Mixture tree model for network inference

Supervision: S. Robin[1] et C. Ambroise[12]

Raphaëlle Momal

[1]UMR AgroParisTech / INRA MIA-Paris
[2]LaMME, Evry

July 6, 2018

Context

Rising interest in jointly analysed species abundances:

- Metagenomics
- Microbiology
- Ecology

Ecological network

Tool to better understand species interactions (direct/indirect), eco-systems organizations (clusters ?)

Allows for resilience analyses, pathogens control, ecosystem comparison, response prediction...

Data example

- Species: bacteria, fungi...
- Abundances: read counts from Next-Generation Sequencing technologies (metabarcoding)
- Covariates: sequencing depth, temperature, water depth...

Repeated signal : n samples, p abundances.

Data table

$Y = [Y_{ij}]_{(i,j) \in \{1,\dots,n\} \times \{1,\dots,p\}}$

- Y_{ij}: abundance of the j^{th} species in the i^{th} sample

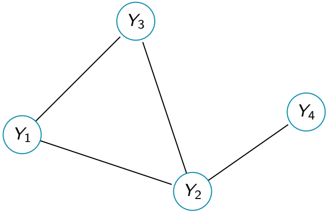
Infer the species interaction network from count data Y

Challenges

- Statistical network inference
- Count data
- Offsets and covariates

Graphical models: a statistical framework for network inference

Example:



- All variables are dependant
- Some are conditionally independent (i.e. indirectly dependeant)

  Y_4 is independent from (Y_1, Y_3) conditionally on Y_2

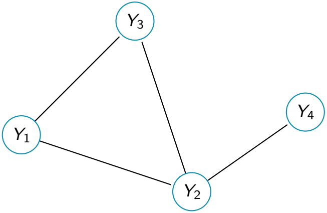
Graphical models

Definition [Lauritzen, 1996]

The joint distribution P is faithful to the graph G iff

$$P(Y_1, \dots, Y_p) \propto \prod_{C \in \mathcal{C}_G} \psi_C(Y_C)$$

where $\mathcal{C}_G$ = set of maximal cliques of G.



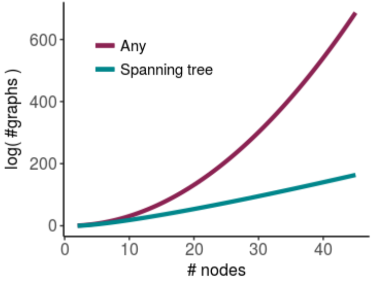
$$P(Y_1, Y_2, Y_3, Y_4) \propto \psi_1(Y_1, Y_2, Y_3) \times \psi_2(Y_3, Y_4)$$

Spanning trees

Unconstrained graph ⇒ very large space to explore: $\#\mathcal{G}_p = 2^{\frac{p(p-1)}{2}}$

Spanning trees are a sparse solution :

$$\left.\begin{array}{l} G \text{ is connected} \\ G \text{ has no cycle} \end{array}\right\} G \text{ has } (p-1) \text{ edges}$$



Much smaller space to explore:

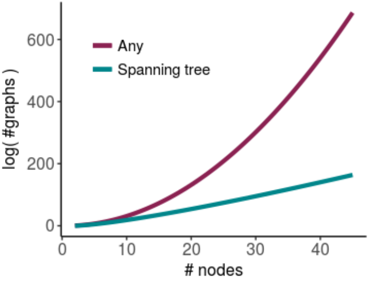
$$\#\mathcal{T}_p = p^{(p-2)}$$

Spanning trees

Unconstrained graph ⇒ very large space to explore: $\#\mathcal{G}_p = 2^{\frac{p(p-1)}{2}}$

Spanning trees are a sparse solution :

$$\left.\begin{array}{l} G \text{ is connected} \\ G \text{ has no cycle} \end{array}\right\} G \text{ has } (p-1) \text{ edges}$$



Much smaller space to explore:

$$\#\mathcal{T}_p = p^{(p-2)}$$

Still a huge complexity...

