Inference for Partial Orders from Random Linear Extensions

Alexis Muir Watt
Worcester College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Trinity 2015

This thesis is dedicated to all members of my family.

Acknowledgements

I would like to thank Dr G.K. Nicholls for his guidance as my supervisor and the Engineering and Physical Sciences Research Council and the Department of Statistics for their support. I would also like to thank many members of the Department for providing a pleasant and productive research environment. Finally I would like to thank Jan Boylan and the computing support staff for their advice and assistance over the years.

Inference for Partial Orders from Random Linear Extensions

Alexis Muir Watt

Worcester College
University of Oxford

A thesis submitted for the degree of Doctor of Philosophy

Trinity 2015

The study of random orders and of rankings models has attracted much work in the combinatorics and statistics literature. However, there has been little focus on random partial order models for statistical modelling. A partial order on a set P corresponds to a transitively closed, directed acyclic graph h(P) with vertices in P. Such orders generalize orders defined by partitioning the elements of P and ranking the elements of the partition. Observed orders are modelled as random linear extensions of suborders of an unobserved partial order h(P) evolving according to a stochastic process, and inference on the partial order is of interest.

Chapter 1 reviews some static random partial order models. Of particular interest for modelling is a latent variables model based on the random k-dimensional orders and a parameter controlling the mean depth of a partial order. In Chapter 2, it is extended to a stochastic process on latent variables to describe a partial order evolving continuously in time. As a Hidden Markov model, the process is observed by taking random linear extensions from suborders of the partial order at a sequence of sampling times. The posterior distribution for the unobserved process is doubly-intractable. The basis for a numerical inference algorithm in Chapter 3 is Particle MCMC with an efficient particle filtering transition distribution on the latent variables. Sampling latent variables relates to the well studied problem of estimating multivariate normal orthant probabilities, for which Chapter 4 gives a new importance sampler. It is competitive with existing samplers under some conditions on the covariance. Inference on the partial order process is computational, and Chapter 5 gives some numerical algorithms to reduce the complexity of some common latent variables computations. Lastly, Chapter 6 applies Chapter 2 and 3 to dynamic ranking problems in the areas of historical research and sport tournaments.

Contents

List of Figures

Chapter 1

Introduction

A motivation for studying random partial order models in this thesis comes from their applications to the statistical modelling of ranking data. A partial order on a set of elements corresponds to a directed acyclic graph. Figure 1.1 shows a partial order on 5 elements. A total order is a set of elements fully ordered, like on Figure 1.2. In applications, a total order could be viewed as a ranking. On the figures shown, the partial order is an intersection in some sense of all three total orders: only the consistent orders in the total orders appear in the partial order. As a ranking model, the partial order represents the true, unobserved order.

A more standard model for these total orders is the Plackett-Luce model of Luce [1959] and Plackett [1975], where each element is given a weight which determines its average rank in the total orders. For example on Figure 1.2, elements 1 and 5 would have high and low weight respectively. However, it is not clear what value should be assigned to element 2: it should not be much different from the weights of elements 3 and 4, otherwise the order of 2 with these elements would not be as variable. Yet, the weights for elements 3 and 4 should be different because they are ordered in all observations. This difficulty arises because the model is one-dimensional: each element corresponds to one scalar quantity. But it could be circumvented with a multi-dimensional model, where elements are each assigned several weights such that the weights of element 2 are not uniformly larger or smaller than those of elements 3 and 4. This is essentially a partial order model.

Chapter 1 reviews some existing static random partial order models from the combinatorics literature. Looking at these from the perspective of statistical modelling, it argues in favour of a latent variables model with parameters controlling the mean partial order depth (hierarchy). A model with controllable depth is important in ranking applications where data shows some ordering between at least some elements, because most partial orders have low depth.

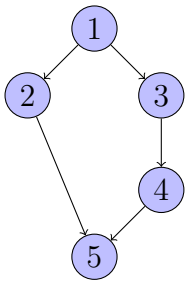

Figure 1.1: A partial order on five elements.

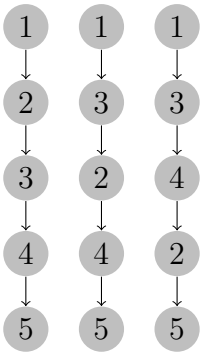

Figure 1.2: A set of three total orders on five elements.

In Chapter 2, it is extended to a new stochastic process in time on partial orders via the latent variables. In applications, observed orders are modelled as random linear extensions of suborders of an unobserved partial order process at a sequence of uncertain sampling times. The process on latent variables is tractable and can be discretized at these sampling times. The latent variable process and the linear extension observation model together form a Hidden Markov model (HMM).

Performing inference on the partial orders and the HMM static model parameters is a doubly-intractable problem. The basis for a numerical inference algorithm in Chapter 3 is Particle MCMC with an efficient particle filtering transition distribution on the latent variables. It involves sampling the latent variables with some constraints, which coincidentally relates to the well studied problem of estimating multivariate normal orthant probabilities. Chapter 4 thus uses the sampler for a new importance sampling estimate that is competitive with existing samplers when the variables are highly correlated.

Since the inference is of computational nature, its implementation matters. Chapter 5 suggests some new numerical algorithms to perform some common partial order computations by making use of the latent variable structure of the model. When the number of elements is large, its running time is lower than that of some simpler

algorithms.

Lastly, Chapter 6 uses the partial order model for dynamic ranking problems in the areas of historical research and sport tournaments. It shows the possibilities offered by the model and the inference algorithm developed in this thesis.

1.1 Background on random partial order models

This section lays out the terminology for partial orders and lists some existing partial order models: the uniform on partial orders, the (K, n)-dimensional order and variants such as the (K, ρ, n)-dimensional order, and the $G_{n,p}$-graph order. This will be followed by a glossary to review the notation and a comparison between models. It will be argued that the (K, ρ, n)-dimensional order is the preferred model for some statistical modelling applications. It will be extended to a dynamic model on partial orders, the main model of interest in this thesis. The uniform and the $G_{n,p}$-graph order models are only discussed to provide a contrast.

1.1.1 Partial order definitions

In the following $P = (A, \prec)$ denotes a partial order with relation $\prec$ over the set A. n denotes the size of A. $\prec$ is an irreflexive, asymmetric and transitive binary relation. $\forall (a, b, c) \in A^3$, if $a \prec b$ then neither $b \prec a$ nor $a \prec a$ can hold, and $a \prec b$ and $b \prec c$ implies $a \prec c$. $a \parallel b$ denotes that elements a and b have no order (neither $a \prec b$ nor $b \prec a$).

When there are several partial orders discussed, $x \prec_P y$ is used to clarify that the relation is with respect to P. A has n elements $a_1, \ldots, a_n$, but typically $A = \{1, \ldots, n\}$.

P corresponds to a directed acyclic graph or DAG D with vertices labelled as the elements of A. Although the DAG D of a partial order is a transitively closure where all relations implied by transitivity have a correspond edge, in the following some partial order DAGs will be displayed as a transitively reduction for a clearer visualization as on Figure 1.3. For a general DAG D, N_c denotes the number of edges in its transitive closure and N_r the number of edges in its transitive reduction. There are $2^{N_c - N_r}$ DAGs that correspond to P up to edges implied by transitivity.

The adjacency matrix denoted by M of $P = (A, \prec)$ is the matrix with rows and columns corresponding the elements of A, with values in $\{0, 1\}$ such that $M_{i,j} = 1$ if and only if $a_j \prec a_i$. In particular, M corresponds to a transitively closed graph.

A *linear order* or a *total order* is a partial order where all elements are ordered, for example $a_1 \prec \ldots \prec a_n$ is a linear order on $A = (a_1, \ldots, a_n)$.[1] For $o \subseteq A$, a linear order $y = (o, \prec_y)$ *extends* $P = (A, \prec)$, denoted by $P \uparrow y$, if its orders respect all orders of P, $a \prec_P b \implies a \prec_y b$ for $(a, b) \in o^2$.

If a linear order y is of length n (orders all elements of A) and extends P, it is a linear extension of P, $y \in \mathcal{L}(P)$. If y is on a subset $o \subseteq A$, then it is a linear extension of the suborder $P[o]$. Figure 1.4 enumerates linear extensions of a partial order. The empty partial order has $n!$ linear extensions, whereas a total order has a single linear extension (itself).

A chain is a suborder of P which is also a total order, informally it is an extract of P with ordered elements only. The length of the longest chain is the depth.

An intersection of k linear orders is the partial order P such that an order belongs to P if and only if it is an order of all the k linear orders. The dimension of P, $\dim(P)$, is the minimum number of linear orders on A whose intersection is P. A set of $\dim(P)$ linear orders on A with intersection equal to P is a decomposition of P. In general, a decomposition is not unique. All of the linear orders in a decomposition belong to $\mathcal{L}(P)$, and the intersection of all linear extensions in $\mathcal{L}(P)$ is also P.

Figure 1.4 gives an example of linear extension, chain, depth and dimension.

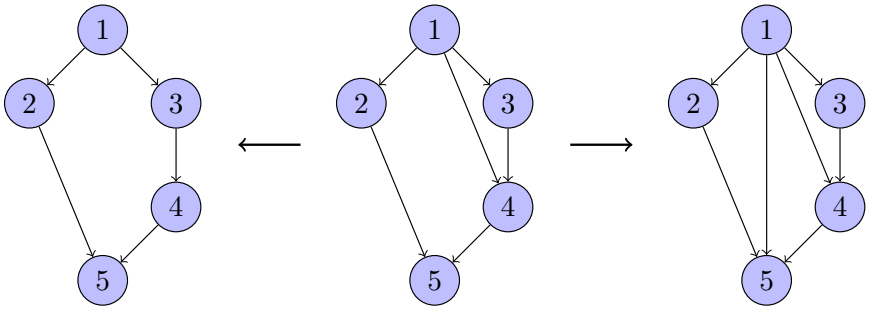

Figure 1.3: A DAG (middle) with its transitive reduction (left) and closure (right).

