10-708 06/06/2020

L9-GGMs, Ising models review

(*) matrix-inverse lemma

- for a block-partitioned matrix $\underline{M} = \begin{pmatrix} \underline{E} & \underline{F} \\ \underline{G} & \underline{H} \end{pmatrix}$ with $\underline{E}$ and $\underline{H}$ invertible.

- matrix-inverse lemma:-

$$(\underline{E} - \underline{F}\underline{H}^{-1}\underline{G})^{-1} = \underline{E}^{-1} + \underline{E}^{-1}\underline{F}(\underline{H} - \underline{G}\underline{E}^{-1}\underline{F})^{-1}\underline{G}\underline{E}^{-1}$$

method:

- diagonalise $\underline{M}$:-

$$\begin{bmatrix} I & -FH^{-1} \\ 0 & I \end{bmatrix}\begin{bmatrix} E & F \\ G & H \end{bmatrix}\begin{bmatrix} I & 0 \\ -H^{-1}G & I \end{bmatrix} = \begin{bmatrix} E-FH^{-1}G & 0 \\ 0 & H \end{bmatrix}$$

- schur's complement: (notable in numerical analysis)

$$M/H = E - FH^{-1}G$$

- Note: for $XYZ = W$, $Y^{-1} = ZW^{-1}X$

$$M^{-1} = \begin{bmatrix} E & F \\ G & H \end{bmatrix}^{-1} = \begin{bmatrix} I & 0 \\ -H^{-1}G & I \end{bmatrix}\begin{bmatrix} E-FH^{-1}G & 0 \\ 0 & H \end{bmatrix}^{-1}\begin{bmatrix} I & -FH^{-1} \\ 0 & I \end{bmatrix}$$

$$= \begin{bmatrix} I & 0 \\ -H^{-1}G & I \end{bmatrix}\begin{bmatrix} (M/H)^{-1} & 0 \\ 0 & H^{-1} \end{bmatrix}\begin{bmatrix} I & -FH^{-1} \\ 0 & I \end{bmatrix}$$

$$= \begin{bmatrix} (M/H)^{-1} & 0 \\ -H^{-1}G(M/H)^{-1} & H^{-1} \end{bmatrix}\begin{bmatrix} I & -FH^{-1} \\ 0 & I \end{bmatrix}$$

$$= \begin{bmatrix} (M/H)^{-1} & -(M/H)^{-1}FH^{-1} \\ -H^{-1}G(M/H)^{-1} & H^{-1}G(M/H)^{-1}FH^{-1} + H^{-1} \end{bmatrix}$$

- ~~Applying M.I.L:~~

$$\underline{M}^{-1} = \begin{bmatrix} E^{-1} + E^{-1}F(M/E)^{-1}GE^{-1} & -E^{-1}F(M/E)^{-1} \\ -(M/E)^{-1}GE^{-1} & (M/E)^{-1} \end{bmatrix} \quad (I)$$

- see Murphy (2012) s.4.3.4.1
- (I) is achieved by decomposing in terms of $\underline{E}$ and
  $M/E = (H - GE^{-1}F)$
- we get MIL as a corollary

(*) Covariance and precision matrices

$$\underline{\Sigma} = \begin{bmatrix} \sigma_{11} & \underline{\sigma}_1^T \\ \underline{\sigma}_1 & \underline{\Sigma}_{-1} \end{bmatrix}$$

σ_{11} - scaler
$\underline{\sigma}_1, \underline{\sigma}_1^T \in \mathbb{R}^{N-1}$
$\underline{\Sigma}_{-1} \in \mathbb{R}^{(N-1)\times(N-1)}$

$$\underline{X} = \begin{bmatrix} X_i \\ | \\ \underline{X}_{-i} \\ | \end{bmatrix} \qquad \underline{X} \in \mathbb{R}^N$$

$$\underline{M}^{-1} = \begin{bmatrix} (M/H)^{-1} & -(M/H)^{-1}FH^{-1} \\ -H^{-1}G(M/H)^{-1} & H^{-1} + H^{-1}G(M/H)^{-1}FH^{-1} \end{bmatrix}$$

