Exact Bayesian inference for some models with discrete parameters

S. Robin

Joint work with A. Cleynen, E. Lebarbier, G. Rigaill, L. Schwaller, M. Stumpf

INRA / AgroParisTech

CIRM, March 2016, Marseille

General framework

Generic Bayesian framework:

$$\begin{aligned} \text{prior:} \quad & p(\vartheta) \\ \text{likelihood:} \quad & p(Y|\vartheta) \\ \rightarrow \text{posterior:} \quad & p(\vartheta|Y) \end{aligned}$$

General framework

Generic Bayesian framework:

$$
\begin{aligned}
\text{prior:} &\quad p(\vartheta) \\
\text{likelihood:} &\quad p(Y|\vartheta) \\
\rightarrow \text{posterior:} &\quad p(\vartheta|Y)
\end{aligned}
$$

3 main approaches

General framework

Generic Bayesian framework:

$$\begin{aligned} \text{prior:} \quad & p(\vartheta) \\ \text{likelihood:} \quad & p(Y|\vartheta) \\ \rightarrow \text{posterior:} \quad & p(\vartheta|Y) \end{aligned}$$

3 main approaches

1. Sampling (MC, MCMC, SMC, IS, ...): get $(\vartheta^b) \sim p(\vartheta|Y)$.

General framework

Generic Bayesian framework:

$$\begin{aligned} \text{prior:} \quad & p(\vartheta) \\ \text{likelihood:} \quad & p(Y|\vartheta) \\ \longrightarrow \text{posterior:} \quad & p(\vartheta|Y) \end{aligned}$$

3 main approaches

1. Sampling (MC, MCMC, SMC, IS, ...): get $(\vartheta^b) \sim p(\vartheta|Y)$.

2. Approximation (e.g. VB, EP, ...): find $q_Y(\vartheta) \simeq p(\vartheta|Y)$.

General framework

Generic Bayesian framework:

$$\begin{aligned} \text{prior:} \quad & p(\vartheta) \\ \text{likelihood:} \quad & p(Y|\vartheta) \\ \longrightarrow \text{posterior:} \quad & p(\vartheta|Y) \end{aligned}$$

3 main approaches

1. Sampling (MC, MCMC, SMC, IS, ...): get $(\vartheta^b) \sim p(\vartheta|Y)$.
2. Approximation (e.g. VB, EP, ...): find $q_Y(\vartheta) \simeq p(\vartheta|Y)$.
3. Exact: actually compute $p(\vartheta|Y)$ or some marginal of interest.

Models with discrete parameters

Mixed parameter: $\vartheta = (\theta, T)$

$$\theta \in \Theta = \text{continuous set}, \qquad T \in \mathcal{T} = \text{discrete (countable) set},$$

$$\Rightarrow \qquad p(Y) = \sum_{T \in \mathcal{T}} \int_\Theta p(Y, \theta, T) \, \mathrm{d}\theta$$

Models with discrete parameters

Mixed parameter: $\vartheta = (\theta, T)$

$$\theta \in \Theta = \text{continuous set}, \qquad T \in \mathcal{T} = \text{discrete (countable) set},$$

$$\Rightarrow \qquad p(Y) = \sum_{T \in \mathcal{T}} \int_\Theta p(Y, \theta, T) \, \mathrm{d}\theta$$

Size of $\mathcal{T}$.

- No big deal of $\mathcal{T}$ is small (e.g. model selection within a small collection).
- Big issue if $|\mathcal{T}|$ grows (super-)exponentially with the number of observations n or the number of variables p.

Main issue

The calculation of

$$\sum_{T \in \mathcal{T}}$$

can often not be achieved in a naive way because of the combinatorial complexity[1].

$\rightarrow$ Need to find algorithmic or algebraic shortcuts[2]

[1]The frequentist counterpart often raises similar issues.

[2]Supposing that $\int_\Theta$ raises no specific issues (e.g. conjugate priors).

Main issue

The calculation of

$$\sum_{T \in \mathcal{T}}$$

can often not be achieved in a naive way because of the combinatorial complexity[1].

$\longrightarrow$ Need to find algorithmic or algebraic shortcuts[2]

2 examples.

- Change-point detection
- 'Network inference' = inference of the structure of a graphical model

[1]The frequentist counterpart often raises similar issues.
[2]Supposing that $\int_\Theta$ raises no specific issues (e.g. conjugate priors).

Outline

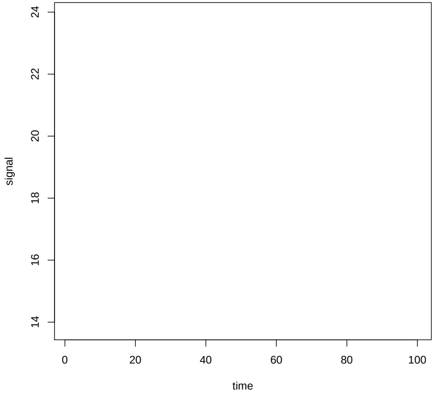
A change-point detection model

Model.

A change-point detection model

Model.

- K segments



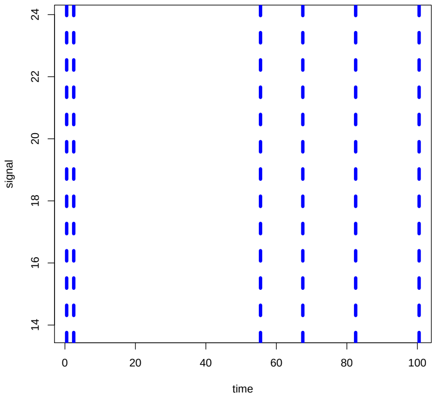
A change-point detection model

Model.