1.1.2 Sampling partial orders uniformly

Let $\mathcal{P}_n$ denote the set of partial orders on $\{1, \ldots, n\}$. A model on a partial order $P = (A, \prec)$ is a distribution on P for a fixed set A of size n. The following discusses the properties of the uniform on $\mathcal{P}_n$ as a prior model on partial orders. It will show why the uniform model is not likely to be useful for modelling purposes.

[1] A total order can be thought of as a permutation. The sequence $(a_1, \ldots, a_n)$ can represent as permutation σ, where $\sigma(a_i) = a_{i+1}$, or a total order y, where $a_i \prec_y a_j$. Alternatively, the permutation could be defined at $\sigma(i) = a_i$, and the total order as $i \prec_y j$ iff $a_i < a_j$.

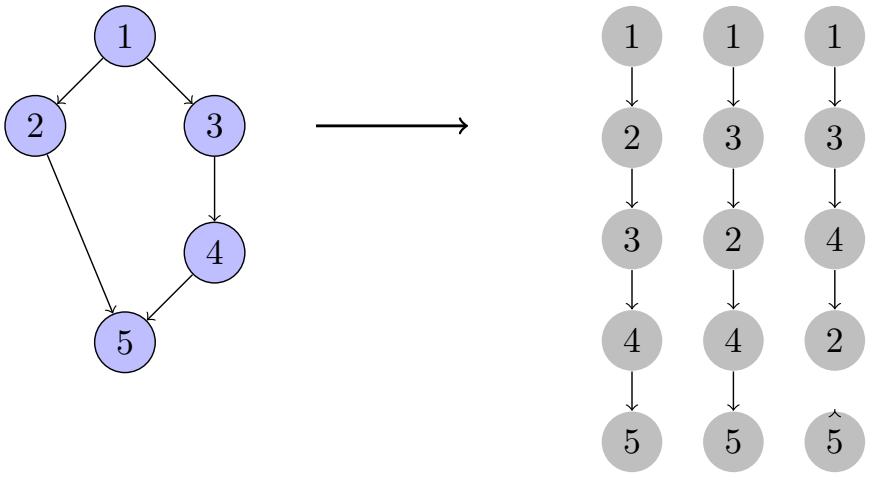

Figure 1.4: A partial order P (left) and the set $\mathcal{L}(P)$ of its three linear extensions (right). The longest chain of P is $5 \prec 4 \prec 3 \prec 1$, implying a depth of 4. The dimension is 2 because the intersection of just two linear extensions equals P: the first and the third.

Some model properties of interest are for example the typical structure of a sampled P. Some statements are made about properties as n is large. In this chapter, a probabilistic statement such as "$P \sim \mathcal{U}[\mathcal{P}_n]$ has almost surely property ... as $n \to \infty$" means that the probability that P has such property as $n \to \infty$ is 1.

Proposition 1 (Kleitman and Rothschild [1975]). *The number of partial orders on n elements is*

$$|\mathcal{P}_n| = 2^{n^2/4 + \frac{3n}{2} + \mathcal{O}(\log(n))}$$

and almost all partial orders from $\mathcal{U}[\mathcal{P}_n]$ have depth of 3 as $n \to \infty$.

It is also shown that almost all partial orders are 3-layered, meaning that there exists a partition of A into three subsets A_1, A_2 and A_3 such that if $a_i \prec a_j$ with $a_i \in A_i$ and $a_j \in A_j$ then $i < j$.

These results show that firstly, $\mathcal{P}_n$ is a large set. Algorithms discussed in this thesis do not resort to enumerations over $\mathcal{P}_n$. Secondly, depth 3 and therefore the uniform cannot serve as model for partial orders of high depth. In this thesis, high depth typically means close to n, or at least an increasing function of n. Thirdly, the 3-layered structure cannot serve as model for more general structures, for example allowing for more asymmetry or for some subsets to be unordered between each other.

MCMC sampling of the uniform The MCMC Algorithm 1 aims to sample from $\mathcal{U}[\mathcal{P}_n]$. It is based on a random walk on DAGs. Although $\mathcal{U}[\mathcal{P}_n]$ as a model may not

posses the properties sought after in this thesis, it is instructive from an algorithmic perspective as it shows a connection between DAGs and partial orders. Sampling a DAG uniformly at random and then computing its transitive closure does not lead to a uniformly random partial order.

Algorithm 1 (Nicholls et al. [2010]) MCMC for $\mathcal{U}[\mathcal{P}_n]$

Given a DAG $D^{(m)}$ at iteration m:
Set $D^* = D^{(m)}$. Sample $(i, j) \in \{1, \ldots, n\}^2$ and set $D^*_{i,j} = 1 - D^*_{i,j}$.
Set $\alpha = 0$ if D^* is cyclic, otherwise set

$$\alpha = \min\left\{1, \frac{2^{N_c^{(m)} - N_r^{(m)}}}{2^{N_c^* - N_r^*}}\right\}$$

Sample $u \sim \mathcal{U}[0, 1]$
if $u < \alpha$ **then**
  Set $D^{(m+1)} = D^*$.
else
  Set $D^{(m+1)} = D^{(m)}$.
end if

Proof. The MCMC chain targets $\pi(P, D) = \pi_P(P)\pi_D(D|P)$ where π_P is the uniform $\mathcal{U}[\mathcal{P}_n]$ of interest and

$$\pi_D(D|P) = \frac{\mathbb{1}_{D^c = P}}{2^{N_c - N_r}}$$

is the uniform distribution on DAGs D corresponding to P, where D_c and N_c denote the transitive closure of D and its number of edges, and similarly for D_r and N_r. The proposal is $q(P^*, D^*|P, D) = q_P(P^*|D^*)q_{D^*|D}(D^*|D)$ where $q_P(P^*|D^*) = \mathbb{1}\{D^*_c = P^*\}$ and $q_{D^*|D}(D^*|D)$ is symmetric. Thus, the acceptance probability reduces to

$$\alpha = \min\left\{1, \frac{\pi_D(D*|P*)}{\pi_D(D|P)}\right\}$$

and thus to expression in the algorithm.

It is irreducible because a walk starting from any DAG D_0 can reach any other DAG D_1 by removing all edges of D_0 and then adding only edges of D_1. Since the marginal of $\pi(P, D)$ over D is $\mathcal{U}[\mathcal{P}_n]$, taking the transitive closure of the output $\{D_c^{(m)}\}$ targets $\mathcal{U}[\mathcal{P}_n]$. □

1.1.3 Partial order models of controllable depth

The fact that a typical partial order from the uniform distribution has low depth motivates the search for models with depth controllable by a parameter. These are the

focus of this thesis. Although many models put too much probability on partial orders of low depth, some models presented here offer the possibility to adjust the typical depth of a random partial order. These models will then be compared according to criteria other than depth distribution.

A survey of classical models is found in Brightwell [1993] and of (K, n)-dimensional random orders in Winkler [1985]. As terminology, the models presented are all *families* of distributions, indexed by parameters such as n. A partial order is sampled from a given model and its given model parameters.

1.1.3.1 (K, n)-dimensional random order model

Definition Winkler [1985] introduces the (K, n)-dimensional random order which is an intersection of K random total orders. Related to this idea is the fact that any partial order P can be written as the intersection of its linear extensions $\mathcal{L}(P) = \{L_1, \ldots, L_K\}$. However, a partial order may be written as the intersection of much fewer total orders (equivalently here, linear extensions), for example the null partial orders is extended by all $n!$ chains whereas it can be written as the intersection of just two total orders. For a fixed integer K, a (K, n)-dimensional random order P satisfies

$$\dim(P) \leq K \leq |\mathcal{L}(P)|. \tag{1.1}$$

Choice of K In the (K, n)-dimensional random order model, total orders $L_1, \ldots, L_K$ on A are sampled independently and uniformly at random (there are $n!$ of them, $n = |A|$). The support and depth distribution both depend on the parameter K. A lower value of K decreases the probability that an order is common to the K orders, therefore samples tend to have higher depth. However, if K is too small then the support becomes smaller than $\mathcal{P}_n$: some partial orders on A are not covered by the model.

From a modelling perspective, if the aim is generality and therefore support on a large set such as $\mathcal{P}_n$, then K should be large enough. However, if K is unnecessarily large then inference algorithms may suffer from the increased dimension. Clearly, $K = n$ is sufficient to include $\mathcal{P}_n$ as support: by induction, if $K = n - 1$ ensures support on $\mathcal{P}_{n-1}$, then adding an order L_n and increasing the lengths of the orders to $|L_i| = n$ gives enough room to specify the order between an n-th element and the previous $n - 1$ elements.