Maximizing and summing over spanning trees

Maximum spanning tree Kruskal's algorithm

$$\hat{T} = \operatorname*{argmax}_{T} \left\{ \prod_{(k,l)\in T} \psi_{k,l}(Y) \right\} \rightarrow \Theta(p^2)$$

Tree averaging Matrix tree theorem [Chaiken and Kleitman, 1978]

$$\sum_{T} \prod_{(k,l)\in T} \psi_{k,l}(Y) = \det(L(Y)) \rightarrow \Theta(p^3)$$

Maximizing and summing over spanning trees

Maximum spanning tree Kruskal's algorithm

$$\hat{T} = \underset{T}{\operatorname{argmax}} \left\{ \prod_{(k,l)\in T} \psi_{k,l}(Y) \right\} \rightarrow \Theta(p^2)$$

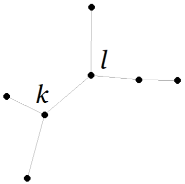
Tree averaging Matrix tree theorem [Chaiken and Kleitman, 1978]

$$\sum_{T} \prod_{(k,l)\in T} \psi_{k,l}(Y) = \det(L(Y)) \rightarrow \Theta(p^3)$$

Approach: infer the network by averaging spanning trees

Tree structured data

- Data dependency structure relies on a tree



- Likelihood factorizes on nodes and edges [Chow and Liu, 1968]:

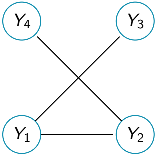
$$\mathbb{P}(Y|T) = \prod_{j=1}^{d} \mathbb{P}(Y_j) \prod_{k,l \in T} \psi_{kl}(Y) \quad ,$$

Where

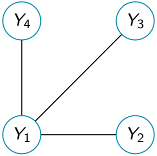
$$\psi_{kl}(Y) = \frac{\mathbb{P}(Y_k, Y_l)}{\mathbb{P}(Y_k) \times \mathbb{P}(Y_l)}.$$

Rmq : with standardised gaussian data, $\hat{\Psi} = [\hat{\psi_{kl}}] \propto (1 - \hat{\rho}^2)^{-1/2}$

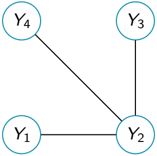
Tree averaging



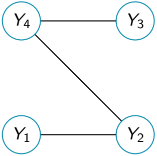
$P\{T = T_1|Y\}$



$P\{T = T_2|Y\}$



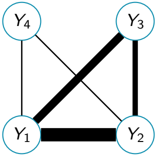
$P\{T = T_3|Y\}$



$P\{T = T_4|Y\}$

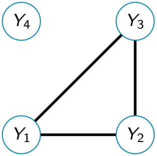
..

Compute edge probabilities:



$P\{(j,k) \in T|Y\}$

Thresholding probabilities:



$P\{(j,k) \in T|Y\}$

Model count data: PLN model

Poisson log-Normal distribution [Aitchison and Ho, 1989]

$$\left.\begin{array}{ll} Z_i \;\; iid & \sim \mathcal{N}_d(0,\Sigma) \\ & (Y_{ij})_j \perp\!\!\!\perp |Z_i \\ Y_{ij}|Z_{ij} & \sim \mathcal{P}(e^{Z_{ij}}) \end{array}\right\} Y \sim \mathcal{PLN}(0,\Sigma)$$

- Dependency structure in the Gaussian latent layer
- Easy handling of multi-variate data (contrary to Negative binomial distribution)

Model count data: PLN model

Poisson log-Normal distribution [Aitchison and Ho, 1989]

$$\left.\begin{aligned} Z_i \ iid &\sim \mathcal{N}_d(0,\Sigma) \\ &(Y_{ij})_j \perp\!\!\!\perp |Z_i \\ Y_{ij}|Z_{ij} &\sim \mathcal{P}(e^{o_{ij}+x_i^{\intercal}\Theta_j+Z_{ij}}) \end{aligned}\right\} Y \sim \mathcal{PLN}(O + X^{\intercal}\Theta, \Sigma)$$

- Dependency structure in the Gaussian latent layer
- Easy handling of multi-variate data (contrary to Negative binomial distribution)
- Allow adjustment for covariates and offsets
- Variational estimation algorithm [Chiquet et al., 2017]

PLN + mixture tree

G is taken as a spanning tree T, the dependency structure is encoded in Σ_T.