- $\underline{Q} = \underline{\Sigma}^{-1}$

$$(M/H)^{-1} = \underline{\sigma}_{11}^{-1} + \underline{\sigma}_{11}^{-1}\underline{\sigma}_1^T(\underline{\Sigma}_{-1} - \underline{\sigma}'\underline{\sigma}_{11}^{-1}\underline{\sigma}_1^T)^{-1}\underline{\sigma}_1\underline{\sigma}_{11} = q_{11}$$

$$\underline{Q} = \begin{bmatrix} q_{11} & -q_{11}\underline{\sigma}_1^T\underline{\Sigma}_{-1}^{-1} \\ -q_{11}\underline{\Sigma}^{-1}\underline{\sigma}_1 & \underline{\Sigma}_1^{-1}(I + q_{11}\underline{\sigma}_1\underline{\sigma}_1^T\underline{\Sigma}_{-1}^{-1}) \end{bmatrix} = \begin{bmatrix} q_{11} & \underline{q}_1^T \\ \underline{q}_1 & \underline{Q}_{-1} \end{bmatrix}$$

(*) Single node conditional

- Partition $\underline{X} = \begin{bmatrix} X_i \\ \underline{X}_{-i} \end{bmatrix}$

$$p\left(\begin{bmatrix} X_i \\ \underline{X}_{-i} \end{bmatrix} \middle| \underline{\mu}, \underline{\Sigma}\right) = N\left(\begin{bmatrix} X_i \\ \underline{X}_{-i} \end{bmatrix} \middle| \begin{bmatrix} \mu_i \\ \underline{\mu}_{\underline{X}_{-i}} \end{bmatrix}, \begin{bmatrix} \Sigma_{X_iX_i} & \Sigma_{X_i\underline{X}_{-i}} \\ \Sigma_{\underline{X}_{-i}X_i} & \Sigma_{\underline{X}_{-i}\underline{X}_{-i}} \end{bmatrix}\right)$$

Then we have:

$$p(x_i|\underline{x}_{-i}) = N(x_i \,|\, \underline{m}_{i,-i}, \underline{V}_{i,-i})$$

- apply cond. Gaussian.

$$\underline{m}_{i,-i} = \mu_i + \underline{\Sigma}_{x_i\underline{x}_{-i}} \underline{\Sigma}^{-1}_{\underline{x}_{-i}\underline{x}_{-i}} (\underline{x}_{-i} - \mu_{\underline{x}_{-i}})$$

$$\underline{V}_{i,-i} = \underline{\Sigma}_{x_ix_i} - \underline{\Sigma}_{x_i\underline{x}_{-i}} \underline{\Sigma}^{-1}_{\underline{x}_{-i}x_i} \underline{\Sigma}_{\underline{x}_{-i}x_i}$$

✓

- set $\mu_i = 0$

$$\Rightarrow p(x_i|\underline{x}_{-i}) = N(\Sigma_{x_i\underline{x}_{-i}} + \underline{\Sigma}^{-1}_{\underline{x}_{-i}\underline{x}_{-i}} \underline{x}_{-i} \,|\, \Sigma_{x_ix_i} - \underline{\Sigma}_{x_i\underline{x}_{-i}} \underline{\Sigma}^{-1}_{\underline{x}_{-i}\underline{x}_{-i}} \underline{\Sigma}_{\underline{x}_{-i}x_i})$$ ✓

$$= N(\underline{\sigma}_i^T \underline{\Sigma}^{-1}_{-i} \underline{x}_{-i}, q_{i|-i})$$ ✓ ?

0/52

$$= N\left(\frac{\underline{q}_i^T}{-q_{ii}} \underline{x}_{-i}, q_{i|-i}\right)$$ ✓ ?

(*) $p(x_i|\underline{x}_{-i})$ is a 1-D Gaussian distr. (univariate)!

$\Rightarrow$ mean, covariance i.e. $\frac{\underline{q}_i^T}{-q_{ii}} \underline{x}_{-i}$, $q_{i|-i}$ are both scalars

(*) Amended lecture notes; clearer now what is going on.

- conduct N conditional autoregressions (using L_1-regularisation LASSO) to of x_i against $\underline{x}_{-i}$.
- Each autoregression will yield an estimate $\frac{\hat{\underline{q}}_i^T}{-q_{ii}}$, which can be used to populate the precision matrix $\underline{Q}$ column by column (and as it is symmetric also)

As we are focusing on presence/absence of non-zero entries rather than value in itself

(*) Note that we do not go for straightforward inversion of $\underline{\Sigma}$ to get precision $\underline{Q}$

- we attempt to estimate $\underline{Q}$ directly (or columns of it) by exploiting one-to-one correspondence of $\underline{Q}$ (in terms of sparseness) with param. estimates of N autoregressions

(*) Achieved with a sparsity constraint

(*) structure learning ⟶ focus on sparseness not value of components of precision. $s_i = \{j : j \neq i, \theta_{ij} \neq 0\}$

(*) known as graphical LASSO

(*) This is the Meinhausen-Buhlman algorithm

(*) Two alternatives:- (Friedman, Hastie, Tibshirani 2007)

1) - Directly estimate Q

- compute sample variance covariance matrix.
- use L1 reg. ML estimation.