Is $K \geq n$ necessary to ensure support on $\mathcal{P}_n$? On one hand, Hiraguchi [1951] showed that when $n \geq 4$, $\dim(P) \leq [n/2]$ for all $P \in \mathcal{P}_n$. On the other hand, it

is possible to find elements in $\mathcal{P}_{2n}$ that have dimension n, for example the so-called "crown" partial order on $a_1, \ldots, a_n, b_1, \ldots, b_n$ such that the only orders are $a_i \prec b_j$ for $i \neq j$. This leads to the following result motivating the typical choice of $K = [n/2]$ in this thesis:

Proposition 2. *$K = [n/2]$ is necessary and sufficient for the (K, n)-dimensional order model to have support on $\mathcal{P}_n$.*

Latent variables Winkler [1985] remarks that a (K, n)-dimensional random order can alternatively be defined by n points $z^{1:n}, z^j = (z_1^j, \ldots, z_K^j)$ sampled uniformly at random in the unit hypercube $[0, 1]^K$ with $P = (A, \prec)$ such that $a_j \prec a_j$ if and only if $\forall k \in [1, K], z_k^i > z_k^j$. Thus, a (K, n)-dimensional random order can be thought of as the intersection of either K total orders (of which there are $n!$ to choose from) or of n points z^j in the hypercube $[0, 1]^K$. The points z^j will be referred to as *latent variables* as they map to a partial order $P(z)$.

1.1.3.2 (K, ρ, n)-dimensional order model

A small K would not eliminate the issue of low depth entirely. The probability that the (K, n)-dimensional order is total (maximum depth of n) is the probability that all $L_1, \ldots, L_K$ are all equal, $1/n!^{K-1}$: the depth is likely to be low compared to for example the depth of a total order, n.

This shortcoming of the original (K, n)-dimensional random order model Nicholls et al. [2010] in terms of depth distribution could be overcome by introducing some correlation between the orders $L_1, \ldots, L_K$. Or alternatively, in the latent variable interpretation, by some correlation between the K values $[0, 1]^K$ of each $z^j = (z_1^j, \ldots, z_K^j)$. This latent variable version will be prove convenient from a modelling perspective.

The (K, ρ, n)-dimensional order is a partial order determined by latent variables $z^i \overset{iid}{\sim} f_{K,\Sigma}$ with some covariance Σ. $Z \in \mathcal{M}_{n,K}(\mathbb{R})$ denotes the matrix of K *features* (columns) and n elements (rows). The i-th row is the latent variables sample $z^i \overset{iid}{\sim} f_{K,\Sigma}$. The partial order $P(Z)$ is determined by $a_j \prec a_i$ if $\forall k \in [1, K], z_k^i > z_k^j$, as shown on Figure 1.5.

$f_{K,\Sigma} = \mathcal{N}_K(0, \Sigma)$ will be used throughout this thesis. It will prove a convenient choice in the inference algorithms. The covariance will be equi-correlated with unit variance and equi-correlated variables: $\Sigma = \Sigma_\rho$ with $\Sigma_{i,i} = 1$ and $\Sigma_{i,j} = \rho$ for $i \neq j$. ρ is the *depth parameter* and controls the typical depth of $P(Z)$. For a distribution on high depth, a prior distribution on ρ assigns values close to 1 with high probability. With $\rho = 1$, a sample is almost surely a total order (depth n).

Rows z^i are *paths*, because when plotted they typically look flat if the depth parameter ρ is high. Two elements in $P(Z)$ are incomparable if their paths cross. Clearly, the probability that two fixed elements a_1 and a_2 are incomparable, ie $a_1 \parallel a_2$ and their paths z^1 and z^2 cross, is equal to the so-called *orthant probability* $\mathbb{P}(w > 0)$ that a sample from $w \sim \mathcal{N}_K(0, 2\Sigma)$ falls entirely into the upper (or lower) quadrant.

The probability of a (K, ρ, n)-dimensional order P is determined by the density of Z with

$$\mathbb{P}(P|K, \rho, n) = \int_{\mathbb{R}^{Kn}} \mathbb{1}\{P(Z) = P\} p(Z|K, \rho, n) dZ. \tag{1.2}$$

where $p(Z|K, \rho, n)$ is the distribution on the latent variable matrix Z.

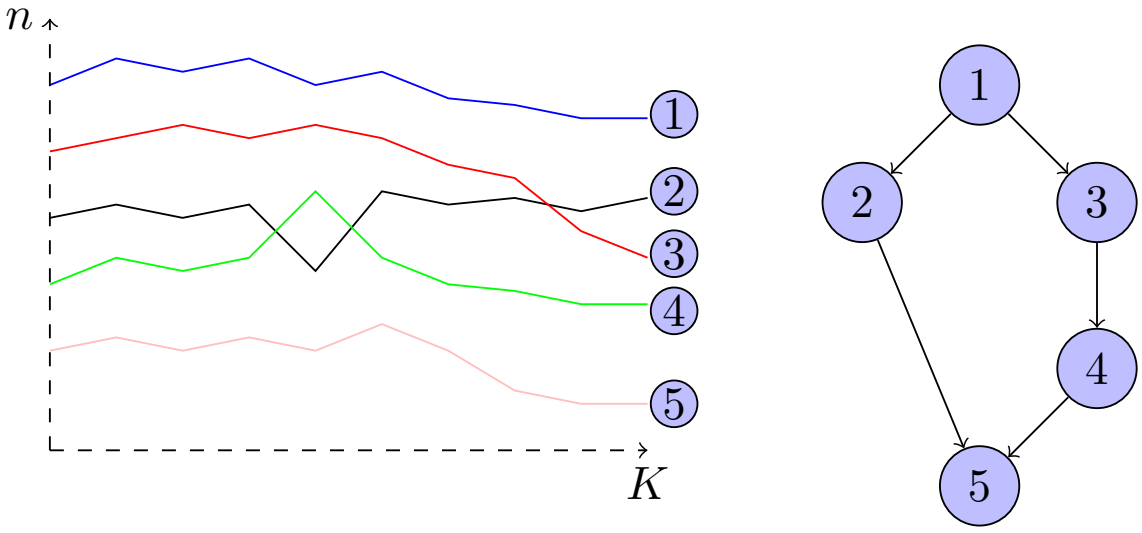

Figure 1.5: Latent variables Z and partial order $P(Z)$ with $n = 5$. For example, $2 \parallel 3$ and $2 \parallel 4$ because the black line crosses the red and green lines.

Alternative latent variables models Putting more weight on partial orders of higher depth can also be achieved with the latent variables z^j from the original distribution $z^j \overset{iid}{\sim} [0, 1]^K$, but with a different rule to determine orders from z^j.

Although only studied in the $K = 2$ case, Balister and Patkos [2010] introduces a generalization here referred to as a (K, α, n)-dimensional order where the order $z^i \prec z^j$ occurs if z^j falls into the open angular domain defined by the point z^i and a given angle $0 \leq \alpha \leq \pi$ which bisector line is parallel to the bisector at the origin. $\alpha = \pi/2$ uncovers the standard case, whereas $\alpha = 0$ and $\alpha = \pi$ almost surely results in partial orders with no comparison and with total order respectively. This is equivalent to sampling points uniformly in some rhombus and proceeding with the usual $\alpha = \pi/2$ ordering, referred to as the (K, α, n)-rhombus order model.

These extensions give control over the depth distribution with the angle parameter α. Compared to the (K, ρ, n)-dimensional order model, they appear less natural as models for the applications of interest.

1.1.3.3 $G_{n,p}$-graph order model

A standard distribution on graphs on n vertices is to set each edge (i, j) independently with (Bernoulli) probability p. Sampling edge orientations independently is not as straightforward for the purpose of sampling a partial order, because transitivity of $\prec$ implies non-independence of edges.

However, given sampled edges, a graph can be converted to a partial order on A by ordering $a_i \prec a_m$ if there exists an increasing sequence of edges $(i, j), (j, k), \ldots, (l, m)$ (from vertices i to j, j to k, etc, l to m, with $i < j < k < \ldots < m$) in the graph. For example if there is an edge between vertices 1 and 3 and between vertices 3 and 4, then $a_1 \prec a_4$, even if there is no edge between 1 and 2. Figure 1.6 shows some realizations

The result is a partial order with edges such that $a_i \prec a_j \implies i < j$, but a random permutation applied to the elements removes this restrictions.

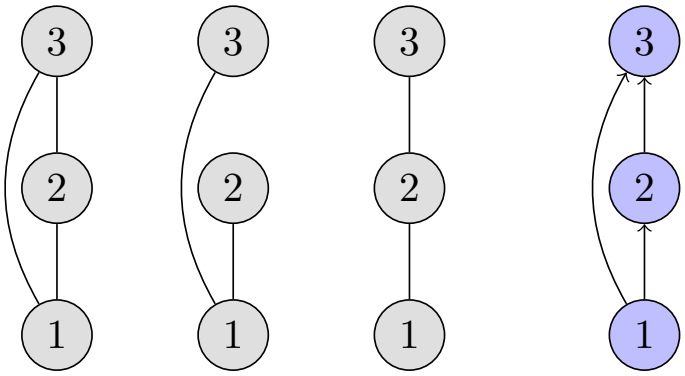

Figure 1.6: The set of 3 graphs leading to the order on the right after directing edges.

