ann Stat* 34:1436–1462.
11. Wainwright M, Ravikumar P, Lafferty J (2006) High dimensional graphical model selection using l_1 -regularized logistic regression. *Advances in Neural Information Processing Systems 19* (MIT Press, Cambridge, MA), pp 1465–1472.
12. Banerjee O, El Ghaoui L, d'Aspremont A (2008) Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *J Machine Learn Res* 9:485–516.
13. Friedman J, Hastie T, Tibshirani R (2007) Sparse inverse covariance estimation with the graphical lasso. Tech rep (Stanford Univ, Stanford, CA).
14. Tibshirani R, Saunders M, Rosset S, Zhu J (2005) Sparsity and smoothness via the fused lasso. *J R Stat Soc* 67:91–108.
15. Tibshirani R (1996) Regression shrinkage and selection via the lasso. *J R Stat Soc* 58:267–288.
16. Koh K, Kim S, Boyd S (2007) An interior-point method for large-scale l_1 -regularized logistic regression. *J Machine Learn Res* 8:1519–1555.
17. Arbeitman M, et al. (2002) Gene expression during the life cycle of *Drosophila melanogaster*. *Science* 297:2270–2275.
18. Meinshausen N, Meier L, Bühlmann P (2008) P-values for high-dimensional regression. [arXiv:0811.2177v2](https://arxiv.org/abs/0811.2177).
19. Listgarten J, Heckerman D (2007) Determining the number of non-spurious arcs in a learned dag model: Investigation of a Bayesian and a frequentist approach. *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence (AUAI)* Press, Redmond, WA.