A Constraint Optimization Framework for Efficient Inference in htERGM

Amr Ahmed
Carnegie Mellon University

Eric Xing
Carnegie Mellon University

Extended Abstract

In many problems arising in biology, social sciences and various other fields, it is often necessary to analyze populations of entities such as molecules or individuals that are interconnected by a set of relationships. Studying networks of these kinds can reveal a wide range of information. While there is a rich literature on modeling static networks, much less has been done towards modeling dynamically evolving networks. Recently, [1] proposed the hidden, temporal Exponential Random Graph (htERGM) as a framework for modeling hidden, temporarily evolving networks. In the htERGM, the network structure at each time step t, A^t, is modeled as a hidden variable that evolves according to a Markovian dynamics, and at each epoch t, a set of data points $\{X^t_{1:D_t}\}$ are emitted. Putting everything together:

$$P(A^{1:T}, X^{1:T}) = \prod_{t=1}^{T} P(A^{t+1}|A^t,\theta)P(X^t_{1:D_t}|A^t,\Lambda) \quad (1)$$

$$P(A^{t+1}|A^t,\theta) \propto \exp\{\theta'\Psi(\mathrm{A^t},\mathrm{A^{t-1}})\}\ ,\quad \mathrm{P(x^t_d|A^t},\Lambda) \propto \exp\{\sum_{\mathrm{ij}} \Phi(\mathrm{x^t_{d,i}, x^t_{d,j}, A^t_{ij}},\Lambda_{\mathrm{ij}})\} \quad (2)$$

Where A^t_{ij} is a binary variable that denotes the existence of an edge between nodes i and j, Ψ and Φ are vector-valued features that encodes the sufficient statistics of the exponential families which are parameterized by θ and Λ respectively. The htERGM defines a family of models that depends on how these sufficient statistics are defined. For instance, in [1], $\Psi = (\Psi_1, \Psi_2, \Psi_3)$, where $\Psi_1 = \sum_{ij} -A^t_{ij}$ capturing network density, $\Psi_2 = \sum_{ij} I(A^{t-1}_{ij} == A^t_{ij})$ capturing edge stability, and finally Ψ_3 capturing transitivity (see [1] for more details). Moreover, in [1], all nodes were assumed for simplicity to be binary. Given a sequence of observation traces $\{X^{1:T}_{1:D_t}\}$, and the model parameters, the inference problem is to find $\{A^{1:T}\}$. A Gibbs sampling algorithm was proposed to carry out this task which can handle a dozen of variables (e.g., 10-20). In this extended abstract we show that for certain htERGM subfamilies, the inference problem can be approximated via an efficiently-solved optimization problem. Specifically, we keep the emission model as defined in [1] and restrict the transition to only capture density and stability features $\Psi = (\Psi_1, \Psi_2)$. This setting is not restrictive and captures the set of evolving pairwise binary Markov random field (MRF) (for example, time varying Icing-model, time-varying regulatory network, etc.). Below we provide two extensions of the technique proposed in [2] for learning the structure of binary MRF. The basic idea in [2] is to regress each node, i, in the graph using an $l1$-regularized logistic regression over the remaining nodes, $-i$. This involves solving N optimization problems each of which results in a coefficient β_i of length N whose non-zero components define the neighborhood structure of node i.

Unconstraint Optimization Formulation

In this formulation we still regress each node using an $l1$-regularized logistic regression over the remaining nodes, $-i$, however, we vary the regression coefficient over time, $\{\beta^{1:T}_i\}$, and to account for stability constraints, we enforce an $L2-$penalty over successive coefficients[1]. Putting everything together, for each node i in the graph,we solve the following optimization problem:

$$\mathrm{Min} \underbrace{\sum_{t=1}^{T}\Big(\frac{1}{D_t}\sum_{d=1}^{D_t}\big[\log(1+\exp(\beta^t_i x^t_{d,-i})) - x^t_{d,i}\beta^t_i x^t_{d,-i}\big] + \theta_1 \parallel \beta^t_{-i} \parallel_1\Big)}_{(3-a)} + \underbrace{\sum_{t=2}^{T}\theta_2 \parallel \beta^t_i - \beta^{t-1}_i \parallel_2}_{(3-b)} \quad (3)$$

[1] It is also possible to use L1-penalty instead — it is not clear to us which one is better at the moment

Where (3-a) is the time-specific $l1$-regularized logistic regression, and (3-b) encodes edge stability. $x^t_{d,-i}$ is data point x^t_d with the i^{th} component replaced with 1, and β^t_{-i} is β^t_i with the i^{th} component removed (i.e. β^t_i acts as the bias term [2]). Each optimization problem has NT variables and results in a set of *smooth*, *sparse* time-specific regression coefficients $\{\beta^{1:T}_i\}$ for each node that when combined, define the neighborhood structure at each time step [2].The smoothness and sparsity are controlled via θ_1, θ_2.