1.2 Glossary

Having introduced the main partial order models, the following offers a glossary of the notation and concepts to be used throughout this thesis:

Partial order symbols

M Adjacency matrix, usually of the transitively closed graph of a partial order. 3

$P \uparrow y$ Order $y = (o, \prec_y), o \subset A$ extends $P = (A, \prec_P)$ if orders in y do not conflict with those in P, ie $a \prec_P b \implies a \prec_y b, (a, b) \in o^2$. 4, 12

$P = (A, \prec)$ Partial order on the set $A = \{a_1, \ldots, a_n\}$ with order $\prec$. 3

$P[o]$ suborder induced on elements o of partial order $P = (A, \prec)$ and subset $o \subseteq A$. 4

Z Latent variables matrix in $\mathcal{M}_{n,K}(\mathbb{R})$ for n elements (rows) and K features (columns). 8, 96

$\mathcal{L}(P)$ Set of linear extensions of P. 4

$\mathcal{M}(P)$ maximal set of P. v, 16, 17, 90

$\mathcal{M}_{n,K}(\mathbb{R})$ Set of real-valued matrices with n rows and K columns. 8

$\mathcal{P}_n$ Set of partial orders on n elements. 4

n Number of elements $n = |A|$ of partial order $P = (A, \prec)$. 3

Partial order definitions

adjacency matrix Matrix M of a graph $G = (V, E)$ such that $M_{i,j} = 1$ if $(v_i, v_j) \in E$. For a partial order $P = (A, \prec)$, this corresponds to $M_{i,j} > 0$ if and only if $a_j \prec a_i$. 3, 14

DAG Directed and acyclic graph. A DAG of a partial order is transitively closed. 3

depth Length of the longest chain(s) of a given partial order. 4, 16

dimension Minimum number of chains whose intersection is P. 4, 22

intersection An intersection of linear orders $y_k = (A, \prec_{y_k})$ is the partial order $P = (A, \prec)$ such that $a_i \prec a_j$ iff $a_i \prec_{y_k} a_j, \forall k$. 4, 7, 23, 29, 34, 111, 115

linear extension A linear order y of length n of a partial order $P = (A, \prec)$ such that $P \uparrow y$, ie $a \prec_P b \implies a \prec_y b$. 4, 11, 15

maximal set Set of elements a such that there is no element b with $a \prec b$. 27

suborder Restriction of a partial order $P = (A, \prec)$ to a subset $o \subseteq A$ inheriting all orders in P between elements in o. 11, 14, 16

transitively closure DAG obtained by adding edges implied by transitivity. 3, 12

transitively reduction DAG obtained by removing redundant edges of a given DAG D implied by transitivity, it is the unique DAG with the least edges which closes to the transitive closure of D. 3

width Length w of the largest subset of elements with no order between any of its elements, for example $w = 1$ for a total order and $w = n$ for the null order.. 16, 18, 19

Partial order models

(K, ρ, n)-dimensional order $a_i \prec a_j$ iff $Z_{i,1:K} < Z_{j,1:K}$, with $Z_{i,1:K} \overset{iid}{\sim} \mathcal{N}_K(0, \Sigma)$, $\Sigma_{i,i} = 1$ and $\Sigma_{i,j} = \rho$ for $i \neq j$. 3, 13–15, 22, 23, 30, 34, 49, 86, 87, 109

(K, n)-dimensional order Intersection of K iid chains from the uniform over permutations on n elements. 3, 14, 15

$G_{n,p}$-graph order Edges $a_i \prec a_j$ with $i < j$ are set independently with probability p, then converted to a partial order by taking the transitively closure. 3, 22, 109

1.3 Properties of the latent variables model

Partial order models presented in Section 1.1.3 have controllable depth. However, other properties such as generality are of interest in this thesis. If partial order samples from a model have a restrictive structure, then it may or may not be suitable as a prior in modelling applications. Other properties such as complexity of sampling are also discussed, however this thesis aims for generality rather than low complexity and this motivates a chapter on inference algorithms of computational nature. For this reason, it will become clear from the following discussion why the (K, ρ, n)-dimensional order model is preferable to the $G_{n,p}$-graph order model.

1.3.1 Model interpretation

There exists a plausible *physical process* or interpretation of the (K, ρ, n)-dimensional order. Latent variables Z from a (K, ρ, n)-dimensional order can be interpreted as scores $z^i = (z_1^i, \ldots, z_K^i)$ of each element a_i on K features. These features may have a physical meaning as covariates, for example if fruit has features sweetness and crispiness then grapefruit $\prec$ apple, but apple and orange are unordered if apple is crispier but less sweet. In this thesis these features are virtual. The Z scores $z^i = (z_1^i, \ldots, z_K^i)$ of each element $a_i \in A$ on the K features determine the order of the elements a_i and a_j on each feature. The feature orders are sufficient to determine the partial order $P(Z)$.

For a fixed K, Equation 1.1 states that (K, n)-dimensional orders have dimension $\leq K$, therefore they are a model for when elements a_i are believed to have at most K different features. In (K, ρ, n)-dimensional orders, ρ is a parameter for the correlation between latent variable, and increasing ρ reduces the typical partial order dimension (in the extreme case, $\rho = 1$ and $K = 1$ are interchangeable). Thus, while ρ affects the average dimension, K set large enough guarantees a desired coverage.

1.3.2 Partial order structure

As seen in Proposition 2, if $K \geq [n/2]$ then the (K, n)-dimensional order has support on $\mathcal{P}_n$. Clearly this also applies to (K, ρ, n)-dimensional orders if $\rho < 1$ (if $\rho = 1$, partial orders are almost surely total).

An *isolated* element a_i is an element unordered with every other element, i.e. $a_i \parallel a_j, \forall a_j \in A$. Winkler [1985] shows that almost every (K, n)-dimensional partial order sample has no isolated elements as $n \to \infty$. Clearly this also applies to (K, ρ, n)-dimensional order if $\rho > 0$.

Since this thesis focusses on partial order models of controllable depth, it is of interest to know the average depth for a given n. For (K, n)-dimensional orders, Winkler [1985] shows that almost all partial order has depth between $cn^{1/K}$ and $en^{1/K}$, for some c such that $0 < c < e$, as $n \to \infty$. Are depth distributions of the (K, n)-dimensional orders and (K, ρ, n)-dimensional orders similar? Introducing some correlation $\rho > 0$ for fixed K increases the average depth, and so does decreasing K in the absence of correlation ($\rho = 0$). However, for (K, n)-dimensional orders, at just $K = 2$ the typical depth is small (a sharp drop compared to a depth of 1 when $K = 1$), and smaller as K increases. In contrast, the depth distribution for the (K, ρ, n)-dimensional orders varies smoothly as a function of ρ.

1.3.3 Marginal consistency

$p_o(h')$ denotes a *family* of distributions, with $o \subseteq A$, $|o| = k$, $|A| = n$ and $h' \in \mathcal{P}_k$. It is *marginal consistent* if $p_o(h')$ is the marginal over $p_A(h)$, $h \in \mathcal{P}_n$ obtained by summing over partial orders h on A which have h' as a suborder, ie

$$p_o(h') = \sum_{h|h[o]=h'} p_A(h).$$

It can be shown that the (K, n)-dimensional model family is marginal consistent. This extends to latent variable models such as (K, ρ, n)-dimensional orders for which features are independent across elements. Although not essential in a subjective Bayesian analysis, this property is often desirable.

1.3.4 Complexity of sampling

This section gives a description of the model in terms of algorithms. The aim is to compute the adjacency matrix M representation of a partial order sampled from the model. The running time of sampling independent samples is a function of the number of elements n and latent variable features K. A more elaborate algorithm making use of the latent variable structure will be presented in Chapter 5. Ultimately, independent sampling is only partly of interest in this thesis.

A straightforward way to simulate a (K, ρ, n)-dimensional order order is by simulating the latent variables for each elements before determining the edges by looping through all element pairs, as in Algorithm 2. A (K, n)-dimensional order can be simulated with Algorithm 2 with $f_{K,\Sigma} = \mathcal{U}[0, 1]^K$ simulated the latent variable (K, n)-dimensional model. It requires nK uniform $\mathcal{U}[0, 1]$ samples and K comparisons for each of $n(n-1)/2$ pairs.

A (K, ρ, n)-dimensional order can be simulated with $f_{K,\Sigma} = \mathcal{N}_K(0, \Sigma)$, for example multiplying a vector of K samples $\mathcal{N}(0, 1)$ with L from the Cholesky decomposition $\Sigma = LL^T$. Compared to the previous uncorrelated $\mathcal{U}[0, 1]^K$ case, the extra cost comes from the K standard normal $\mathcal{N}(0, 1)$ samples and $K(K+1)/2$ multiplications, for each of the n elements. Only one initial Cholesky decomposition is needed. Clearly, a different covariance structure could reduce the cost, for example $\mathcal{O}(nK)$ with a serial correlation where $z^i_{k+1}|z^i_k \sim \mathcal{N}(z^i_k, \epsilon)$ for the K features of element a_i.

As an aside: can alternative formulations of these latent variable models reduce the complexity? Unfortunately, the version of the (K, n)-dimensional order defined

Algorithm 2 (K, ρ, n)-dimensional order sampling

```
Allocate Z ∈ M_{n,K}(ℝ)
Allocate M ∈ M_{n,n}({0,1}) with initial values 0
for i ∈ [1,n] do
    Simulate Z_{i,1:K} ~ f_{K,Σ}
end for
for i ∈ [2,n] do
    for j ∈ [1,i−1] do
        Get signs s_{1:K} of Z_{i,1:K} − Z_{j,1:K}
        if all s_k < 0 then
            Set M_{j,i} = 1
        else
            if all s_k > 0 then
                Set M_{i,j} = 1
            end if
        end if
    end for
end for
Return M
```

in terms of an intersection of K total orders has similar complexity.[2] A potential benefit of the (K, α, n)-dimensional order over the (K, ρ, n)-dimensional order is that sampling multivariate correlated samples is not required.[3]

1.3.5 Counting linear extensions

Counting linear extensions fast will be important for inference as it is needed to compute a likelihood: the number of linear extensions enters the description of an observation model. Such count may have other uses elsewhere in statistics, for example as a measure of partial order complexity (an alternative is depth).

The main purpose of the following is to review an elementary algorithm for an exact count on an arbitrary $P \in \mathcal{P}_n$ in $\mathcal{O}(n!)$ in worst-case, but with shorter average

[2] That is, by sampling K permutations of length n independently, each costing $\mathcal{O}(n)$ (with the *Knuth shuffles* algorithm of Knuth [1969]), and intersecting them after $Kn(n-1)/2$ comparisons.