$$Z \sim \mathcal{N}(0, \Sigma_T)$$

Tree averaging (mixture model):

$$Z \sim \sum_T w_T \mathcal{N}(0, \Sigma_T)$$

$$\Rightarrow \mathbb{P}(Z) \propto \sum_T \prod_{k,l \in T} \beta_{k,l} \psi_{k,l}(Z)$$

Hierarchical model with latent tree

1. A spanning tree is drawn in a distribution decomposable on the edges:

Decomposable distribution for a tree T [Meilă and Jaakkola, 2006]

$$\mathbb{P}(T) = \frac{1}{B} \prod_{(k,l)\in T} \beta_{kl} \text{ , avec } B = \sum_{T\in\mathcal{T}} \prod_{(k,l)\in T} \beta_{kl}$$

- A weight β_{kl} is assign to each edge (k, l)
- The dependence tree probability is proportional to its weights product
- We consider varying weights

Hierarchical model with latent tree

1. A spanning tree is drawn in a distribution decomposable on the edges:

Decomposable distribution for a tree T [Meilă and Jaakkola, 2006]

$$\mathbb{P}(T) = \frac{1}{B} \prod_{(k,l)\in T} \beta_{kl} \text{ , avec } B = \sum_{T\in\mathcal{T}} \prod_{(k,l)\in T} \beta_{kl}$$

- A weight β_{kl} is assign to each edge (k,l)
- The dependence tree probability is proportional to its weights product
- We consider varying weights

2. Data is simulated conditionally on the drawn tree :

$$Z|T \sim \mathcal{N}_d(0, \Sigma_T)$$

Hierarchical model with latent tree

1. A spanning tree is drawn in a distribution decomposable on the edges:

Decomposable distribution for a tree T [Meilă and Jaakkola, 2006]

$$\mathbb{P}(T) = \frac{1}{B} \prod_{(k,l)\in T} \beta_{kl} \text{ , avec } B = \sum_{T\in\mathcal{T}} \prod_{(k,l)\in T} \beta_{kl}$$

- A weight β_{kl} is assign to each edge (k, l)
- The dependence tree probability is proportional to its weights product
- We consider varying weights

2. Data is simulated conditionally on the drawn tree :

$$Z|T \sim \mathcal{N}_d(0, \Sigma_T)$$

The data shaping tree is treated as a latent variable.

$$\mathbb{P}(Z) = \sum_{T\in\mathcal{T}} \mathbb{P}(T)\mathbb{P}(Z|T) : \text{ mixture tree}$$

E step

- Complete likelihood :

$$\mathbb{P}(Y,Z,T) = \mathbb{P}(T) \times \mathbb{P}(Z|T) \times \mathbb{P}(Y|Z)$$

$$\begin{aligned}\log(\mathbb{P}(Y,Z,T)) = &\sum_{k,l} \mathbb{1}_{\{(k,l)\in T\}}(\log(\beta_{kl}) + \log(\psi_{kl}(Z))) - \log(B) \\ &+ \sum_{k}(\log(\mathbb{P}(Z_k)) + \log(\mathbb{P}(Y_k|Z_k)))\end{aligned}$$

E step

- Complete likelihood :

$$\mathbb{P}(Y,Z,T) = \mathbb{P}(T) \times \mathbb{P}(Z|T) \times \mathbb{P}(Y|Z)$$

$$\log(\mathbb{P}(Y,Z,T)) = \sum_{k,l} \mathbb{1}_{\{(k,l)\in T\}}(\log(\beta_{kl}) + \log(\psi_{kl}(Z))) - \log(B) \\ + \sum_{k}(\log(\mathbb{P}(Z_k)) + \log(\mathbb{P}(Y_k|Z_k)))$$

- Conditional expectation :

$$\mathbb{E}_\theta[\log(\mathbb{P}(Y,Z,T))|Y] = \sum_{k,l\in V} \mathbb{P}((k,l)\in T|Y)\log(\beta_{kl}) + \mathbb{E}[\mathbb{1}_{\{(k,l)\in T\}}\log(\psi_{kl}(Z)|Y)] \\ + \sum_{k} \mathbb{E}[\log(\mathbb{P}(Z_k))|Y] + \mathbb{E}[\log(\mathbb{P}(Y_k|Z_k))|Y] - \log(B)$$

