Zero Augmented Distributions with Score Driven Parameters for Dynamical Sparse Weighted Networks (With Applications to Systemic Risk Forecasting ?)

Domenico Di Gangi

December 9, 2019

1 Recent Comments

- consider alternative choice for the distribution of the weights: log normal
- Model checking a posteriory for the goodness of fit of the distribution chosen for the weights
- Would shifting and censoring allow for a test of misspecification? if not that would be an argument in favour of zero augmentation
- Try to fit the WTN and, if the data allow it , estimate the model with link specific regressors or a constant term that depends on the geographical distance between two countries $e^{\varphi_i + \varphi_j + c_{ij} or c d_{ij}}$
- Consider also the possibility of adding node specific regressors in the score driven update equation, e.g. the libor interest rate for eMid data

2 Planned Paper Outline

We focus on models for **Dynamical Sparse Weighted Networks** described by sequences of matrices of positive real numbers $\left\{Y_{ij}^{(t)}\right\}_{t=1}^{T}$.

- Discuss Zero Augmented distributions to define a framework for a flexible description of **Sparse Weighted Networks**, already at the static level:

$$P\left(Y_{ij} = y\right) = \begin{cases} \left(1 - \pi_{ij}\right) & for \quad y = 0 \\ \\ \pi_{ij}\; g\left(y | \mu_{ij}, \lambda_{ij}\right) & for \quad y > 0 \end{cases}, \tag{1}$$

  where $0 < \pi_{ij} < 1$ and $g\left(y\right)$ is a gamma distribution, with link specific parameters μ_{ij} and λ_{ij}.

  Motivation :

  - The main motivation for the ZA approach would be the possibility to decouple the probability of a link's presence from the distribution of it's weight. This seems to allow a greater flexibility compared to Tobit models. Flexibility that we intend to fully exploit in the **Dynamical** context.

– The simplest choice for g would be the exponential. We consider the gamma because it allows to adjust the dispersion around the mean . Moreover, with the gamma, our approach includes, as a limiting case, the method of **?**. This is interesting, since the latter has been found to be a good solution in reconstructing financial networks (see the wide comparative study of **?**). Additionally a recent maximum entropy model (cite the latest paper from Lucca that proposes once more an ensemble "physically" motivated) results in a ZA exponential. [1]

- Discuss the **Dynamical** modeling and **propose, test and apply our Score Driven approach**.

  – The idea is to model $P\left(Y^{(t)}_{ij}|\mathbf{Y}^{(t-1)},\dots\mathbf{Y}^{(1)}\right)$. Even if we consider dependencies at one lag, the problem is extremely high dimensional. To cope with the issue of dimensionality, different approaches have been considered in the literature:

    * Driven by system specific insights, the researcher selects a set of network statistics $G_i\left(\mathbf{Y}\right)$ and estimates the dependency of links at time t on $G\left(\mathbf{Y}^{(t-1)}\right)$. **?**, for example, estimate a Tobit model with few regressors, for each link. Moreover they use a local-likelihood method to estimate time varying coefficients of the regression.
    * Latent space models, where a set of parameters is associated to each node and an exogenous time evolution is assumed for those parameters, e.g. **?** define a latent space Tobit
    * Models that allow each one of the matrix elements $Y^{(t)}_{ij}$ to depend on each of the ${Y_{ij}}^{(t-1)}$ have also been considered in the literature:
      1. **?** Estimate a tensor regression (very similar to a VAR on $vec(\mathbf{Y})$), with rank restrictions on the (huge) matrix of model's parameters . (Not clear how they take sparsity into account)
      2. **?** Consider a penalized logistic auto-regression model for binary networks (basically a logistic regression for each link, using all lagged matrix elements, and also products, with a lasso penalization). The same approach can in principle be extended to sparse weighted networks, and in the ZA framework.

  – **We propose** to start from a static model (in principle one that fits well the cross sectional data) and define its dynamical version using the SD approach. Applying this to the ZA-Gamma model results in the **Zero Augmented Score Driven (Gamma?) Network Model ZA-SD-Nets (or maybe weighted random graphs ?ZA-SD-WRG)**

    * It "decouples" the modeling of the weights, and their dynamics, from that of the links' presence. Basically we can use it in combination with any model for dynamical binary networks, e.g. constant (uniform or link specific) probabilities, ERGMs, SD-ERGM, neural networks based models[2] Ideally we can evaluate the best performing model for link prediction in binary networks (as discussed for example in **?**), and combine it with our score driven approach for the dynamics of the weights.