[3] A (K, α, n)-dimensional order can be sampled by Algorithm 2 with $f_{K,\Sigma} = \mathcal{U}[0,1]^K$, except that the orders are not determined by the signs $s_{1:K}$ of $Z_{i,1:K} - Z_{j,1:K}$. When $K = 2$, the original definition is: $a_j \prec a_i$ if $Z_{j,1:K}$ lies in the angular domain at $Z_{i,1:K}$ of angle α. If α is larger, then $a_j \prec a_i$ is more likely and average depth increases. When $K > 2$, here is a suggestion for a definition: the angular domain is the inside of the cone with vertex $O = Z_{i,1:K}$ with unit vector $\vec{u}$ parallel to the angle bisector (average of the plane basis vectors). Then, $P = Z_{j,1:K}$ belongs to the inside if the angle between $\vec{u}$ and $\overrightarrow{OP}$ is smaller than $\alpha/2$. This involves computing the cosine of $(\vec{u} \cdot \overrightarrow{OP})/|\overrightarrow{OP}|$ and comparing it to $\cos(\alpha/2)$: $\mathcal{O}(K)$ operations for the normalized scalar product, a cosine computation (constant in K and n) and a comparison.

case running time. This involves counting linear extensions on sub-orders recursively, but each step of the recursion returns the count without delay if the sub-order is of a form known to admit a closed-form formula for its count. Chapter 5 will present an optimisation specific to partial orders from the (K, ρ, n)-dimensional order model.

1.3.5.1 General counting algorithm

Algorithm 3 is a recursion for an exact linear extension count of an arbitrary partial order P. It is illustrated on Figure 1.7. It is closely related to a complete linear extension enumeration and the total number of recursive calls is $n|\mathcal{L}(P)|$. Each step of the recursion involves computing the maximal set of the current order $\mathcal{M}(P)$ (the set of elements a of P such that there is there no b with $a \prec b$), then by recursion counting on the suborder $P[-l]$ with l removed for each $l \in \mathcal{M}(P)$.

Algorithm 3 (Knuth and Szwarcfiter [1974]) Linear extension count

```
procedure COUNT(P)
    if P has only one element then
        Return 1
    end if
    Compute the set of maximal elements M(P)
    Return Σ_{l∈M(P)} COUNT(P[−l])          ▷ P[−l]: suborder without l
end procedure
```

In the worst case $|\mathcal{L}(P)| = n!$ (for the null order) and the running time is exponential in n.[4] In practice there may be an improvement if the algorithms aim for a count without enumerating linear extensions, for example by using the following shortcut formulae. To get an idea of computational time, when $n \approx 20$ then counting with Algorithm3 could take at least a few seconds in C/C++ on a basic desktop but orders of magnitude less when avoiding the full enumeration. This measurement comes from practice and agrees with the numerical results of Peczarski [2002] and Li et al. [2005].

[4] Inoue and Minato [2014] give a modification of the recursion that avoids recomputing counts for redundant sub-graphs, reducing the $n!$ term to 2^n or to a polynomial if the width is bounded. Brightwell and Winkler [1991] show that counting linear extensions is #P-complete. Knuth and Szwarcfiter [1974] and Varol and Rotem [1981] outlined an algorithm to generate the set of linear extensions $\mathcal{L}(P)$. Linear extensions were also referred to as "topological sorts". Since then, algorithms have been proposed to reduce the complexity down to $\mathcal{O}(|\mathcal{L}(P)|)$, for example Pruesse and Ruskey [1994] generate all linear extensions in a sequence, where consecutive ones differ by at most a few adjacent transpositions.

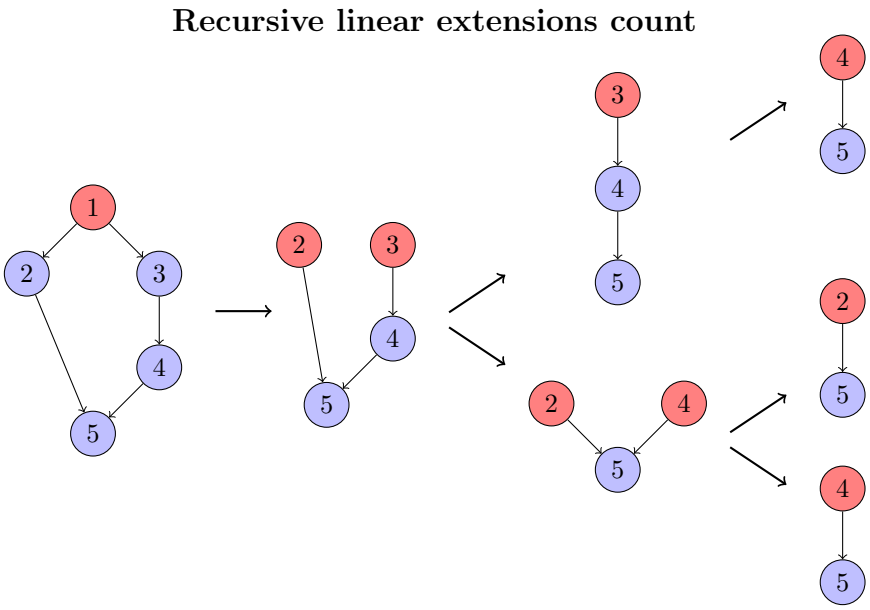

Figure 1.7: Linear extension count recursion Algorithm 3 with successive sub-orders (left to right) after removing maximal elements $\mathcal{M}(P)$ (red). $|\mathcal{L}(P)| = 3$ because three instances of the procedure reach a stop.

Shortcut formulae To reduce running time, the recursion Algorithm 3 could return a count before reaching the last step when P is a single element, if the input P has a recognisable form with a known count formula. These recognisable partial orders are listed here.

A *series partial order* $P = P_1 \otimes P_2$ from partial orders P_1 and P_2 is defined as the orders and disjoint union of elements in P_1 and P_2 and in addition orders $x \prec y$ if $x \in P_1$ and $y \in P_2$. [5]

A *parallel partial order* $P = P_1 \oplus P_2$ from partial orders P_1 and P_2 is defined as the orders and disjoint union of elements in P_1 and P_2 with incomparability between elements of P_1 and P_2.

A *series-parallel partial order* is a partial order obtained from applying series $\otimes$ and $\oplus$ parallel operations in sequence to combine partial orders.

Counting linear extensions for series and parallel partial orders is simplified due to the following from Wells [1971]:

$$|\mathcal{L}(P_1 \otimes P_2)| = |\mathcal{L}(P_1)||\mathcal{L}(P_2)|, \tag{1.3}$$

$$|\mathcal{L}(P_1 \oplus P_2)| = |\mathcal{L}(P_1)||\mathcal{L}(P_2)| \binom{|P_1| + |P_2|}{|P_1|} \tag{1.4}$$

[5] The terminology differs from Brightwell [1993], where a series is called a "linear sum" with operator "+".

and

$$|\mathcal{L}(P)| = \sum_{(Q,R)\in P_d} |\mathcal{L}(Q)||\mathcal{L}(R)| \tag{1.5}$$

where for an arbitrary element d, P_d is the is the set of partition (Q, R) of P with respect to d such that Q and R must include all elements $d \prec q$ and $r \prec d$ respectively. In particular for a single element e, $|\mathcal{L}(P \otimes \{e\})| = |\mathcal{L}(P)|$ and $|\mathcal{L}(P \oplus \{e\})| = (|P| + 1)|\mathcal{L}(P)|$.

As application, a list of isolated elements (no orders) could be computed at each step of the recursion. The recursion then returns $|\mathcal{L}(P)| = n!$ if the order is null and 1 if it is total. Otherwise, Equation 1.4 is applied and the isolated elements are removed from the subsequent sub-orders. Peczarski [2002] and Li et al. [2005] take a step further with a divide-and-conquer approach by finding connected components of P, using Equation 1.5 for each of the components and then combining the counts with Equation 1.4.

1.3.5.2 Counting algorithm in this thesis

The recursion Algorithm 3 with a basic application of the shortcuts formulae is the algorithm of choice in this thesis for counting linear extension of samples from the (K, ρ, n)-dimensional order model. The shortcuts were found to reduce the average running time to the point where counting linear extensions is not the time limiting factor for the applications considered. Chapter 5 will suggest a way to compute the maximal sets that can be useful when n is large by making use of the latent variables.

The number of recursive calls in Algorithm 3 is $n|\mathcal{L}(P)|$, therefore the complexity is controlled by the number of linear extensions. There are at most $n!$ linear extensions (exactly for the null order). A better bound can be found in terms of the width w: for $P \in \mathcal{P}_n$, $|\mathcal{L}(P)| \leq w^n$. As the correlation parameter ρ of the (K, ρ, n)-dimensional order model increases, the expected number of linear extension and width w decreases. These remarks suggest that counting with Algorithm 3 is on average easier ρ is high, which is the case for partial orders of high depth modelled in this thesis.

1.3.5.3 Approximations and special cases

Exact count for special cases Counting linear extensions is faster for some special cases. Möhring [1989] gives methodology for different classes of partial orders, such as series-parallel orders, N-free orders (an order without an N shape occurring, ie $a \prec b$, $c \prec d$ and $c \prec b$) and orders of bounded width. There is ongoing work on improving algorithms for some of these classes, for example Cooper [2013] on a count in $\mathcal{O}(n^w)$

where w is the width. Inoue and Minato [2014] give a method shown numerically to count in a few when $n = 35$ for sparse edges, which the authors consider to be worst-cases because of their high width.