Two steps solution

The PLNmodels package approximates the distribution parameters. Using PLNmodels:

1. Estimate $\hat{\Sigma}_Z$
2. Apply EM mixture tree to $Z \sim \mathcal{N}(0, \hat{\Sigma}_Z)$

Simplified conditional expectation writing:

$$\mathbb{E}_\theta[\log(\mathbb{P}(Z,T))|Z] = \sum_{k,l} \mathbb{P}((k,l) \in T|Z)(\log(\beta_{kl}) + \log(\psi_{kl})) - \log(B) + \sum_k \log(\mathbb{P}(Z_k))$$

$\Rightarrow$ **EM algorithm**

EM output post-treatment

Thresholding Output of EM: conditional probabilities for each edge to be part of the graph.

Probability for an edge to be part of a tree drawn uniformly $= \lambda = 2/p$.

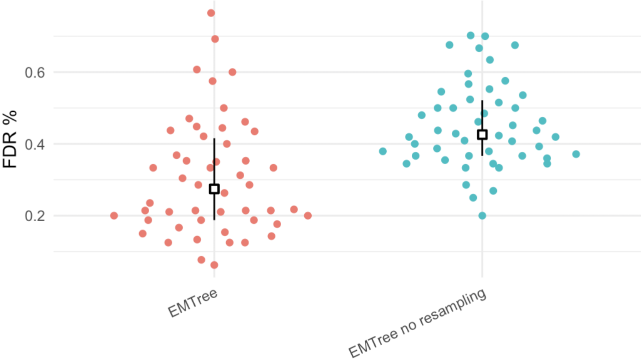
Resampling

- B sub-samples using a fraction f of available observations.
- For $b = 1 \dots B$, $\widehat{G}^b$ is made of the edges having probability $P^b_{\geq} 2/p$.
- Only edges selected in more than a fraction f' of the estimated graphs $\widehat{G}^b$ are kept to build the final $\widehat{G}$.

Can be parallelized. We set $f = f' = 80\%$.

Resampling example

B=150, Cluster structure, n=100, p=20 :



Three alternative methods

Two methods which take compositional data as inputs :

- SpiecEasi algorithm [Kurtz et al., 2015] (glasso on transformed counts)
- gCoda [Fang et al., 2017] (logistic-normal model with MM algorithm)

One taking raw counts and covariates :

- MInt [Biswas et al., 2016] (uses PLN model, greedy inference with a penalized approach)

Three main questions

1. Effect of difficulty level
2. Effect of structure
3. Robustness to the tree hypothesis

Simulation design

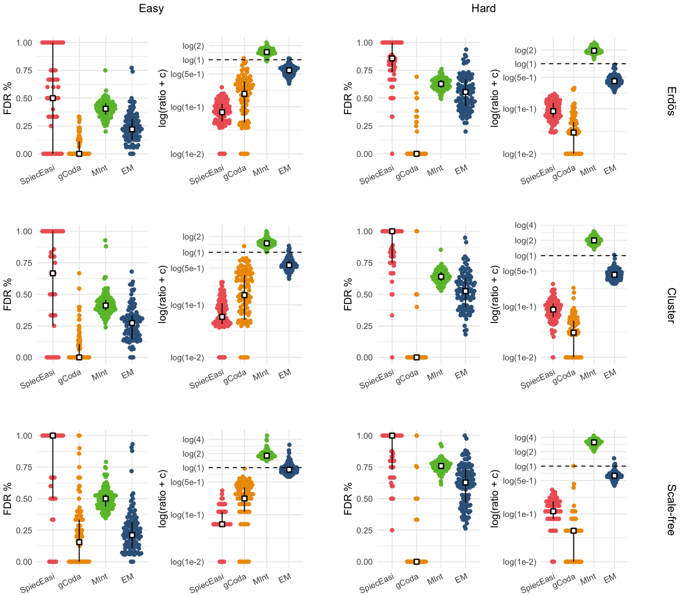
1. Choose G and define Ω accordingly
2. Sample count data Y from $\mathcal{P}\updownarrow\mathcal{N}(0,\Omega^{-1})$ with possible covariates
3. Infer the network with PLN + mixture tree VEM, SpiecEasi, gCoda, and MInt
4. Compare results with presence/absence of edges (FDR, AUC)