[1] More alternatives considered in the literature:

1. **?** proposes a Sparse Latent Position Model (SLPM), modifies the original LPM to account for nonnegative weighted edges using finite mixtures of exponential distributions. (Only Static)
2. A number of papers that model static weighted networks and do not take into account sparsity.
3. Models for dynamical weighted networks, cited in the following.

[2] Sadly I have completely ignored this stream of literature so far (see, for example **?????**).

* It can be used as DGP to simulate realistic (in what sense?) dynamical sparse weighted networks
* It can be easily estimated via maximum likelihood (show it by simulating it as a DGP and estimating the static parameters) and can be used as a filter in case of misspecified dynamics (by filtering known DGPs for the time varying parameters or simulating a completely different network model and using it for forecast on synthetic data?)
* When estimated on data, copes with the challenge of dimensionality and allows to forecast links.
* As final application, and proof of flexibility, we can show how to use the framework for forecast of systemic risk from partial information.
* It allows also to include exogenous regressors or, more interestingly, features that can be extracted using deep learning techniques for dimensionality reduction.

3 Done so far

3.1 ZA Distribution for Sparse Weighted Networks

We need to choose the probability of observing a link $\Theta\left(Y_{ij}^{(t)}\right)$ and the probability to observe a specific weight $Y_{ij}^{(t)}$. Without specifying one particular temporal evolution for the probabilities of observing a link $\pi_{ij}^{(t)}$, the ZA-SD-Nets is

$$P\left(Y_{ij}^{(t)}=y\right)=\begin{cases}\left(1-\pi_{ij}^{(t)}\right) & for \quad y=0\\ \\ \pi_{ij}^{(t)}\frac{\left(\mu_{ij}^{(t)}\right)^{-\alpha}y^{(\alpha-1)}}{\Gamma(\alpha)}e^{-\frac{y}{\mu_{ij}^{(t)}}} & for \quad y>0\end{cases} \tag{2}$$

Choosing to associate two parameters $(\overleftarrow{\varphi}_i, \overrightarrow{\varphi}_i)$ to each node i, we consider two different relations between the latters and matrix $\boldsymbol{\mu}$:

$$\text{POX A} \quad \mu_{ij}^{(t)}=\frac{1}{\alpha}\frac{e^{\left(\overleftarrow{\varphi}_i^{(t)}+\overrightarrow{\varphi}_j^{(t)}\right)}}{\pi_{ij}^{(t)}}, \qquad E\left[Y_{ij}^{(t)}\right]=e^{\left(\overleftarrow{\varphi}_i^{(t)}+\overrightarrow{\varphi}_j^{(t)}\right)}, \quad E\left[Y_{ij}^{(t)}|y>0\right]=\frac{e^{\left(\overleftarrow{\varphi}_i^{(t)}+\overrightarrow{\varphi}_j^{(t)}\right)}}{\pi_{ij}^{(t)}} \tag{3}$$

$$\text{POX B} \quad \mu_{ij}^{(t)}=\frac{1}{\alpha}e^{\left(\overleftarrow{\varphi}_i^{(t)}+\overrightarrow{\varphi}_j^{(t)}\right)} \quad, E\left[Y_{ij}^{(t)}\right]=\pi_{ij}^{(t)}e^{\left(\overleftarrow{\varphi}_i^{(t)}+\overrightarrow{\varphi}_j^{(t)}\right)}, \qquad E\left[Y_{ij}^{(t)}|y>0\right]=e^{\left(\overleftarrow{\varphi}_i^{(t)}+\overrightarrow{\varphi}_j^{(t)}\right)} \tag{4}$$

while the choice of the exponential link between μ and φ allows the latters to be unbounded [3], it is not yet clear what should we use between pox A and B.

