Project Preparation: TESLA algorithm

Manuel Llamas, ECS Department, University of Southampton

***Abstract*—These instructions give you guidelines for preparing papers for IEEE TRANSACTIONS and JOURNALS. Use this document as a template if you are using Microsoft *Word* 6.0 or later. Otherwise, use this document as an instruction set. The electronic file of your paper will be formatted further at IEEE. Define all symbols used in the abstract. Do not cite references in the abstract. Do not delete the blank line immediately above the abstract; it sets the footnote at the bottom of this column.**

***Index Terms*—About four key words or phrases in alphabetical order, separated by commas. For a list of suggested keywords, send a blank e-mail to keywords@ieee.org or visit** http://www.ieee.org/organizations/pubs/ani_prod/keywrd98.txt

I. INTRODUCTION

In many problems it is helpful and even necessary to study the relationships between entities of a system since we find them interconnected building a network in which some small changes on an element can propagate and affect the entire system. We find a large list of examples in which it is necessary to apply network theory on different and wide subjects as social science, finance, biology or climatology.

Studying such networks can reveal lots of information like how an event can trigger a topological change of the entire network, how entities of the network can depend on each other or share similar properties and organize themselves into groups or which individuals are the main agents in the network ruling its evolution.

Classically these analysis have been carried in the simplest way assuming they are static however we truly know that this is false in most of the cases. The internet, biological processes in a body or the economy are some examples in which networks change and evolve over time shrinking or creating new links as the attributes or states of the individuals change.

Although there is wide literature and methods on modelling static networks (previous references in static networks) it has not been until recent years that interest has grown on studying dynamic or time-varying networks (cite main references).

Usually, only time series measurements, such as microarray, stock price, etc., of the activity of the nodal entities, but not their linkage status, are available. The goal is to recover the latent time-varying networks with temporal resolution up to every single time point based on time series measurements.

To do so this masters' project is going to focus on a Machine Learning algorithm called TESLA, built to recover the structure of time-varying networks over a fixed set of nodes from these time series nodal attributes (It is obvious that in these networks an edge or link between two nodes implies dependency between their nodal states), aiming to recover finance network structures.

To be able to understand the current state of the research on the field and the strengths, weakness and challenges of the techniques proposed some literature review has been carried out.

II. METHODS

A. htERGM model

A classical model of static network analysis is the exponential random graph model (ERGM) (Guo F, Hanneke S, Fu W, Xing E (2007)) that was used as a base for two of the first algorithms proposed to study dynamic networks, the temporal exponential random graph model (tERGM) and the hidden tERGM for modelling a sequence of node attribute observations. When tERGM assume that the sequence of networks is available, htERGM explores the possible dependencies of unobserved rewiring networks and leads to the algorithm that can reconstruct such networks from a snapshots' sequence of nodal attributes.

Although this last algorithm overcame approaches that recover single time-invariant networks it is not the most useful one since it depends on unobserved network variables being unable to compute likelihood ratios, needing inference algorithms specifically for each problem.

B. HMDBM

Another classical approach is using Dynamic Bayesian Networks (DBN) that have been widely used to recover gene regulatory relationships from time-series data in computational systems biology. Its standard assumption is 'stationarity', and therefore, several research efforts have been recently proposed to relax this restriction. However, those methods suffer from three challenges: long running time, low accuracy and reliance on parameter settings. [Zou & Wang 2015] propose a novel non-stationary DBN model by extending each hidden node of Hidden Markov Model into a DBN (called HMDBN), which properly handles the underlying time-evolving networks resulting in a promising experimental evaluation of the method, demonstrating more stably high prediction accuracy and significantly improved computation efficiency (even with no prior knowledge and parameter settings) on both synthetic and real biological data.

Although this probabilistic model is a lot more complex than the TESLA algorithm it could be a good method to implement in order to compare results and performance.

C. *Bayesian non-parametric model*

In this line Durante & Dunson 2014 proposed a Bayesian non-parametric model including time-varying predictors in dynamic network inference precisely for financial studies.

$$\pi_{ij}(t) = \frac{1}{1 + e^{-s_{ij}(t)}}$$

$$s_{ij}(t) = \mu(t) + z_{ij,t}^{T}\beta(t) + x_i(t)^{T}x_j(t)$$

This model computes edge specific predictors where the link probabilities ($s_{ij}(t)$) are estimated via a logistic regression, with $\mu(t)$ a baseline process quantifying the overall propensity to form links in the network across time, $x_i(t)^T x_j(t)$ are vectors containing the latent coordinates favouring a higher link probability when units *i* and *j* have latent coordinates in the same direction and $z_{ij,t} = [z_{ij1,t}, \dots, z_{ijP,t}]^T$ is a P-dimensional vector of time-varying edge-specific predictors for units *i* and *j* at time t and $\beta(t)$ are the corresponding dynamic coefficients. This allows the proximity between units *i* and *j* at time *t* to depend on predictors in a manner that varies smoothly with time.

The main issue regarding this method is that it assumes time-constant smoothness while in finance and other network applications we expect smoothness to vary over time.

D. *TESLA algorithm*

TESLA, from the acronym TESLLOR, which stands for temporally smoothed *l*1-regularized logistic regression represents an extension of the 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 *l*1-regularized logistic regression algorithm for estimating single sparse networks (*Wainwright M, Ravikumar P, Lafferty J (2006) High dimensional graphical model selection using l1-regularized logistic regression. Advances in Neural Information Processing Systems 19 (MIT Press, Cambridge, MA), pp 1465–1472*) it was developed a regression regularization scheme that connects multiple time-specific network inference functions via a first-order edge smoothness function that encourages 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, what means an advantage in contrast of htERGM algorithm.