Remarks

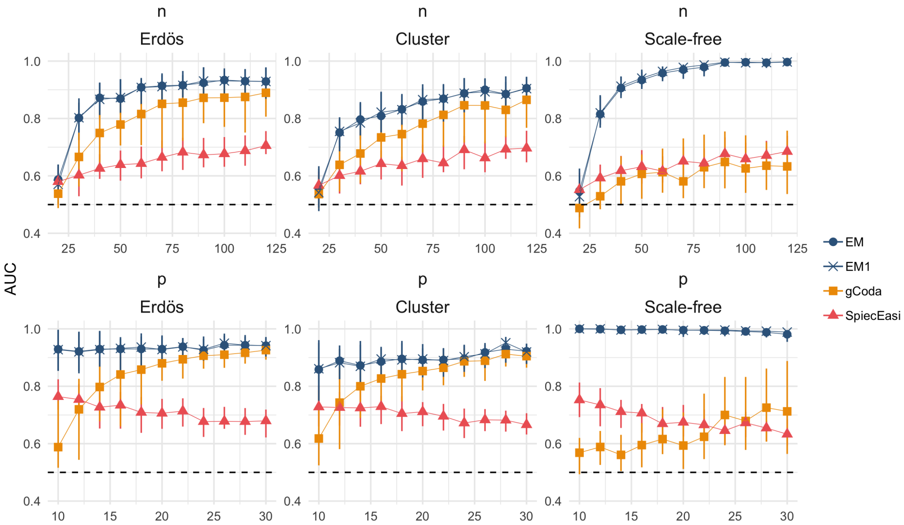
- gCoda and SpiecEasi do not account for covariates : residuals from the regression of transformed data.
- The MInt method gives an optimized network and we can only compare ourmethod to it with the FDR criterion.

$\Rightarrow$ 100 replicates for each setting (parameters × structure)

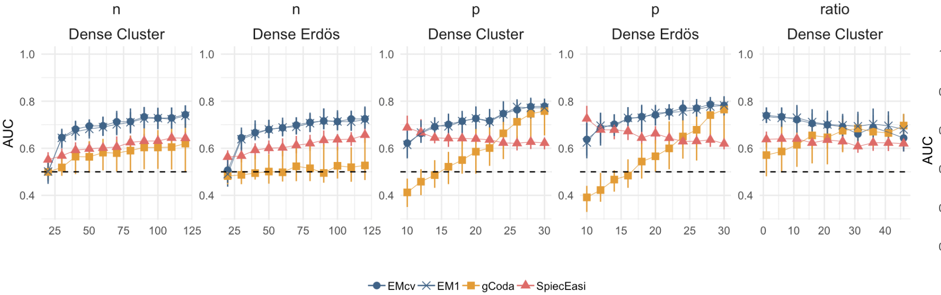
Difficulty level



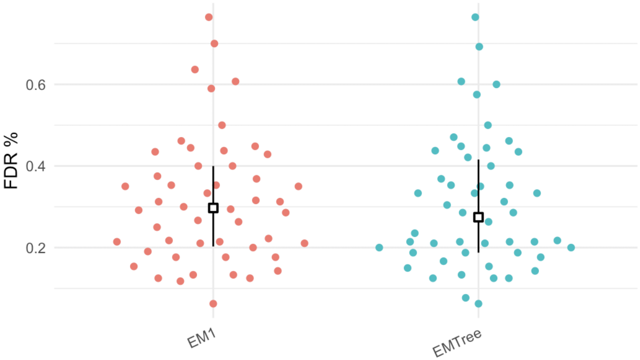
Effect of structure



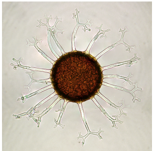
Away from tree-like density: $5/p$



With one iteration



Oak Mildew



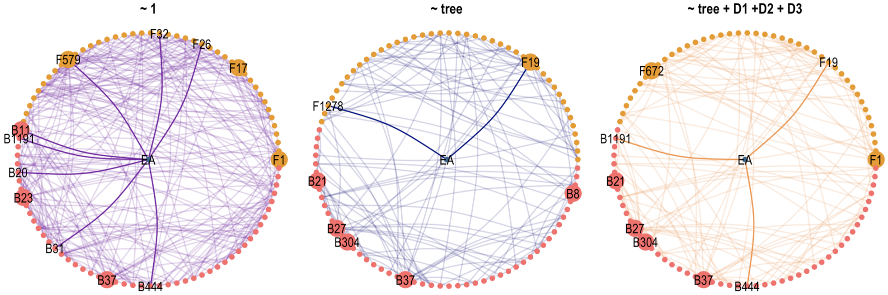
Pathogen Erysiphe alphitoides (EA).