In the statistics literature, Mannila and Meek [2000] uses series-parallel orders for their tractability and stores a binary construction tree representation available for such orders. This bypasses the need for finding the connected components of P before applying Equation 1.4,1.5 as in Peczarski [2002].

Approximate count It is also of interest to know about approximate counting. If such approximation has certain properties, they may have an application in inference algorithms such as some MCMC extensions that aim to avoid computing an exact but more computational likelihood evaluation.

Brightwell and Winkler [1991] give an approximation $\hat{L}$ of the form

$$\mathbb{P}\left(|\frac{\hat{L}}{|\mathcal{L}(P)|} - 1| > \epsilon\right) < \beta$$

where $\hat{L}$ is the estimate of $|\mathcal{L}(P)|$.[6] The scheme uses the uniform sampler of Karzanov and Khachiyan [1991] based on a Markov chain on $G(P)$, the graph whose vertices are linear extensions of P such that adjacent vertices differ by only an adjacent transposition.

Although the bound on the running time given is a polynomial of a seemingly high order, the authors feel it is more conservative than what practice suggests. This may be useful for MCMC schemes such as the two-stage scheme of Christen and Fox [2005], where a quick approximation of the posterior at a proposed sample may lead to reject the sample if it is not likely to be accepted based on the exact acceptance probability. An alternative is the "penalty method" of Ceperley and Dewing [1999] where a posterior density approximation is used in the acceptance probability but a corrective term is applied. However, some strong assumptions are required of the approximation which are not met here exactly.

1.3.6 Comparison with the graph order model

Model interpretation The $G_{n,p}$-graph order model for orders has a less obvious interpretation as a physical process, because there is a dissociation between the assignment of the edges and their orientations.

[6] The running time is shown to be $\mathcal{O}(n^9 \log^6(n) \log(1/\epsilon)\epsilon^{-2} \log(1/\beta))$. It is a fully polynomial randomized approximation scheme, polynomial in n and in $1/\epsilon$.

Partial order structure If $0 < p < 1$, it is easy to show that $G_{n,p}$ has support on $\mathcal{P}_n$. For fixed p, Alon et al. [1994] shows that the depth grows almost surely linearly in n and the order tends to be that of a sum of partial orders (partial orders ordered in a sequence). An element is a *post* if it is comparable with every element other than itself. Bollobas and Brightwell [1997] show that there almost surely exist posts if and only if $p = p(n)$ is parameterized such that $np^{-1}\exp(-\pi^2/3p) \to \infty$. Together these results show that depth may be controlled with the parameter p, which is of interest, however the sum of partial order structure is restrictive.

Marginal consistency The $G_{n,p}$ graph order model is not marginal consistent as can be seen in the following counter-example: $A = \{a_1, a_2, a_3\}$, $o = \{a_1, a_3\}$, h' is the suborder $a_1 \prec a_3$ and $p(h') = p$. Yet, as seen on Figure 1.8 there are 5 graphs leading to $a_1 \prec a_3$ and the sum of their probabilities is

$$\sum_{h|h[o]=h'} p_A(h) = 3p^2(1-p) + p(1-p)^2 + p^3 > p.$$

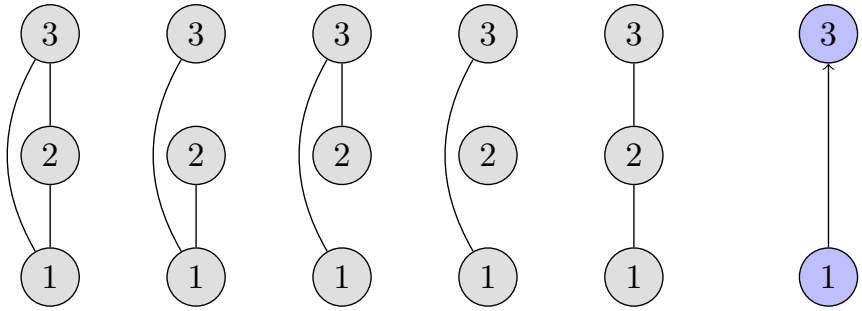

Figure 1.8: The set of 5 graphs on three elements leading to the suborder on two elements on the right.

Complexity of sampling The $G_{n,p}$ graph order model has a sampling complexity in $\mathcal{O}(n^2)$, which may be smaller than that of the (K, n)-dimensional random order model in $\mathcal{O}(Kn^2)$ depending on $K(n)$. This assumes the following sampling procedure: a graph G on n nodes has edges occurring independently with Bernoulli probability p, and then edges $a_i \prec a_j, i < j$ implied by transitivity are added. The cost is the simulation of $n(n-1)/2$ uniform $\mathcal{U}[0,1]$ random variables (then converted to a Bernoulli samples), then $n(n-1)/2$ comparisons to get M from G. Applying a random permutation on the vertices is an additional $\mathcal{O}(n)$.

Counting linear extensions The $G_{n,p}$ orders tend to have structure for which the shortcut formulae apply, decreasing the average running time of the recursion Algorithm3. Since a sample P from $G_{n,p}$ for fixed p and $n \to \infty$ has almost surely infinitely many posts, it can be written as a series $P = P_1 \otimes P_2 \otimes \ldots$. From Equation 1.3, the linear extension count is then a product over counts on much smaller partial orders, $|\mathcal{L}(P)| = \prod_{i \geq 1} |\mathcal{L}(P_i)|$. It is therefore easy to imagine a counting algorithm which computes the lists of posts, applies the counting recursion on each inter-post sub-order and then returns the products of these counts. Its complexity grows linearly in the number of posts and this suggests a much smaller running time than when counting samples from the (K, ρ, n)-dimensional order model.[7]

1.4 Background on order estimation

This section is a literature review on applications of random order models and in particular on estimating orders given a set of observed total orders or rankings. In the literature, the estimated orders are usually assumed to be total not partial. When partial orders are used as models, there is no well developed Bayesian treatment. As will be shown on a toy example, a partial order model may be more attractive in some situations. This thesis will give a general Bayesian solution with partial orders, focussing on the case of dynamic orders.

1.4.1 Total order models

Total order models are popular in ranking problems. For example, observed rankings on some elements a_i may be modelled as random samples from an unobserved full order $a_{i_1} \prec a_{i_2} \prec \ldots \prec a_{i_n}$, to be estimated.

A total order can be thought as a degenerate partial order. Enforcing orders to be total with the (K, ρ, n)-dimensional order model would be possible by choosing the number of latent variable features as $K = 1$. However, for any fixed $K > 1$ and $\rho < 1$ the model still ensures support on total orders: with some non-zero probability a sample has a dimension of only 1. Modelling or performing inference with a partial order model is likely to attract the difficulties encountered in one dimensional model studies. It is therefore interesting to review these here as a prelude to this thesis's work on general partial orders.

[7] Indeed, it is tempting to apply the Central Limit theorem of Brightwell [1993] to the complexity itself in order to establish linear growth as $n \to \infty$ for a sample $P \sim G_{n,p}$ with posts as random variables, but it is not clear whether the CLT conditions apply here.

1.4.1.1 Latent variable models

In the Bradley and Terry [1952] model for pairwise comparisons, elements $\{a_1, \ldots, a_n\}$ have respective weights $\{\lambda_1, \ldots, \lambda_n\}$ such that $p(a_j \prec a_i) = \lambda_i/(\lambda_i + \lambda_j)$. This is generalized by the Plackett-Luce model of Luce [1959] and Plackett [1975] where

$$p(a_n \prec \ldots \prec a_1) = \prod_{i=1}^{n-1} \lambda_i / (\sum_{j=i}^{n} \lambda_j).$$

Caron and Doucet [2010] perform Bayesian inference on λ. These rely on the following observation: if $Y_i \sim \mathcal{E}(\lambda_i)$ and $Y_j \sim \mathcal{E}(\lambda_j)$ are two exponentially distributed random variables, then $p(Y_i < Y_j) = \lambda_i/(\lambda_i + \lambda_j)$. These latent variables are the basis for data augmentation algorithms to perform numerical inference on λ.

Caron and Doucet [2010] also studies the undirected graph version of the Bradley-Terry model. In the undirected graph model, an edge between a_i and a_j is inserted with probability $\lambda_i\lambda_j/(1 + \lambda_i\lambda_j)$. A question is whether this can be extended to a partial model. It is easy to imagine a variant of the $G_{n,p}$-graph order model, where the edge probabilities now depend on each element pair rather than just p. However, as a model this would suffer from the same drawbacks.

More recent works features dynamic Plackett-Luce models. Caron and Teh [2012] extend the Plackett-Luce model for when the number of elements a_i is infinite and observations are rankings on finite sets. In the dynamic version, the measure on the rankings sets and weights follow a Markov process. Baker and Mchale [2014] extend the Plackett-Luce model to include time-varying weights $\alpha(t)$ defined as interpolations between weights $\lambda(k)$ at several time nodes. Maximum likelihood estimation of $\lambda(k)$ is performed with Newton-Raphson.