- K segments

- $T = (\tau_k)_k$ change points

$r_k = [\![\tau_{k-1} + 1; \tau_k]\!]$

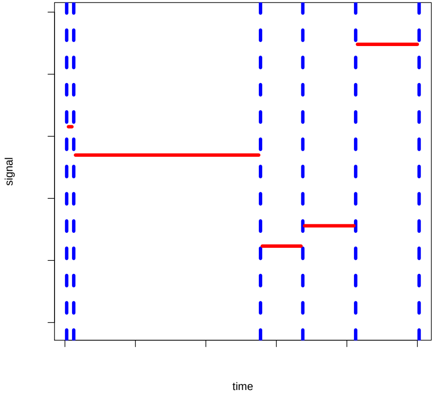


A change-point detection model

Model.

- K segments
- $T = (\tau_k)_k$ change points

  $r_k = [\![\tau_{k-1} + 1; \tau_k]\!]$
- $\theta = (\theta_k)_k$ parameters



A change-point detection model

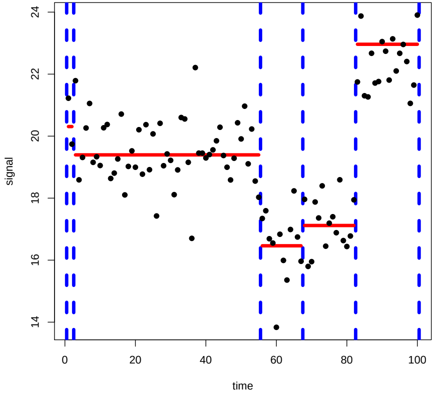
$$\{Y^r\}_r \text{ indep}, \quad Y^r \sim p(\cdot|\theta_r)$$

Model.

- K segments
- $T = (\tau_k)_k$ change points

  $r_k = [\![\tau_{k-1} + 1; \tau_k]\!]$
- $\theta = (\theta_k)_k$ parameters
- $Y = (Y_t)_{1 \leq t \leq n}$ observed data

  $Y^r = (Y_t)_{t \in r}$



A change-point detection model

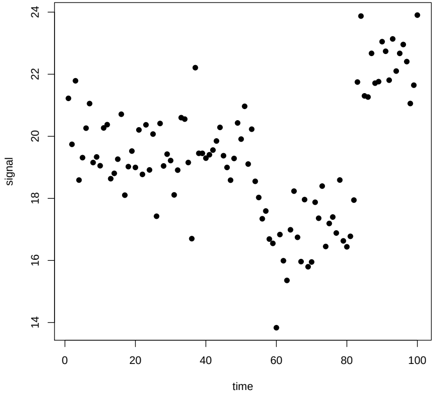
$$\{Y^r\}_r \text{ indep}, \quad Y^r \sim p(\cdot|\theta_r)$$

Model.

- K segments
- $T = (\tau_k)_k$ change points

  $r_k = [\![\tau_{k-1} + 1; \tau_k]\!]$
- $\theta = (\theta_k)_k$ parameters
- $Y = (Y_t)_{1 \leq t \leq n}$ observed data

  $Y^r = (Y_t)_{t \in r}$



Bayesian version: on the top of this, add $p(K), p(T|K), p(\theta|K)$.

Maximum likelihood inference (1/2)

Log-likelihood:

$$\log p(Y; \theta, T) = \sum_{r \in T} \log p(Y^r; \theta^r)$$

Maximum likelihood inference (1/2)

Log-likelihood:

$$\log p(Y; \theta, T) = \sum_{r \in T} \log p(Y^r; \theta^r)$$

Inference

- continuous part (θ):

$$\widehat{\theta}_r = \arg\max_{\theta_r} \log p(Y^r; \theta^r) \qquad \text{standard MLE}$$

Maximum likelihood inference (1/2)

Log-likelihood:

$$\log p(Y; \theta, T) = \sum_{r \in T} \log p(Y^r; \theta^r)$$

Inference

- continuous part (θ):

$$\widehat{\theta}_r = \arg\max_{\theta_r} \log p(Y^r; \theta^r) \qquad \text{standard MLE}$$

- discrete part (T):

$$\widehat{T} = \arg\max_{T} \sum_{r \in T} \log p(Y^r; \widehat{\theta}^r) = \arg\max_{T} \sum_{r \in T} \log \widehat{p}(Y^r)$$

$\longrightarrow$ discrete optimization problem

Maximum likelihood inference (2/2)

Segmentation space $\mathcal{T} = \mathcal{T}_{1:n}^K =$ set of all possible segmentations of $[\![1; n]\!]$ with K segments:

$$|\mathcal{T}| = \binom{n-1}{K-1} \approx \left(\frac{n}{K}\right)^K$$

$\rightarrow$ exhaustive search is prohibited.

Maximum likelihood inference (2/2)

Segmentation space $\mathcal{T} = \mathcal{T}_{1:n}^K =$ set of all possible segmentations of $[\![1; n]\!]$ with K segments:

$$|\mathcal{T}| = \binom{n-1}{K-1} \approx \left(\frac{n}{K}\right)^K$$

$\rightarrow$ exhaustive search is prohibited.