Oak leaf with powdery mildew.

Metabarcoding of oak tree leaves microbiome [Jakuschkin et al., 2016].

- 114 sample of 94 microbial species counts (bacteria/fungi)
- Different read depth for bacteria and fungi: unsuited for compositional data normalization
- covariates: tree identifier and distances locating leaves in space

Inferred networks



Conclusion

Contributions:

- Formal probabilistic model for network inference with count data
- Variational estimation algorithm
- Inclusion of offsets and covariates

Perspectives:

- Network comparison
- Model for the inference in the observed counts layer
- Missing major actor (species/covariable)

Acknowledgments

Special thanks :

Supervisors	Stéphane Robin, Christophe Ambroise
PLN team	Julien Chiquet (MIA-Paris), Mahendra Mariadassou (INRA Jouy)
Data	Corinne Vacher (INRA Bordeaux)

Contact :

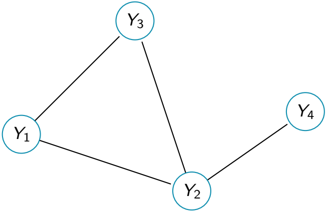
email	raphaelle.momal@agroparistech.fr
Web	Rmomal.github.io

Gaussian Graphical Models (GGM)

Gaussian distribution:

$$Y_i \sim \mathcal{N}_p(\mu, \Sigma),\ \mu = \text{vector of means},\ \Sigma = \text{covariance matrix}.$$

A nice property:



Inverse covariance matrix

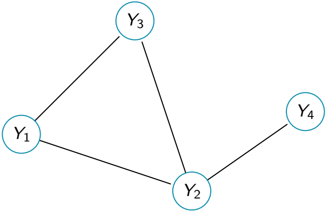
$$\Sigma^{-1} = \Omega \propto \begin{bmatrix} 1 & .5 & .5 & 0 \\ .5 & 1 & .5 & .5 \\ .5 & .5 & 1 & 0 \\ 0 & .5 & 0 & 1 \end{bmatrix}$$

Gaussian Graphical Models (GGM)

Gaussian distribution:

$$Y_i \sim \mathcal{N}_p(\mu, \Sigma),\ \mu = \text{vector of means},\ \Sigma = \text{covariance matrix.}$$

A nice property:



Inverse covariance matrix

$$\Sigma^{-1} = \Omega \propto \begin{bmatrix} 1 & .5 & .5 & 0 \\ .5 & 1 & .5 & .5 \\ .5 & .5 & 1 & 0 \\ 0 & .5 & 0 & 1 \end{bmatrix}$$

Glasso on gaussian data: $\widehat{\Omega}_\lambda = \arg\min_{\Omega \in \mathcal{S}_d^+} \left\{ L(Y, \Omega) + \lambda \sum_{i \neq j} |\omega_{ij}| \right\}$

⇒ SpiecEasi method [Kurtz et al., 2015]: glasso on transformed counts

Conditional probability computation

Kirchhoff's theorem (matrix tree, [Aitchison and Ho, 1989])

For all $W = (a_{kl})_{k,l}$ a symmetric matrix, the corresponding Laplacian $Q(W)$ is defined as follows:

$$\mathcal{Q}_{uv}(W) = \begin{cases} -a_{uv} & 1 \leq u < v \leq n \\ \sum_{i=1}^{n} a_{vi} & 1 \leq u = v \leq n. \end{cases}$$

