Then we have:

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

- apply cond. Gaussian.

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

$$\underline{v}_{i,-i} = \Sigma_{x_i x_i} - \Sigma_{x_i x_{-i}} \Sigma_{x_{-i} x_{-i}}^{-1} \Sigma_{x_{-i} x_i}$$ ✓

- set $\mu_i = 0$

$$\Rightarrow p(x_i | \underline{x}_{-i}) = N\left(\Sigma_{x_i x_{-i}} + \Sigma_{x_{-i} x_{-i}}^{-1} \underline{x}_{-i} \,\Big|\, \Sigma_{x_i x_i} - \Sigma_{x_i x_{-i}} \Sigma_{x_{-i} x_{-i}}^{-1} \Sigma_{x_{-i} x_i} \right)$$ ✓

$$= N\left(\underline{\sigma}_i^T \Sigma_{-i}^{-1} \underline{x}_{-i} , q_i|_{-i} \right)$$ $\ell \, \textcircled{e}$

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

$\boxed{0/s2}$

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

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

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

- Conduct N conditional autoregressions (using L1-regularisation LASSO) to of x_i against $\underline{x}_{-i}$.

- Each autoregression will yield an estimate $\dfrac{\hat{\underline{q}}_i^T}{-q_{ii}}$, 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 Σ to get precision Q

- We attempt to estimate Q directly (or columns of it) by exploiting one-to-one correspondence of 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 L_1 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.

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

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

(✱) Summing our time-series of vertex examples

- weight function defined over two time points $w(x^{t'}; x^t)$
- (kernel?) - considers 'distance' between network at two time points.
- use this to do a weighted sum of condt. 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.
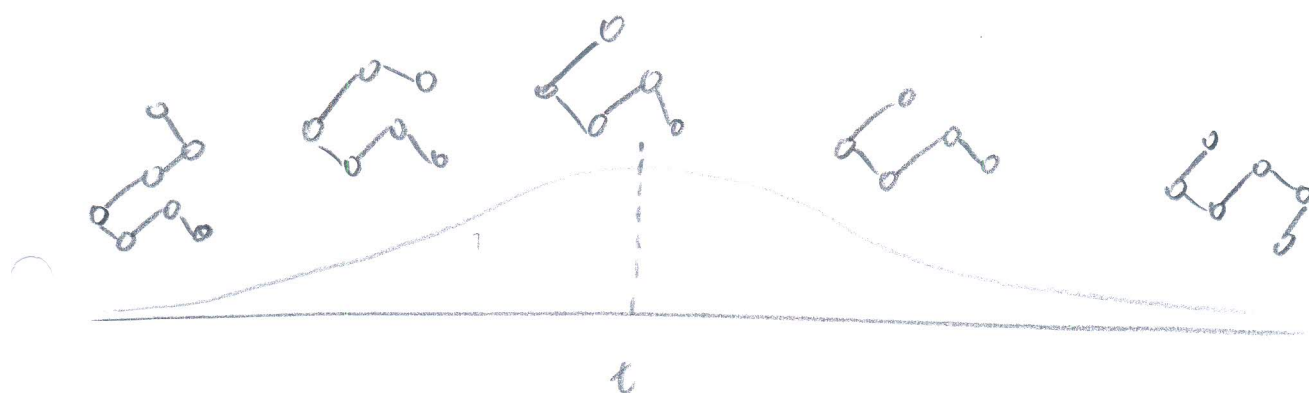
- introducing this weight function intuitively allows for use of <u>entire dataset</u> at <u>every time point</u> (some kind of temporal-based proximity measure aka <u>kernel</u>)

- an example of a non-parametric <u>neighborhood selection</u>
- recall → kernel density estimation by weights samples according to a 'centre of gravity'
- see Kolar, Le Song, Ahmed, Xing (2010)
- nonparametric neighborhood selection ___

- cond. like.
$$P_{\theta^t}(x_i^t | x_{\backslash i}^t) = \text{logistic}(2x_i^t \langle \theta_{\backslash i}^t, x_{\backslash i}^t \rangle)$$

- neighborhood sel. $S(x_i) = \{ j | \theta_{i,j}^t \neq 0 \}$

- time specific graph reg.
  - estimate at $t^* \in [0,1]$

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

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

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

(*) Inference (II)

TESLA: Temporally smoothed ℓ_1 reg. logistic regression

$$\hat{\theta}_i^1, \dots \hat{\theta}_i^T = \underset{\theta_i^1 \dots \theta_i^T}{\text{argmin}} \sum_{t=1}^{T} \ell_{avg}(\theta_i^t) + \lambda_1 \underbrace{\sum_{t=1}^{T} \|\theta_i^t\|_1}_{(i)} + \lambda_2 \underbrace{\sum_{t=2}^{T} \|\theta_i^t - \theta_i^{t-1}\|_q^q}_{(ii)} \qquad (I)$$

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

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

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

TESLA: $\underset{\theta_i^1 \dots \theta_i^T}{\min} \sum_{t=1}^{T} \ell(\underline{x}^t; \theta_i^t) + \lambda_1 \sum_{t=1}^{T} \underline{1} u_i^t + \lambda_2 \sum_{t=2}^{T} \underline{1} v_i^t$

$u_i^1 \dots u_i^T ; v_i^2 \dots v_i^T$

s.t. $-u_{i,j}^t \le \theta_{i,j}^t \le u_{i,j}^t \quad t=1,\dots,T \quad \forall j \in V \backslash i$

$-v_{i,j}^t \le \theta_{i,j}^t - \theta_{i,j}^{t-1} \le v_{i,j}^t \quad t=2,\dots,T \quad \forall j \in V \backslash i \quad \left(\begin{array}{c}\text{total variation}\\\text{differences}\end{array}\right)$

- can run different optimization 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 $\longrightarrow$ shows predictive power on Chafee's political movements!

  - Breast cancer $\longrightarrow$ use network inference

- TV distance is significant if you know dependency structure <u>over networks</u> (thought time)

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

- Paper contains rigorous formulation
  (; time-<u>varying</u> graphical structure estimation of GGM
  parameter r.v.s. (temporally smoothed l_1-reg. log. regression)

- Assumptions required on structure of parameter vector.

- Optimisation algorithms presented.

- Hyperparam selection $\longrightarrow$ (graph sparsity-BIC and a <u>heuristic</u>)

- Real-world datasets $\longrightarrow$ particularly interesting

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

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

- Pattern of zero-entries in precision matrix of MVG $\rightarrow$ <u>C.I. restrictions</u>
                                                                    <u>between variables</u>

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

- Is 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).