Comparison to this thesis The $Z \in \mathcal{M}_{n,K}(\mathbb{R})$ latent variables of the (K, ρ, n)-dimensional order model are reminiscent of the weights λ of the Plackett-Luce model and the latent variables Y_i.[8] This thesis will also treat the dynamic version of the problem where features change over time by introducing a Markov process on Z_t. However, the Plackett-Luce model is conceptually different because it does not accommodate a general partial order structure. In some situations, a partial order model is more suitable. As a toy example, Figure 1.9 shows three total orders on the left. Minimizing the log-likelihood for the Plackett-Luce model assuming these

[8] In the early account of a latent variables model in Thurstone [1927] for pairwise comparisons, an order $a_i \prec a_j$ is set if $X_i < X_j$ where X_i and X_j are two univariate normal random variables. The partial order model is a multivariate generalisation.

are independent observations with constraints $\sum_{i=1}^{5} \lambda_i = 1$ and $0.001 \leq \lambda_i \leq 1$ for $i = 1, \ldots, 5$ yields an MLE $\hat{\lambda}_{1:5} \approx (0.90, 0.02, 0.06, 0.01, 0.001)$.[9] This predicts a probability of approximately 0.16 for the order $a_5 \prec a_4 \prec a_3 \prec a_2 \prec a_1$ and 0.19 for $a_4 \prec a_3 \prec a_2$ (when considering orders on these three elements only). This is larger than the 1/3 probability that one would expect from inspecting the observations. As an alternative, consider observations sampled uniformly at random from the set of linear extensions of a partial order. The MLE order is the partial order shown Figure 1.9 and predicts a probability of 1/3 for all three observations.

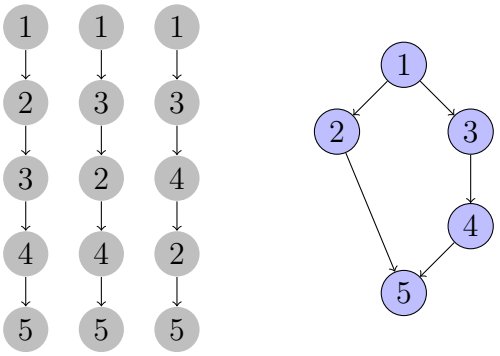

Figure 1.9: Three observations and a plausible partial ordering.

1.4.1.2 Permutation models

The Plackett-Luce model is based on some latent weights λ, which determine a distribution on total orders. This section turns to modelling total orders directly, without latent variables. After all if the end goal is to estimate orders, then latent variables appear like a mere intermediary. However, the following studies will show that some drawbacks exist. For example, it appears difficult to find a tractable continuous time model on total orders. This would also apply to partial orders because they include total orders.

A total order can be seen as a permutation and indeed this section will refer to permutations to reflect the terminology and concepts more commonly used in the literature in this context.

Permutation definitions Some of the work cited involves standard permutation concepts. The set S_n denotes all permutations on $\{1, \ldots, n\}$. A *cycle* decomposition is illustrated by the following example in S_3: π is the permutation $1 \mapsto 1$, $2 \mapsto 3$,

[9] Numerical optimization was performed in Python with the SciPy package of Jones et al. [2001–] using sequential least squares.

$3 \mapsto 2$. Its decomposition $\pi = (1)(2,3)$ has two cycles of length 1 and 2 respectively. A *partition* of $p \in S_n$ is a vector $(b_1, \ldots, b_n)$ where $\sum_{i=1}^n ib_i = n$ and b_i count the number of cycles in p with i elements. In the previous example, the partition is $(1,1)$. Permutations p and q are *conjugate* if there exists $s \in S_n$ such that $p = sqs^{-1}$ (s^{-1} is the inverse of s, in the previous example $\pi^{-1}(3) = 2$). A general result on S_n is that they are conjugate (belong to the same *equivalence class*) if and only if they have the same partition.

Permutation distance models For a metric d on S_n and some location $p_0 \in S_n$ and scale $\lambda \geq 0$ parameters, a Mallows distribution Mallows [1957] for $p \in S_n$ is

$$\mathbb{P}(p|\lambda, p_0) = K(\lambda) \exp\{-\lambda d(p, p_0)\} \tag{1.6}$$

where $K(\lambda)$ is a normalizing constant independent of p_o if d is *right-invariant*, ie satisfies $d(p, p_0) = d(pq, p_0 q)$ for all p, p_0, q in S_n. Right-invariance implies invariance to relabelling of elements. A popular metric is "Kendall's τ" equal to the number of pairs (i, j) such that $p_0(i) < p_0(j)$ and $p(i) > p(j)$. For such metric, $K(\lambda)$ can be computed exactly while avoiding computing a sum of $n!$ terms over $p \in S_n$.

For other metrics d, $K(\lambda)$ may need to be estimated. Sørensen et al. [2014] gives an importance sampling approximation practical for medium n avoiding a sum over $n!$ permutations. Values of $K(\lambda)$ are pre-estimated and then Bayesian inference on λ and p_0 is performed with MCMC with as prior a uniform on $p \in S_n$ and an exponential prior on the scale parameter λ. In the time-varying version, the transition from p_{t-1} to p_t is a Mallows distribution $\mathbb{P}(p_t|\lambda, p_{t-1})$ as in Equation 1.6, with p_{t-1} as location and λ as scale parameter. The observations are permutations R_t. The observation model is also a Mallows distribution $\mathbb{P}(R_t|\alpha_t, p_t)$, where d is the same metric and the scale α_t follows a normal random walk with scale σ. Inference on (p, R, α, σ) is performed with MCMC. The authors note that a numerical difficulty is searching through the large set S_n of permutations on n elements in MCMC, rather than estimating the normalizing constant $K(\lambda)$.

Gupta and Damien [2002] suggests a new distribution so that permutations within the same conjugacy class are close to each other. The distance between the conjugacy classes of p and q is defined as their Hausdorff distance.[10] A prior on $p \in S_n$ is then specified as a Mallows distribution based on the distance between the conjugacy

[10]The *Hausdorff distance* between two subsets X and Y of a space with metric d is defined as $d_H(X, Y) = \max(\sup_{x \in X} \inf_{y \in Y} d(x, y), \sup_{y \in Y} \inf_{x \in X} d(x, y)$. It is the maximum distance between a point in one set and the other set.

classes of p and that of a hyper-parameter $q \in S_n$, and this is extended to partially ranked data.

Permutation processes There seems to be a limited number of studies of continuous time models on S_n with fixed n.[11] Berestycki and Durrett [2003] studies a continuous time walk on S_n where at times of a rate 1 Poisson process, two elements sampled uniformly at random are transposed. If the two elements belong to two different cycles then the cycles merge, whereas if they belong to the same cycle then the cycle splits. Asymptotics are given for the minimum number of transpositions D_t^n (*Cayley distance*) needed at time t to return to the identity at $t = 0$, for example $\mathbb{E}\{D^n_{cn/2}\} \sim cn/2$ with $n \to \infty$ and $0 < c < 1$.

An alternative is a transposition affecting only adjacent elements, where at t the values $p_t^n(i)$ and $p_t^n(i+1)$ are exchanged. The minimum number of adjacent transpositions $d_{adj}(p)$ to get $p \in S_n$ (the shortest path in the random walk) is the Kendall's τ distance. When p_t^n is the random walk where adjacent transpositions occur with a rate 1 Poisson process and p_k^n is its discrete time chain, formulae for the expected distance $\mathbb{E}\{d_{adj}(p_k^n)\}$ and asymptotics for $\frac{1}{n}d_{adj}(p_{nt}^n)$ as $n \to \infty$ are given in Berestycki and Durrett [2007].

As an aside, the reader may have noticed a difference between the distance and process based permutation models. The Mallows distance based model is tractable (for some metrics d), but simulating it may require an algorithm such as MCMC. In contract, the transposition process on permutations appears less arbitrary as a model and can be simulated easily, but its distribution density may be less tractable.

Comparison to this thesis The previous results on permutation processes with random transpositions suggest that it is difficult to compute the distribution of $d(p_{t_1}, p_{t_0})$ (rather than merely the expected distance) for two arbitrary times t_0 and t_1 in $\mathbb{R}$. The latent variables models circumvent this difficult. In the latent variables Z-process model of Section 2.2, the observations are permutations (total orders) and states are latent variables matrices Z_t inducing the partial order $P(Z_t)$. The state is not restricted to a permutation. Furthermore, the Z-process is continuous in time

[11] There exists some standard partition and population models, but they describe the process of a permutation growing with n whereas this thesis focusses on fixed n. A Yule process is a Markov process on permutations where the number of elements increases over time (a jump Markov chain on $\cup_{n\geq 0} S_n$), where a new element n enters the permutation either as a new single element cycle or as a member of an existing cycle.

and can be discretized at arbitrary observation times. The inference scheme developed is exact (asymptotically) and does not pre-compute some approximations of a normalizing constant.

1.4.2 Partial order models

This section turns to the literature where partial orders are used in modelling, for example to represent the structure of a network. Some partial order models can be found but are often restricted to a partial order subset for computational reasons, for example series-parallel orders in Mannila and Meek [2000]. A partial order maximum likelihood estimate is studied in Beerenwinkel et al. [2007] and in Mannila and Meek [2000]. The heuristics of Ukkonen et al. [2005] implicitly control the partial order depth. Work such as Fernandez et al. [2009] involves inferring several partial orders from a large set of total orders, for compression or visualization purposes, but without a statistical model.

In contrast, this thesis gives a more developed Bayesian methodology for partial orders. It is for a general partial order model and is attractive for applications with its control over partial order depth and time series structure. This comes at a computational cost, therefore several chapters are dedicated to numerical inference.

Conjunctive Bayesian Networks Beerenwinkel et al. [2007] studies Conjunctive Bayesian Networks (CBN). A CBN (A, θ) is a partial order on A where each element or *event* a_i has a parameter θ_i for the probability that a_i occurs given that its predecessor events have occurred. An *order ideal* is a subset $g \subseteq A$ such that if $a_i \in g$, $a_j \prec a_i \implies a_j \in g$. In applications, g is an observation and is interpreted as a *genotype* indicating whether each event occurred or not. Data is the number of observations of each genotype g. The likelihood is

$$p(g|\theta) = \prod_{e \in g} \theta_e \prod_{e \in \min(g^c)} (1 - \theta_e)$$

where $\min(g^c)$ is the set of events which could happen after those in g occur (in DAG terms, the set of outgoing vertices adjacent to the maximal set of vertices of g).