Evolving Social Networks

- Inference I

- time-evolving graphs are dependent
- small subset of nodes/edges modified between each time point
- Find an algorithm to estimate time-specific graph using entire dataset of graphs evolving through time (one for each time point?)

KELLER - cyclically run graph algorithm that performs neighbourhood selection. (

- similar loss to earlier

$$\hat{\theta}_i^t = \underset{\theta_i^t}{\operatorname{argmin}}\ l_w(\theta_i^t) + \lambda_1 \|\theta_i^t\|_1 \qquad \forall t$$

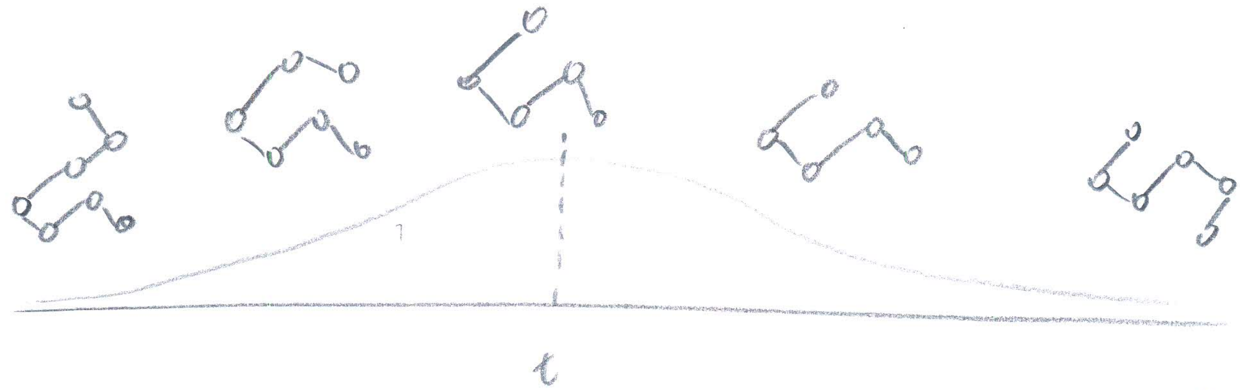
$$l_w(\theta_i^{(t)}) = \sum_{t'=1}^{T} w(\underline{x}^{t'}; \underline{x}^t) \log P(x_i^{t'} | \underline{x}_{-i}^{t'}, \theta_i^t)$$

(*) Summing over time-series of vertex examples

- weight function defined over two time points $w(\underline{x}^{t'}; \underline{x}^t)$
- considers 'distance' between network at two time points.
- (kernel?)
- use this to do a weighted sum of condit. likelihoods of a node of interest x_i given the rest of nodes. (over time points)

(*) Intuition:

- For graph of interest at time t, and you have one example at that time.
- Acknowledge that other examples of graphs not from time t may be relevant (due to dependence) to graph at time t.
- The closer the graph at another time t' is to graph of interest, the more relevant info this may contain.



- Introducing this weight function intuitively allows for use of entire dataset at every time point (some kind of temporal-based proximity measure aka kernel)
- An example of a non-parametric neighbourhood selection
- Recall → kernel density estimation ~~by~~ weights samples according to a 'centre of gravity'
- See Kolar, Le Song, Ahmed, Xing (2010)
- Nonparametric neighbourhood selection
- cond. like

$$P_{\theta^t}(x_i^t \mid x_{\setminus i}^t) = \text{logistic}(2x_i^t \langle \theta_{\setminus i}^t, x_{\setminus i}^t \rangle)$$