Then for all u et v:

$$|Q^*_{uv}(W)| = \sum_{T \in \mathcal{T}} \prod_{\{k,l\} \in E_T} a_{kl}$$

$$\begin{aligned} \mathbb{P}((k,l) \in T | Z) = \sum_{T \in \mathcal{T}:(k,l) \in T} \mathbb{P}(T|Z) = \frac{\sum_{(k,l) \in T} \mathbb{P}(T)\mathbb{P}(Z|T)}{\sum_T \mathbb{P}(T)\mathbb{P}(Z|T)} \\ = 1 - \frac{|Q^*_{uv}(B\Psi^{-kl})|}{|Q^*_{uv}(B\Psi)|} \\ = \tau_{kl} \end{aligned}$$

M step

Goal : optimization of weights β_{kl}.

$$\underset{\beta_{kl}}{\operatorname{argmax}} \left\{ \sum_{k,l \in V} \tau_{kl} (\log(\beta_{kl}) + \log(\psi_{kl})) - \log(B) + \sum_{k} \log(\mathbb{P}(Z_k)) \right\}$$

With high combinatorial complexity of $B = \sum_{T \in \mathcal{T}} \prod_{k,l \in T} \beta_{kl}$

How to compute $\frac{\partial B}{\partial \beta_{kl}}$?

β_{kl} update

A result from Meilă [Meilă and Jordan, 2000]

Inverting a minor of the laplacien Q, we define M :

$$\begin{cases} M_{uv} = [\mathcal{Q}^{*-1}]_{uu} + [\mathcal{Q}^{*-1}]_{vv} - 2[\mathcal{Q}^{*-1}]_{uv} & u, v < n \\ M_{nv} = M_{vn} = [\mathcal{Q}^{*-1}]_{vv} & v < n \\ M_{vv} = 0. \end{cases}$$

On peut montrer que :

$$\frac{\partial |Q^*_{uv}(W)|}{\partial \beta_{kl}} = M_{kl} \times |Q^*_{uv}(W)|$$

$$\frac{\partial \mathbb{E}_\theta[\log(\mathbb{P}(Z, T))|Z]}{\partial \beta_{kl}} = \frac{1}{\beta_{kl}} \tau_{kl} - \frac{1}{B} \frac{\partial B}{\partial \beta_{kl}}$$

Update formula at iteration $h + 1$

$$\hat{\beta}^{h+1}_{kl} = \frac{\tau^h_{kl}}{M^h_{kl}}$$

References I

Aitchison, J. and Ho, C. (1989).
The multivariate Poisson-log normal distribution.
Biometrika, 76(4):643–653.

Biswas, S., McDonald, M., Lundberg, D. S., Dangl, J. L., and Jojic, V. (2016).
Learning microbial interaction networks from metagenomic count data.
Journal of Computational Biology, 23(6):526–535.

Chaiken, S. and Kleitman, D. J. (1978).
Matrix tree theorems.
Journal of combinatorial theory, Series A, 24(3):377–381.

Chiquet, J., Mariadassou, M., and Robin, S. (2017).
Variational inference for probabilistic Poisson PCA.
Technical report, arXiv:1703.06633.
to appear in *Annals of Applied Statistics*.

Chow, C. and Liu, C. (1968).
Approximating discrete probability distributions with dependence trees.
IEEE Transactions on Information Theory, 14(3):462–467.

References II

Fang, H., Huang, C., Zhao, H., and Deng, M. (2017).
gcoda: conditional dependence network inference for compositional data.
Journal of Computational Biology, 24(7):699–708.

Jakuschkin, B., Fievet, V., Schwaller, L., Fort, T., Robin, C., and Vacher, C. (2016).
Deciphering the pathobiome: Intra- and interkingdom interactions involving the pathogen erysiphe alphitoides.
Microb Ecol, 72(4):870–880.

Kurtz, Z. D., Müller, C. L., Miraldi, E. R., Littman, D. R., Blaser, M. J., and Bonneau, R. A. (2015).
Sparse and compositionally robust inference of microbial ecological networks.
PLoS computational biology, 11(5):e1004226.

Lauritzen, S. L. (1996).
Graphical Models.
Oxford Statistical Science Series. Clarendon Press.

Meilă, M. and Jaakkola, T. (2006).
Tractable bayesian learning of tree belief networks.
Statistics and Computing, 16(1):77–92.

Meilă, M. and Jordan, M. I. (2000).
Learning with mixtures of trees.
Journal of Machine Learning Research, 1:1–48.