Dynamic programming allows to retrieve $\widehat{T}$ [1] using

$$\max_{T \in \mathcal{T}_{1:j}^K} \sum_{r \in T} \log \widehat{p}(Y^r) = \max_{K-1 \leq i < j} \left(\max_{T \in \mathcal{T}_{1:i}^{K-1}} \sum_{r \in T} \log \widehat{p}(Y^r) \right) + \log \widehat{p}(Y^{[\![i+1;j]\!]})$$

Maximum likelihood inference (2/2)

Segmentation space $\mathcal{T} = \mathcal{T}_{1:n}^K$ = set of all possible segmentations of $[\![1; n]\!]$ with K segments:

$$|\mathcal{T}| = \binom{n-1}{K-1} \approx \left(\frac{n}{K}\right)^K$$

$\rightarrow$ exhaustive search is prohibited.

Dynamic programming allows to retrieve $\widehat{T}$ [1] using

$$\max_{T \in \mathcal{T}_{1:j}^K} \sum_{r \in T} \log \widehat{p}(Y^r) = \max_{K-1 \leq i < j} \left(\max_{T \in \mathcal{T}_{1:i}^{K-1}} \sum_{r \in T} \log \widehat{p}(Y^r) \right) + \log \widehat{p}(Y^{[\![i+1;j]\!]})$$

Still, further inference is hard to achieve

$\rightarrow$ Standard likelihood theory does not apply to discrete parameters
(no simple confidence intervals for the τ_k).

$\rightarrow$ Bayesian inference can circumvent some difficulties.

Bayesian inference

Factorability assumptions

- Prior distribution for the segmentation:

$$p(T|K) = \prod_{r \in T} a(r), \qquad \text{e.g. } a(r) = n_r^{\alpha}$$

- Independent parameters in each segment:

$$p(\theta|T) = \prod_{r \in T} p(\theta_r)$$

- Data are independent from one segment to another

$$p(Y|T, \theta) = \prod_{r \in T} p(Y^r|\theta_r)$$

Some quantities of interest

Marginal likelihood.

$$p(Y|K) = \sum_{T \in \mathcal{T}^K} \int p(Y, \theta, T|K) \, \mathrm{d}\theta \propto \sum_{T \in \mathcal{T}^K} \prod_{r \in T} a(r) p(Y^r)$$

where $p(Y^r) = \int p(Y^r|\theta_r) p(\theta^r) \, \mathrm{d}\theta_r$ (supposed to be easy to compute using e.g. conjugate priors) and the normalizing constants is

$$\sum_{T \in \mathcal{T}^K} \prod_{r \in T} a(r).$$

Some quantities of interest

Marginal likelihood.

$$p(Y|K) = \sum_{T \in \mathcal{T}^K} \int p(Y, \theta, T|K) \, \mathrm{d}\theta \propto \sum_{T \in \mathcal{T}^K} \prod_{r \in T} a(r) p(Y^r)$$

where $p(Y^r) = \int p(Y^r|\theta_r) p(\theta^r) \, \mathrm{d}\theta_r$ (supposed to be easy to compute using e.g. conjugate priors) and the normalizing constants is

$$\sum_{T \in \mathcal{T}^K} \prod_{r \in T} a(r).$$

Posterior distribution of a change-point.

$$\Pr\{\tau_k = t|Y, K\} \propto \left(\sum_{T \in \mathcal{T}^k_{1:t}} \prod_{r \in T} a(r) p(Y^r) \right) \left(\sum_{T \in \mathcal{T}^{K-k}_{t+1:n}} \prod_{r \in T} a(r) p(Y^r) \right)$$

Summing over segmentations [9]

Property: *Define the upper triangular* $(n+1) \times (n+1)$ *matrix* A*:*

$$A_{i,j+1} = f([\![i,j]\!]), \qquad 1 \leq i < j \leq n$$

Summing over segmentations [9]

Property: *Define the upper triangular $(n+1) \times (n+1)$ matrix A:*

$$A_{i,j+1} = f([\![i,j]\!]), \qquad 1 \leq i < j \leq n$$

Then

$$\left[A^K\right]_{1,n+1} = \sum_{T \in \mathcal{T}^K_{1:n}} \prod_{r \in T} f(r)$$

$\rightarrow$ *all terms are computed in $O(Kn^2)$.*

Summing over segmentations [9]

Property: *Define the upper triangular* $(n+1) \times (n+1)$ *matrix A:*

$$A_{i,j+1} = f(\llbracket i,j \rrbracket), \qquad 1 \leq i < j \leq n$$

Then

$$\left[A^K\right]_{1,n+1} = \sum_{T \in \mathcal{T}^K_{1:n}} \prod_{r \in T} f(r)$$

$\rightarrow$ *all terms are computed in* $O(Kn^2)$*.*

- To compute $p(Y)$, take $f(r) = a(r)p(Y^r)$.
- Similar ideas in [5].

Summing over segmentations [9]

Property: *Define the upper triangular* $(n+1) \times (n+1)$ *matrix* A*:*

$$A_{i,j+1} = f([\![i,j]\!]), \qquad 1 \leq i < j \leq n$$

Then

$$\left[A^K\right]_{1,n+1} = \sum_{T \in \mathcal{T}_{1:n}^K} \prod_{r \in T} f(r)$$

$\longrightarrow$ *all terms are computed in* $O(Kn^2)$*.*

- To compute $p(Y)$, take $f(r) = a(r)p(Y^r)$.
- Similar ideas in [5].
- 'sum-product' = counterpart of 'max-sum' in the dynamic programming algorithm.