- Neighbourhood sel. $\quad S(x_i) = \{ j \mid \theta_{i,j}^t \neq 0 \}$
- Time specific graph reg.
  - estimate at $t^* \in [0,1]$

$$\min_{\theta \in \mathbb{R}^{p_n - 1}} \left\{ -\sum_{t \in T^n} w_t(t^*) \gamma(\theta_i; x^t) + \lambda_1 \|\theta_i\|_1 \right\}$$

$$\gamma(\theta_i^t; x^t) = \log P_{\theta_i^t}(x_i^t \mid x_{\setminus i}^t)$$

$$w_t(t^*) = \frac{K_{h_n}(t - t^*)}{\sum_{t' \in T^n} K_{h_n}(t' - t^*)}$$

(*) Inference (II)

TESLA: Temporally smoothed l_1 reg. logistic regression

$$\hat{\theta}_i^1, \ldots \hat{\theta}_i^T = \underset{\theta_i^1 \ldots \theta_i^T}{\arg\min} \sum_{t=1}^{T} l_{avg}(\theta_i^t) + \lambda_1 \underbrace{\sum_{t=1}^{T} \|\theta_{-i}^t\|_1}_{(i)} + \lambda_2 \underbrace{\sum_{t=2}^{T} \|\theta_i^t - \theta_i^{t-1}\|_q^q}_{(ii)} \quad (I)$$

$$l_{avg}(\theta_i^t) = \frac{1}{N^t} \sum_{d=1}^{N^t} \log P(x_{d,i}^t \mid \underline{x}_{d,-i}^t, \theta_i^t)$$

- Previous graph algo (KELLER); estimate param for one graph at time point t and summed over all examples.
- TESLA: simultaneously estimate params of graphs simult. from $t=1 \ldots T$.
  - loss - sum of condit. likelihoods of $X_i \mid \underline{X}_{\bar{i}}$ with no reweighting
  - sparsity → every graph has sparse structure (i)
    - ↳ differences between adjacent pair of graph structures in time also sparse (ii)
    - structure is 'smooth' over time;
    - evolution of graph structure between two time points minimally
    - graph structure between two time points minimally distinct

Can rewrite (I) using box constraints (rewrite above constraints)

TESLA:
$$\min_{\substack{\theta_i^1 \ldots \theta_i^T \\ \underline{u}_i^1 \ldots \underline{u}_i^T;\ \underline{v}_i^2 \ldots \underline{v}_i^T}} \sum_{t=1}^{T} l(\underline{x}^t; \theta_i^t) + \lambda_1 \sum_{t=1}^{T} \underline{1} \underline{u}_i^t + \lambda_2 \sum_{t=2} \underline{1} \underline{v}_i^t$$

s.t. $-u_{i,j}^t \leq \theta_{i,j}^t \leq u_{i,j}^t \quad t=1, \ldots, T \quad \forall j \in V \setminus i$

$-v_{i,j}^t \leq \theta_{i,j}^t - \theta_{i,j}^{t-1} \leq v_{i,j}^t \quad t=2, \ldots, T \quad \forall j \in V \setminus i$ (total variation differences)

- Can run different optimisation algorithms
- Consistency guarantees

(*) Theoretical analysis of KELLER, TESLA → see Kolar, Xing (2009)
Kolar, Le Song, Ahmed, Xing (2010)

(*) Applications

- social network time-series data
- senator network ⟶ shows predictive power on chafee's political movements!

- breast cancer ⟶ use network inference

- TV distance is significant if you know dependency structure over networks (through time)

(*) Estimating time varying networks, Kolar, Le Song, Ahmed, Xing (2010)

- paper contains rigorous formulation of timevarying graphical structure estimation of GGM for discrete r.v.s. (temporally smoothed l_1-reg. log. regression)
- Assumptions required on structure of parameter vector.
- Optimisation algorithms presented.
- Hyperparam selection ⟶ (graph sparsity - BIC and a heuristic)
- Real-world datasets ⟶ particularly interesting

voting: - Time varying characteristics of voting bill records 109th congress (eg. senators)
- senators with swaying political stance can be discovered (not possible for time-invariant network struc. estimation)

(*) High-dimensional graphs and variable selection with the LASSO, Meinshausen and Buhlmann (2006)

- Pattern of zero-entries in precision matrix of MVG → C.I. restrictions between variables

- neighbourhood selection via LASSO is computationally attr. alt. to standard covariance selection for sparse high-dimensional graphs

- ≙ equivalent to variable selection for Gaussian linear models.

(*) Gives results on consistency

- consistent estimation of the full-edge set in a sparse high-dim graph is possible (asymptotically, prob. of estimating correct neighbours converges exponentially to 1).

-