TESLA estimates $\{\hat{\theta}^t\}_{t=1}^{T}$, the correlation (or dependency strength) matrix between the nodes using a time-series of the observed stated of the nodes $\mathbf{x}^t$.

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

where,

$$\ell(\boldsymbol{x}^t; \boldsymbol{\theta}_i^t) = \sum_{d=1}^{N^t} \left(\log\left(1 + \exp(\boldsymbol{x}_{d,-i}^t \boldsymbol{\theta}_i^t)\right) - \boldsymbol{x}_{d,-i}^t \boldsymbol{\theta}_i^t \boldsymbol{x}_{d,i}^t\right)$$

Where the graph structure is given by the locations of the nonzero elements of the parameter vectors $\boldsymbol{\theta}_i^t$. Components of the vectors $\boldsymbol{\theta}_i^t$ are indexed by distinct pairs of nodes and a component *j* of the vector $\boldsymbol{\theta}_i^t$ is nonzero if and only if the corresponding edge (*i*, *j*) ∈ Eτ. $\boldsymbol{x}_{d,-i}^t$ denotes the observed states of all nodes but node *i* and $\ell()$ is the log conditional likelihood of state $\boldsymbol{x}_{d,i}^t$ under a logistic regression model.

First term of the algorithm are p logistic regressions, one for each node with respect the rest of them ($\boldsymbol{x}_{d,-i}^t$ are the states of all nodes but node I in the dth sample in time epoch t).

The second term is a lasso regularization that introduces sparsity and consistency in neighbourhood selection, as seen in (*Wainwright M, Ravikumar P, Lafferty J (2006) High dimensional graphical model selection using l1-regularized logistic regression. Advances in Neural Information Processing Systems 19 (MIT Press, Cambridge, MA), pp 1465–1472.*). Wainwright et al. have shown that that applying this l1 norm-regularized logistic regression leads to consistent neighbourhood selection, so it converges to the true graph structure. Also a sparse graph effectively limits the degree of freedom of the model, which makes structure recovery possible given a small sample size (*Kolar M., Song L, Ahmed A., Xing E. P. Estimating time-varying networks. Institute of Mathematical Statistics in The Annals of Applied Statistics, 2010, Vol. 4, No. 1, 94–123*)

Since in this problem we are estimating dynamic networks (so more than one single graph), and, as assumed before, temporary adjacent networks are going to be similar one to each other, we need to introduce the third term in the equation that penalises the discrepancy between time-adjacent parameters θ.

Summarizing, the first term is the main one while the last two terms are the penalties ($L^{shrink} + L^{smooth}$) that are introduced to enforce sparsity and smoothness.

E. *TESLA update*

It is important to note that in this specific method we are assuming that $\boldsymbol{\theta}_i^t$ are a piecewise constant function with abrupt changes in parameters (jumps).

However in some cases we need the changes in parameters to be even less abrupt, so further work adapted this algorithm with a weighting term that works with $\boldsymbol{\theta}_i^t$ as real smooth functions instead of a penalty in the structural changes (L^{smooth}) that assumes “jumps” between adjacent values.

$$\hat{\boldsymbol{\theta}}_i^{\tau} = \min_{\theta \in \mathbb{R}^{p-1}} \{\ell(\boldsymbol{\theta}_i; \mathcal{D}_n) + \lambda_1 \|\boldsymbol{\theta}_i^{\tau}\|_1\}$$

$$\ell(\boldsymbol{\theta}_i; \mathcal{D}_n) = -\sum_{t \in \tau_n} \omega_t^{\tau} \gamma(\boldsymbol{\theta}_i; \boldsymbol{x}^t)$$

Where the weights are defined by

$$\ell(\boldsymbol{\theta}_i;\ \mathcal{D}_n) = -\sum_{t\in\tau_n} \omega_t^\tau\, \gamma(\boldsymbol{\theta}_i; \boldsymbol{x}^t)$$

And $K_h()$ is a symmetric nonnegative kernel function.

III. Math[1]

If you are using *Word*, use either the Microsoft Equation Editor or the *MathType* add-on (http://www.mathtype.com) for equations in your paper (Insert | Object | Create New | Microsoft Equation *or* MathType Equation). “Float over text” should *not* be selected.

IV. Conclusion

Summarizing we find crucial studying data through a network lens since it can add substantial new insights. This work evaluated the state-of-the-art methods in the dynamic networks field in order to prepare further research of applying (mainly) TESLA algorithm in a finance context to study the changes that may warn or even drive significant global events like the 2008 crisis.

It is important to use scalable computational methods that can fit very large networks (huge number of nodes), so models like TESTLA that exploit sparsity are promising in this matter.

Reviewing the two TESLA-based algorithms this project is going to be focused on we find they represent two different ends: one is tailored toward estimation of structural changes in the model while the other is able to estimate smoothly changing networks. This gives us a great opportunity to study the same field with these two different approaches and test which one better models the stocks’ relationships.

.

References

[1] Guo F, Hanneke S, Fu W and 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).

[2] Zhu S. and Wang Y. (2015). Hidden Markov induced Dynamic Bayesian Network for recovering time evolving gene regulatory networks. *Scientific Reports*, *5*, 17841.

[3] Durante D. and Dunson D. (2014) Bayesian dynamic financial networks with time-varying predictors. *Statistics & Probability Letters*, 93, 19-26

[4] Ahmed A and Xing E. Recovering time-varying networks of dependencies in social and biological studies. *PNAS* (2009) 106 (29) 11878-11883

[5] Kolar M, Song L, Ahmed A and Xing, Eric P. Estimating time-varying networks. *Annals Applied Statistics Vol. 4* (2010). No. 1, 94-123