Summing over segmentations [9]

Property: *Define the upper triangular* $(n+1) \times (n+1)$ *matrix* A:

$$A_{i,j+1} = f(\llbracket i,j \rrbracket), \qquad 1 \leq i < j \leq n$$

Then

$$\left[A^K\right]_{1,n+1} = \sum_{T \in \mathcal{T}^K_{1:n}} \prod_{r \in T} f(r)$$

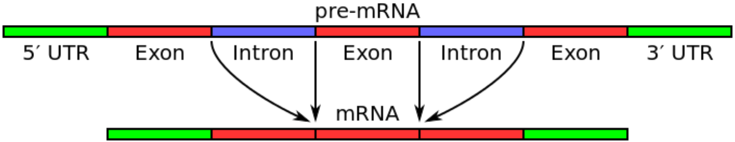
$\rightarrow$ *all terms are computed in* $O(Kn^2)$*.*

- To compute $p(Y)$, take $f(r) = a(r)p(Y^r)$.
- Similar ideas in [5].
- 'sum-product' = counterpart of 'max-sum' in the dynamic programming algorithm.

$\rightarrow$ R package EBS (exact Bayesian segmentation) [3]

Illustration: Of exons, introns and UTR's

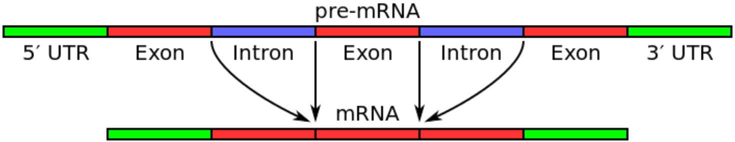
Regions for a same gene are not adjacent along the genome



[Wikipedia]

Illustration: Of exons, introns and UTR's

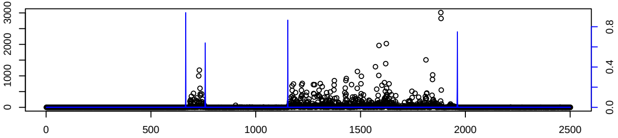
Regions for a same gene are not adjacent along the genome



[Wikipedia]

- The transcribed regions are made of both exons and untranslated regions (UTR)
- Alternative splicing: some exons can be skipped or the boundaries may vary.

Posterior distribution of transcript boundaries in yeast

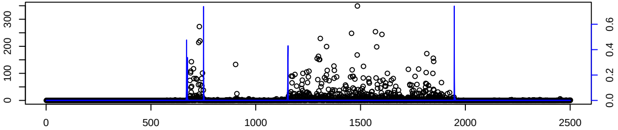


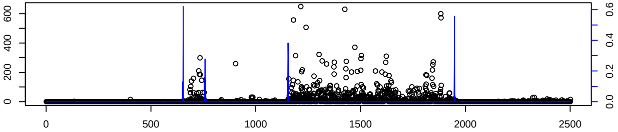
RNA-seq data:

One gene

$\times$

Three growth conditions A, B, C





Comparing change-point locations [3]

One series. We know how to compute (in $O(Kn^2)$)

$$\Pr\{\tau_k = t | Y, K\} \qquad \text{or} \qquad \Pr\{\tau_k = t | Y\}.$$

[3] Requires a probability change, as $Y^A, \dots Y^I$ are not independent conditionally on $\tau_k^A = \cdots = \tau_k^I$.

Comparing change-point locations [3]

One series. We know how to compute (in $O(Kn^2)$)

$$\Pr\{\tau_k = t | Y, K\} \qquad \text{or} \qquad \Pr\{\tau_k = t | Y\}.$$

Two series (Y^A, Y^B): Consider the shift of the kth change-point

$$\Pr\{\tau_k^A - \tau_k^B = 0 | Y^A, Y^B, K^A, K^B\}$$

[3] Requires a probability change, as $Y^A, \dots Y^I$ are not independent conditionally on $\tau_k^A = \dots = \tau_k^I$.

Comparing change-point locations [3]

One series. We know how to compute (in $O(Kn^2)$)

$$\Pr\{\tau_k = t | Y, K\} \qquad \text{or} \qquad \Pr\{\tau_k = t | Y\}.$$

Two series (Y^A, Y^B): Consider the shift of the kth change-point

$$\Pr\{\tau_k^A - \tau_k^B = 0 | Y^A, Y^B, K^A, K^B\}$$

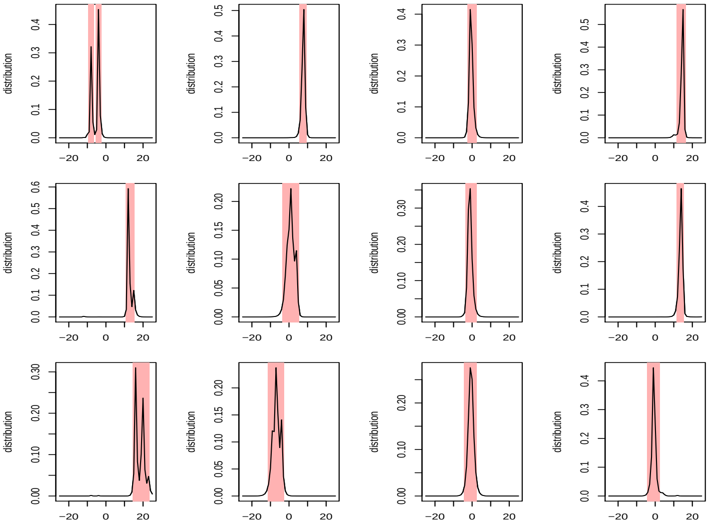
I series $(Y^A, \dots Y^I)$: Check if the kth change-point is conserved[3]:

$$\Pr\{\tau_k^A = \dots = \tau_k^I | Y^A, \dots Y^I, K^A, \dots, K^I\}$$

[3] Requires a probability change, as $Y^A, \dots Y^I$ are not independent conditionally on $\tau_k^A = \dots = \tau_k^I$.

Boundary shifts between conditions

3 comparisons (A/B, A/C, B/C) $\times$ 4 change points:



Comparing transcript boundaries

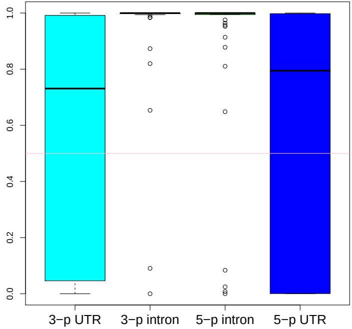
Setting $\Pr\{\tau_k^A = \tau_k^B | K\} = 1/2$.

	τ_1	τ_2	τ_3	τ_4
$\Pr\{\tau_k^A = \tau_k^B \| Y, K\}$	0.32	0.30	0.99	10^{-5}
$\Pr\{\tau_k^A = \tau_k^C \| Y, K\}$	$4\ 10^{-4}$	0.99	0.99	$6\ 10^{-3}$
$\Pr\{\tau_k^B = \tau_k^C \| Y, K\}$	$5\ 10^{-2}$	0.60	0.99	0.99
$\Pr\{\tau_k^A = \tau_k^B = \tau_k^C \| Y, K\}$	10^{-3}	0.99	0.99	$6\ 10^{-3}$

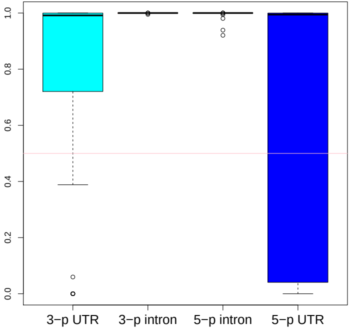
$\rightarrow$ Differences at the UTR's end but not at internal exon boundaries.

Various isoforms in yeast?

$\Pr\{\tau_k^A = \tau_k^B = \tau_k^C | Y, K\}$ for all yeast genes with 2 expressed exons



$p_0 = (.5,\ .5,\ .5,\ .5)$



$p_0 = (.9,\ .99,\ .99,\ .9)$

Outline

Graphical model framework

Property [Hammersley-Clifford]. The joint distribution $p(Y) = p(Y_1, \dots Y_p)$ is Markov wrt the (decomposable) graph G iff it factorizes wrt the maximal cliques of G:

$$p(Y) \propto \prod_{C \in \mathcal{C}(G)} \psi(Y^c), \qquad Y^c = (Y_j)_{j \in C}.$$

$\rightarrow$ G reveals the structure of conditional independences between the variables $Y_1, \dots Y_p$.

Graphical model framework

Property [Hammersley-Clifford]. The joint distribution $p(Y) = p(Y_1, \dots Y_p)$ is Markov wrt the (decomposable) graph G iff it factorizes wrt the maximal cliques of G:

$$p(Y) \propto \prod_{C \in \mathcal{C}(G)} \psi(Y^c), \qquad Y^c = (Y_j)_{j \in C}.$$

$\rightarrow$ G reveals the structure of conditional independences between the variables $Y_1, \dots Y_p$.

'Network inference' problem: Based on $\{(Y_{i1}, \dots Y_{ip})\}_i$ iid $\sim p$, infer G.

Tree-structured network

Suppose the graph G is a tree T, $p(Y)$ is Markov wrt T iff

$$
\begin{aligned}
p(Y|\theta) &= \prod_j p(Y_j|\theta_j) \prod_{(j,k)\in T} \frac{p(Y_j, Y_k|\theta_{jk})}{p(Y_j|\theta_j)p(Y_k|\theta_k)} \\
&= \prod_{(j,k)\in T} p(Y_j, Y_k|\theta_{jk}) \Bigg/ \prod_j p^{d_j-1}(Y_j|\theta_j)
\end{aligned}
$$

where d_j is the degree (number of neighbors in T) of node j.

Tree-structured network

Suppose the graph G is a tree T, $p(Y)$ is Markov wrt T iff

$$\begin{aligned} p(Y|\theta) &= \prod_j p(Y_j|\theta_j) \prod_{(j,k)\in T} \frac{p(Y_j, Y_k|\theta_{jk})}{p(Y_j|\theta_j)p(Y_k|\theta_k)} \\ &= \prod_{(j,k)\in T} p(Y_j, Y_k|\theta_{jk}) \bigg/ \prod_j p^{d_j-1}(Y_j|\theta_j) \end{aligned}$$

where d_j is the degree (number of neighbors in T) of node j.

Tree structure assumption.