In both cases however the model is well defined whatever the $\pi_{ij}^{(t)}$ are. Those can be obtained by any model for binary networks and then plugged into the previous definition. We describe the binary probabilities in terms of positive parameters p_{ij} defined by $\pi_{ij}=\frac{1}{1+p_{ij}}$ The loglikelihood for a single observation, omitting the temporal dependency, is

$$\log P\left(\mathbf{Y}|\mathbf{p},\boldsymbol{\mu}\right)=\sum_{ij}-\log\left(1+p_{ij}\right)+\Theta\left(y_{ij}\right)\left[\log p_{ij}+(\alpha-1)\log y_{ij}-\log\Gamma\left(\alpha\right)-\alpha\log\mu_{ij}-\frac{y_{ij}}{\mu_{ij}}\right] \tag{5}$$

[3] That is very convenient since we are going to turn them into dynamical ones, and to keep the dynamics into bounded regions turns out to be complicated in this setting.

and the score can be

$$\text{Pox A} \quad \frac{\partial \log P}{\partial \overleftarrow{\varphi}_i} = \sum_j \left(-\alpha \Theta\left(y_{ij}\right) + y_{ij} \pi_{ij} \frac{1}{\alpha} e^{-\left(\overleftarrow{\varphi}_i + \overrightarrow{\varphi}_j\right)} \right) \quad \rightarrow \quad s = s\left(\varphi, \mathbf{Y}^{(t)}, \mathbf{p}, \alpha\right) \tag{6}$$

$$\text{Pox B} \quad \frac{\partial \log P}{\partial \overleftarrow{\varphi}_i} = \sum_j \left(-\alpha \Theta\left(y_{ij}\right) + y_{ij} \frac{1}{\alpha} e^{-\left(\overleftarrow{\varphi}_i + \overrightarrow{\varphi}_j\right)} \right) \quad \rightarrow \quad s = s\left(\varphi, \mathbf{Y}^{(t)}, \alpha\right) \tag{7}$$

in both cases

$$\frac{\partial \log P}{\partial \varphi_i} = \sum_j \frac{1}{\alpha} \left(\frac{y_{ij}}{E\left[y_{ij} | y_{ij} > 0\right]} - \alpha^2 K_i \right)$$

and

$$\overleftarrow{\varphi}_i^{(t+1)} = \overleftarrow{w}_i + \overleftarrow{B}_i \overleftarrow{\varphi}_i^{(t)} + \overleftarrow{A}_i \frac{\partial \log P}{\partial \overleftarrow{\varphi}_i}$$

Pox A is convenient because we can probably prove that it is equal to that of **?** in the limit of zero variance, while in pox B the dynamics of the φ parameters is completely decoupled by that ot p, because the score wrt to the formers does not depend on the latters.

For the moment, the code is written for pox B, with $\alpha = 1$, hence exponential.[4]

3.1.1 Progress

I am writing the code in python at `https://github.com/domenicodigangi/DynWNets`. Working code:

- Regarding the static version I tested that sampling and estimating (2), with μ_{ij} given by the static version of POX B, the parameters seem unbiased.
- I coded Pox B, hence only the DGP depends on $\pi_{ij}^{(t)}$ (that I assumed constant and uniform for the initial tests)
- I can successfully sample the DGP, estimate the single snapshot models, compute the temporal averages of the SS estimates for $(\overleftarrow{\varphi}_i, \overrightarrow{\varphi}_i)$ to target their unconditional mean. I coded Pox B, hence the estimates are independent from $\pi_{ij}^{(t)}$
- First positive tests on the, targeted, estimate of the ZA-SD-Net model, sampled from a DGP with uniform and constant $\pi_{ij}^{(t)}$

4 notes and next steps

4.1 Note

Anche se molto in ritardo rispetto ai miei piani iniziali, mi sembra di avere un per un contributo . Penso sia un buon momento per un confronto/ feedback.

4.2 Forecasting Sparse weighted Networks

For the purposes of links' forecasting, we have 3 alternatives :

1. forecast $Y_{ij}^{(t)}$ using the expected value of $Y_{ij}^{(t-1)}$ (forecast a dense, if not always fully connected network)

[4] Note that numpy samples $p(x) = x^{k-1} \frac{e^{-x/\theta}}{\theta^k \Gamma(k)}$.

2. repeated sample of the ZA distribution

3. (a) use the SD update to compute $\varphi^{(t)}$ (using only observations at $t-1$)
   (b) use the forecast of probabilities $p_{ij}^{(t)}$ of observing a single link to sample a binary matrix
   (c) compute the expected strengths sequence and distribute the weights using RAS (criterium needed for the extra weights)

The last method might allow us to take into account the fact that often banks split their weights in simple fractions (e.g. one third and two thirds): potremmo verificare quaòli banche nel passato mostrano propensione a divisione in frazioni ben definite e dare prioritá a queste nell'allocare i pesi. alloco prima i le piú grandi con maggiore propensione a integer division e poi il resto.