Integer Programming Formulation

One problem with the previous formulation is that it ignores the role of Λ, the *shared* emission parameters [1][2]. In fact Λ_{ij} measures the correlation between nodes i and j and thus plays a similar role as a single β^t_{ij}, thus the solution provided by the above formulation is approximate at best. To account for that, we replace $\beta^{1:T}_i$ in (3) with the time-invariant correlations vector $\lambda_i = \Lambda'_{i.}$ and explicitly model the existence of an edge between node i and j at time t via the binary variable A^t_{ij}. We also denote the neighborhood of node i at time t via $A^t_i = A^t_{i.}$, putting everything together, for every node i, we solve the following integer optimization:

$$\text{Min} \underbrace{\sum_{t=1}^{T} \Big(\frac{1}{D_t} \sum_{d=1}^{D_t} \big[\log(1+\exp(A^t_i y^t_{d,-i})) - x^t_{d,i} A^t_i y^t_{d,-i} \big] + \theta_1 \parallel A^t_{-i} \parallel_1 \Big)}_{(4-a)} + \underbrace{\sum_{t=2}^{T} \theta_2 \parallel A^t_i - A^{t-1}_i \parallel_2}_{(4-b)} \quad (4)$$

$$\text{Subject to} \quad A^t_i \in \{0,1\}^N \quad \forall\, t \quad (5)$$

where $y^t_{d,-i} = x^t_{d,-i} \times \lambda_i$ and '$\times$' denotes component-wise multiplication. Intuitively, we factored β^t_i from (3) into two parts: one real part that gives the coefficients vector λ_i,which is fixed, and another binary part that controls whether a given coefficient is in or out, A^t_i, which becomes the variable in the optimization in (4). We then folded the fixed part, λ_i, with the fixed covariates with respect to node i, $x^t_{d,-i}$, to obtain $y^t_{d,-i}$ just for the sake of readability. It is quite straightforward to note the equivalence of (4-a,b) to (3-a,b). In order to solve the above problem efficiently, we relax the integrality of $\{A^{1:T}_i\}$ and constrain them to be between 0 and 1. Solving the above optimization problem results in real values for all edges that can be later thresholded to recover the the network at all time points. It is interesting to note that even though we relaxed $\{A^{1:T}_i\}$, the *l1*-penalty should force most of them to be zero and hopefully the remaining variables would be moved towards 1. Clearly this formulation is more tight, and slightly expensive than the one in (3).

Current Status and Future Work

We implemented the optimization problem in (3) in matlab using the CVX generic optimization package[3]. On a standard desktop, we were able to solve problems with 100-300 nodes and 66 time points efficiently (we used the Drosophila network in [1] that contains up to 4000 genes and 66 time points). The time to solve each optimization problem in (3) varies from 1.5 minutes (100 variables) to 12 minutes (300 variables), which is compared to days when using Gibs sampling over 20 variables. We believe that we can scale the model up to 1000 variables via moving to a C-environment. Moreover, the CVX solver is a generic convex solver, thus the structure of the problem is not utilized, in the future we intend to extend the interior-point method in [3] to account for the time-dependent constraints (part 4-b). We are still in the process of implementing the relaxed integer-programming formulation. In the final NIPS poster, we will provide a comparison of these two formulations, along with Gibbs sampling, in terms of the trade-off between the precision achieved and the computational resources used. It should be noted that the result of the optimization problem in either (3) or (4) can be used as an approximation to the posterior mode, or can be used to initialize the Gibbs sampler so that the sampler would not wander over states with low-probabilities and quickly enhances over the obtained approximate solution — we are investigating this direction as well.

References

[1] F. Guo, S. Hanneke, W. Fu and E. P. Xing, Recovering Temporally Rewiring Networks: A model-based approach, Proceedings of the 24th International Conference on Machine Learning, ICML 2007

[2] M. Wainwright, P. Ravikumar, and J. Lafferty, High dimensional graphical model selection using l1-regularized logistic regression. In Neural Information Processing Systems, 2006.

[3] K. Koh, S. Kim, and S. Boyd. An interior-point method for large-scale l1-regularized logistic regression. Journal of Machine learning research, 2007.

[2]We followed [1], but Λ can be time-specific and (4) can be generalized accordingly

[3]http://www.stanford.edu/ boyd/cvx/