- Consistent (although much stronger) with the usual assumption that the graph is sparse.
- Not true in general, but may be sufficient for the inference on local structures, such as the existence of a given edge.

Maximum likelihood inference (1/2)

Log-likelihood.

$$\log p(Y; \theta, T) = \sum_{(j,k)\in T} \log p(Y_j, Y_k|\theta_{jk}) - \sum_j (d_j - 1) \log p(Y_j|\theta_j)$$

Maximum likelihood inference (1/2)

Log-likelihood.

$$\log p(Y; \theta, T) = \sum_{(j,k) \in T} \log p(Y_j, Y_k | \theta_{jk}) - \sum_j (d_j - 1) \log p(Y_j | \theta_j)$$

Inference:

- continuous part (θ): MLE

$$\widehat{\theta}_j = \arg\max_{\theta_j} \log p(\{Y_{ij}\}_i; \theta_j), \quad \widehat{\theta}_{jk} = \arg\max_{\theta_{jk}} \log p(\{(Y_{ij}, Y_{ik})\}_i; \theta_{jk})$$

- discrete part (T)

$$\widehat{T} = \arg\max_T \sum_{(j,k) \in T} \log \frac{p(Y_j, Y_k | \widehat{\theta}_{jk})}{p(Y_j | \widehat{\theta}_j) p(Y_k | \widehat{\theta}_k)}$$

Maximum likelihood inference (2/2)

Chow & Liu algorithm [2]: Taking

$$f(j,k) = \log p(Y_j, Y_k|\widehat{\theta}_{jk}) - \log p(Y_j|\widehat{\theta}_j) - \log p(Y_k|\widehat{\theta}_k)$$

as the weight of edge (j,k),

$$\widehat{T} = \arg\max_T \sum_{(j,k)\in T} f(j,k)$$

is the maximum spanning tree with weights $\{f(j,k)\}$, which can be retrieved by Kruskal's algorithm in $O(p^2)$ [6].

Maximum likelihood inference (2/2)

Chow & Liu algorithm [2]: Taking

$$f(j,k) = \log p(Y_j, Y_k|\widehat{\theta}_{jk}) - \log p(Y_j|\widehat{\theta}_j) - \log p(Y_k|\widehat{\theta}_k)$$

as the weight of edge (j,k),

$$\widehat{T} = \arg\max_T \sum_{(j,k)\in T} f(j,k)$$

is the maximum spanning tree with weights $\{f(j,k)\}$, which can be retrieved by Kruskal's algorithm in $O(p^2)$ [6].

Retrieves the maximum likelihood tree but with no measure of uncertainty.
$\rightarrow$ Exploring the whole tree space allows to evaluate uncertainty.
$\rightarrow$ Bayesian inference can again be a solution.

Bayesian setting [11]

Model: prior on T: $p(T)$
prior on θ: $p(\theta|T)$ $\rightarrow$ posterior: $p(T|Y)$
likelihood: $p(Y|\theta, T)$

Bayesian setting [11]

Model: prior on T: $p(T)$
prior on θ: $p(\theta|T)$ $\rightarrow$ posterior: $p(T|Y)$
likelihood: $p(Y|\theta, T)$

Prior on T: factorizes over the edges:

$$p(T) \propto \prod_{(j,k) \in T} a(j,k)$$

Bayesian setting [11]

Model:

$$\left.\begin{array}{rl} \text{prior on } T: & p(T) \\ \text{prior on } \theta: & p(\theta|T) \\ \text{likelihood:} & p(Y|\theta, T) \end{array}\right. \quad \rightarrow \quad \text{posterior:} \quad p(T|Y)$$

Prior on T: factorizes over the edges:

$$p(T) \propto \prod_{(j,k)\in T} a(j,k)$$

Prior on θ: displays factorability properties, i.e. needs to satisfy

$$p(\theta_{jk}|T) \equiv p(\theta_{jk}) \quad \text{for all } T \ni (j,k).$$

$\rightarrow$ Compatible family of strong Markov hyper-distributions [4]:
multinomial-Dirichlet (conjugacy), normal-Wishart (conjugacy), Gaussian copulas (numerical integration), ...?

Quantities of interest

Marginal distribution.

$$p(Y) \propto \sum_{T \in \mathcal{T}} \prod_{j,k} \frac{a(j,k) \int p(Y_j, Y_k, \theta_{jk}) \, \mathrm{d}\theta_{jk}}{\int p(Y_j, \theta_j) \, \mathrm{d}\theta_j \times \int p(Y_k, \theta_k) \, \mathrm{d}\theta_k}$$

where $\mathcal{T}$ stands for the set of all spanning trees.

Quantities of interest

Marginal distribution.

$$p(Y) \propto \sum_{T \in \mathcal{T}} \prod_{j,k} \frac{a(j,k) \int p(Y_j, Y_k, \theta_{jk}) \, \mathrm{d}\theta_{jk}}{\int p(Y_j, \theta_j) \, \mathrm{d}\theta_j \times \int p(Y_k, \theta_k) \, \mathrm{d}\theta_k}$$

where $\mathcal{T}$ stands for the set of all spanning trees.

Posterior probability for an edge to be absent.

$$\Pr\{(j,k) \notin T | Y\} \propto \sum_{T \in \mathcal{T} : (j,k) \notin T} \prod_{j,k} \frac{a(j,k) \int p(Y_j, Y_k, \theta_{jk}) \, \mathrm{d}\theta_{jk}}{\int p(Y_j, \theta_j) \, \mathrm{d}\theta_j \times \int p(Y_k, \theta_k) \, \mathrm{d}\theta_k}$$

Quantities of interest

Marginal distribution.

$$p(Y) \propto \sum_{T \in \mathcal{T}} \prod_{j,k} \frac{a(j,k) \int p(Y_j, Y_k, \theta_{jk}) \, \mathrm{d}\theta_{jk}}{\int p(Y_j, \theta_j) \, \mathrm{d}\theta_j \times \int p(Y_k, \theta_k) \, \mathrm{d}\theta_k}$$

where $\mathcal{T}$ stands for the set of all spanning trees.

Posterior probability for an edge to be absent.

$$\Pr\{(j,k) \notin T | Y\} \propto \sum_{T \in \mathcal{T} : (j,k) \notin T} \prod_{j,k} \frac{a(j,k) \int p(Y_j, Y_k, \theta_{jk}) \, \mathrm{d}\theta_{jk}}{\int p(Y_j, \theta_j) \, \mathrm{d}\theta_j \times \int p(Y_k, \theta_k) \, \mathrm{d}\theta_k}$$

Typical form:

$$\sum_{T \in \mathcal{T}} \prod_{(j,k) \in T} f(j,k),$$

with cardinality of $\mathcal{T} = p^{p-2}$.

Summing over spanning trees

Matrix-tree theorem.

- $F = [f(j,k)]$: *a symmetric matrix with* $f(j,j) = 0, f(j,k) > 0$;
- $\Delta = [\Delta_{jk}]$ *its Laplacian:* $\Delta_{jj} = \sum_k f(j,k), \Delta_{jk} = -f(j,k)$.

Summing over spanning trees

Matrix-tree theorem.

- $F = [f(j,k)]$*: a symmetric matrix with* $f(j,j) = 0, f(j,k) > 0$*;*
- $\Delta = [\Delta_{jk}]$ *its Laplacian:* $\Delta_{jj} = \sum_k f(j,k), \Delta_{jk} = -f(j,k)$*.*

Then the minors Δ^{uv} *of* Δ *are all equal to*

$$\sum_{T \in \mathcal{T}} \prod_{(j,k) \in T} f(j,k).$$

Summing over spanning trees

Matrix-tree theorem.

- $F = [f(j,k)]$: *a symmetric matrix with* $f(j,j) = 0, f(j,k) > 0$;
- $\Delta = [\Delta_{jk}]$ *its Laplacian:* $\Delta_{jj} = \sum_k f(j,k), \Delta_{jk} = -f(j,k)$.

Then the minors Δ^{uv} *of* Δ *are all equal to*

$$\sum_{T \in \mathcal{T}} \prod_{(j,k) \in T} f(j,k).$$

- Can be used to compute $p(Y)$, the normalizing constant of $p(T)$, ... at the cost of computing a $p \times p$ determinant.
- Already used in [7] for tree learning.
- Again 'sum-product' in place of 'max-sum'.

Posterior probability of an edge

The existence of an edge between variables Y_j and Y_k can be assessed by

$$\Pr\{(j,k) \in T|Y\} \propto \sum_{T \ni (j,k)} p(T)p(Y|T)$$

which depends on the prior $p(T)$.

The prior probability $\Pr\{(j,k) \in T\}$ can be tuned

- with the prior coefficient $a(j,k)$
- or set to an arbitrary value using an edge-specific probability change.

Posterior probability of an edge

The existence of an edge between variables Y_j and Y_k can be assessed by

$$\Pr\{(j,k) \in T | Y\} \propto \sum_{T \ni (j,k)} p(T) p(Y|T)$$

which depends on the prior $p(T)$.

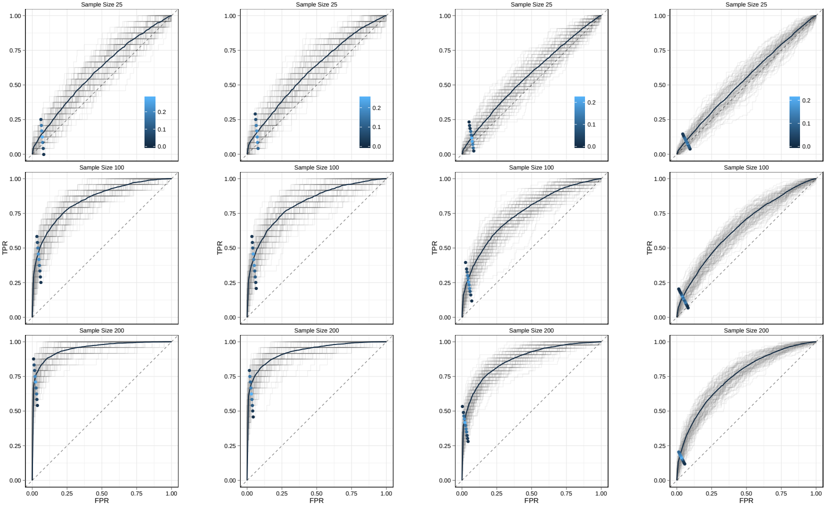
The prior probability $\Pr\{(j,k) \in T\}$ can be tuned

- with the prior coefficient $a(j,k)$
- or set to an arbitrary value using an edge-specific probability change.

All posterior probabilities can be computed in $O(p^3)$.
$\rightarrow$ R package Saturnin (spanning trees used for network inference)

Simulations: ROC curves for edge detection

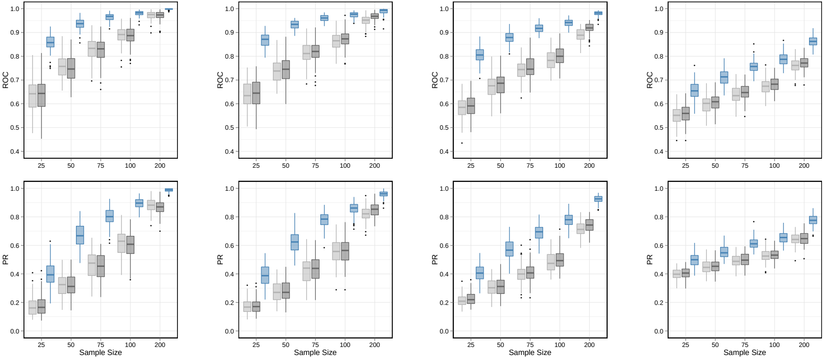
For various graph topologies ($p = 25$, $n = 25, 50, 200$, $B = 100$ simulations)



Tree | Erdös-Rényi $p_c = 2/p$ | Erdös-Rényi $p_c = 4/p$ | Erdös-Rényi $p_c = 8/p$

Simulations: Comparison with sampling among DAGs

[8]: MCMC sampling over the directed acyclic graphs (multinomial case)

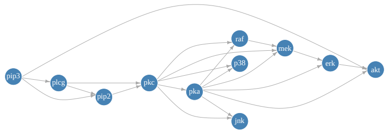


Tree	Erdös-Rényi $p_c = 2/p$	Erdös-Rényi $p_c = 4/p$	Erdös-Rényi $p_c = 8/p$

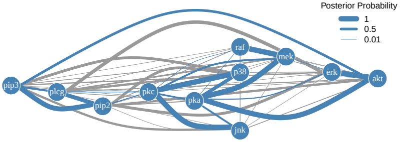
Area under the curves: top=ROC, bottom=PR
light grey = multinomial trees (2.2"), dark grey: multinomial DAGs (1393")

Illustration: Raf pathway

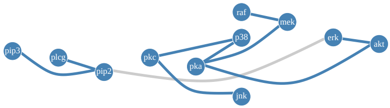
Flow cytometry data for $p = 11$ proteins from the Raf signaling pathway [10]



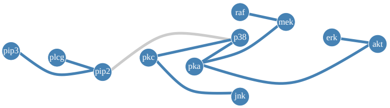
'ground truth'



posterior probabilities



most likely tree



second most likely tree

Outline

Discussion

To summarize.

- Exact Bayesian inference can still be achieved for some fairly complex models with discrete parameter.
- Do not have to care about sampling and convergence.
- No systematic way to check when this is possible $\rightarrow$ ad-hoc developments.

Discussion

To summarize.

- Exact Bayesian inference can still be achieved for some fairly complex models with discrete parameter.
- Do not have to care about sampling and convergence.
- No systematic way to check when this is possible $\rightarrow$ ad-hoc developments.

Future works.

- Combining the two problems: finding change-points in a network structure.
- Dealing with dependency along time.
- Influence of the prior: $p(T)$ depends on n and/or p.
- The exact evaluation of the key quantity raises numerical issues.

References I

Ivan E. Auger and Charles E. Lawrence.
Algorithms for the optimal identification of segment neighborhoods.
Bull. Math. Biol., 51(1):39–54, 1989.

C.K. Chow and C.N. Liu.
Approximating Discrete Probability Distributions with Dependence Trees.
IEEE Transactions on Information Theory, IT-14(3):462–467, 1968.

A. Cleynen and S. Robin.
Comparing change-point location in independent series.
Statistics and Computing, pages 1–14, 2014.

A Philip Dawid and Steffen L. Lauritzen.
Hyper Markov Laws in the Statistical Analysis of Decomposable Graphical Models.
The Annals of Statistics, 21(3):1272–1317, 1993.

Paul Fearnhead.
Exact and efficient bayesian inference for multiple changepoint problems.
Statistics and computing, 16(2):203–213, 2006.

Joseph B. Kruskal.
On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem.
Proceedings of the American Mathematical Society, 7(1):48–50, February 1956.

Marina Meilă and Tommi Jaakkola.
Tractable Bayesian learning of tree belief networks.
March 2006.

Teppo Niinimaki, Pekka Parviainen, and Mikko Koivisto.
Partial order mcmc for structure discovery in bayesian networks.
In Fabio Gagliardi Cozman and Avi Pfeffer, editors, *UAI*, 2011.

References II

G. Rigaill, E. Lebarbier, and S. Robin.
Exact posterior distributions over the segmentation space and model selection for multiple change-point detection problem.
Stat. Comp., 22:917–29, 2011.
DOI: 10.1007/s11222-011-9258-8.

Karen Sachs, Omar Perez, Dana Pe'er, Douglas A Lauffenburger, and Garry P Nolan.
Causal protein-signaling networks derived from multiparameter single-cell data.
Science (New York, N.Y.), 308:523–529, 2005.

L. Schwaller, S. Robin, and M. Stumpf.
Bayesian Inference of Graphical Model Structures Using Trees.
Technical report, April 2015.
ArXiv:1504.02723.