For a fixed partial order, it is shown that the MLE $\widehat{\theta_e}$ has a closed-form expression in terms of the number of observations of each g and of the partial order. Furthermore, the MLE partial order maximising $\widehat{\theta_e}$ is the unique *largest* partial order compatible with the observations. For a partial order, P is larger than Q means $a \prec_P b \implies a \prec_Q b$. The MLE is such that $a_i \prec a_j$ if there is no genotype g with only a_j occurring

without a_i, ie all g are such that $g \cap \{a_i, a_j\} \neq \{a_j\}$. In some situations, this CBN MLE is different than the MLE for the models that will be discussed in this thesis. This is due to the particularity of a CBN model where events are constrained in the order of their occurrence.

Network graph prior Froehlich et al. [2007] describes inference on a directed graph Φ representing a biological *network*. Some of the methods presented do not restrain Φ to a partial order DAG (edges of a partial order DAG are defined up to transitive closure). Nevertheless, the following could apply to a partial order DAG.

A prior assumes that edges are sampled independently as $p(\Phi) = \prod_{i,j} \Phi_{i,j}$, where $\Phi \in \{0, 1\}$ is positive if $a_i \prec a_j$. A likelihood describes the gene effects observations given network parameters include Φ. A score function for Φ involves the prior and the likelihood with AIC. Stochastic Annealing is used to search for a DAG to (approximately) maximize the score function. For this purpose, one-step ahead transition distributions are suggested: $add(G)$ adds an edge to the transitive closure of G and $del(G)$ deletes one from the transitive reduction of G such that $add \circ del = del \circ add = \mathbb{1}$. This is to guarantee that G can reach any other graph, first by deleting all edges then adding edges.

Fragment orders for seriation A paleontological data matrix D with columns corresponds to taxa and rows to sites, with $D_{i,j} = 1$ if taxon j has been found at site i and $D_{i,j} = 0$ otherwise. The aim is to find a permutation of the rows such that the 1 elements in each columns are consecutive, or least such that the number of 0 gaps is minimized. There may be several permutations that achieve the same total number of 0 gaps.

Rather than minimising some criteria to find a single best permutation, Ukkonen et al. [2005] suggests finding a partial order P between the rows themselves. A cost function $L(D|P)$ defines the total number of gaps. Because $L(D|P) = 0$ for the null partial order which is not considered interesting, the objective should be

$$\min_{P \in S} L(M|P) + \alpha(P)$$

where $\alpha(P)$ is large as a penalty if P has few orders, and $S \subseteq \mathcal{P}_n$ a set of partial orders. Without expliciting α nor S, the following heuristics is used: sample a random subset of k rows $M' \subseteq M$, select some of the $k!$ total orders T such that $L(M'|T) \leq \mu$ but discard M' if there are many candidates T (non-information likelihood). The

authors note that a problem is how to choose μ based on the data, a poor choice affects numerical results.

Given the set of sampled T, a partial order is formed where an edge $x \prec y$ is added sequentially if its appearance frequency exceeds that of $x \prec y$ by a threshold and does not lead to a cycle. This is slightly different from the MLE in Section 2.1.3 where the order $x \prec y$ is added if it occurs in some T and $x \prec y$ never does.

Series-Parallel orders In earlier work, to describe a given set of sequences, Mannila and Meek [2000] suggests as model a single partial order of the series-parallel orders set. The binary construction tree representation of series-parallel partial orders is used to facilitate counting linear extensions[12] and searching through partial order with a transition distribution. The likelihood is a 2-mixture component, each a uniform over linear extensions, with one component as the null partial order guaranteeing a non-zero likelihood. Starting from the null partial order, a greedy algorithm finds an optimal order by applying a fixed number of transitions, determining the mixture weights with the EM algorithm and selecting one maximizing the likelihood for the subsequent iteration.

Amer et al. [1994] also uses a series-parallel order as model. It is for application to multimedia transmission flows, where an order between two items is a temporal constraint where one must be transmitted before the other. The number of linear extensions of a partial order is used a metric of its complexity, in this application a higher number is associated with more possible orderings and therefore less memory demand.

Bucket orders A *bucket order* $P = (A, \prec)$ is a total order with ties, a simple special case of series-parallel when there exists a partition of $A_1, \ldots, A_m$ of A with $x \prec_P y$ if and only if $x \in A_i$ and $y \in A_j$ for some $i < j$ and $x \parallel y$ otherwise if $i = j$. Given some observed comparisons, a *pair order matrix* $T_{i,j}$ counts events $a_i \prec a_j$ up to a factor such that $T_{i,j} + T_{j,i} = 1$. A bucket order corresponds to $T^B_{i,j}$ with value 1 if $a_i \in a_j$, 1/2 if $a_i \parallel a_j$ and 0 otherwise.

Gionis et al. [2006] gives an approximation T^B of the closest bucket order pair order matrix to T in L_1 norm with expected number of comparisons $\mathcal{O}(n \log n)$. Obtaining the pair order matrix from paleontology data is done with MCMC as in Puolamäki et al. [2006], on a state space that includes total orders and parameters of an error

[12] For a given binary construction tree representation, equations in Section 1.3.5 allow to count linear extensions fast.

model (partial orders are not part of the MCMC state) such that the likelihood given error parameters and *any* total order is never exactly zero, unless this is enforced by the prior.

Cover with hammock orders The model of Fernandez et al. [2009] is a deterministic counterpart of Mannila and Meek [2000] and of some models developed in this thesis. The aim is to reconstruct a partial order P given some total order y. An algorithm is given for the problem of computing P such that $y \subseteq \mathcal{L}(P)$ and $|\mathcal{L}(P)|$ is minimum. It reduces to computing the intersection of P, and is the MLE of a model discussed in the next chapter. In general there does not exist a partial order P such that $y = \mathcal{L}(P)$. The authors therefore aim to find a minimum number of partial orders P_k as *cover*, ie such that $y = \cup_k \mathcal{L}(P_k)$. Finding a cover is computationally challenging because $\mathcal{P}_n$ is a large set.

The author therefore consider the problem of finding a cover for the class of *hammock* partial orders. A hammock partial order is special case of a bucket order where there is no consecutive sets of ties: the layered partition $A_1, \ldots, A_m$ of A is such that either $|A_i| = 1$ or $|A_{i+1}| = 1$. A *kite-k* is a hammock where there is only one A_i such that $|A_i| > 1$ and its size is $|A_i| = k$. The authors give a polynomial algorithm (in the number of total orders in y and in the number of elements $n = |A|$) for finding a cover of kite-2 partial orders, but show that for another form of hammock the problem is NP-complete.

Casas-Garriga [2005] attempts to reconstruct some partial orders given a set of sequences (total orders with repeat of a same element allowed). The numerical results show partial orders not unlike the hammock partial orders. The reconstructed partial orders have high depth, a benefit of allowing for multiple partial orders to explain the observations.

Chapter 2

Latent variables process

In Chapter 1, the (K, ρ, n)-dimensional order was introduced as a latent variable static model. In this chapter, the static latent variable model is extended to a partial order process $P_t = P(Z_t)$ evolving in time $t \in \mathbb{R}$. At discrete times t_i, observations y_i are emitted as linear extensions from the partial order $P(Z_{t_i})$. This fits the framework of a Hidden Markov model, with the latent variables Z_{t_i} as part of the hidden state X_i.

This chapter starts with the description of the observation model, i.e. the distribution of the linear extensions y_i given the partial orders $P(Z_{t_i})$. The main model of interest is the uniform model where y_i is sampled uniformly from the set of linear extensions of $P(Z_{t_i})$. For this uniform model, the maximum likelihood partial order estimate is characterized. Then the dynamic model on $P(Z_t)$ is described. The model retains some of the attractive properties of the (K, ρ, n)-dimensional order as a model. It is tractable and can be discretized at the sampling times t_i. Finally, the full posterior distribution is stated. Performing Bayesian inference on the partial orders and Hidden Markov model static parameters by sampling for such posterior will be the topic of Chapter 6.

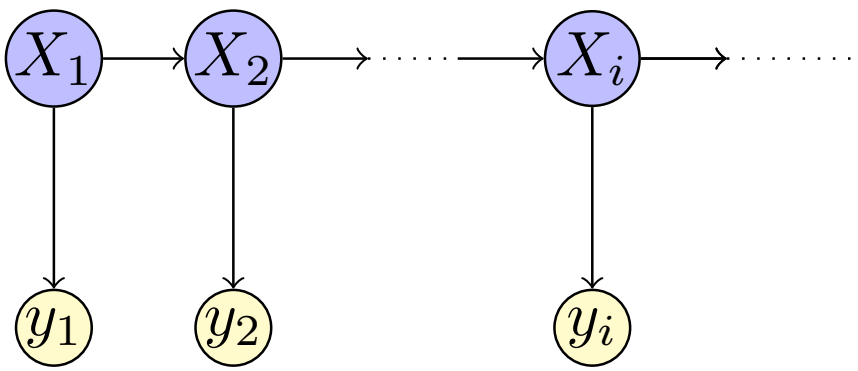

Figure 2.1: A generic Hidden Markov model where the observations y_i are independent given the hidden states X_i. States X_i follow a Markov chain.

2.1 Observation model

This section describes the observation model for the HMM. The model used in the implementation is the observation without errors described first. In the static case, the Maximum Likelihood estimate for this model is characterized. For completeness, an observation model with errors is also included.

2.1.1 Observation without errors

Mannila and Meek [2000] study a model for series-parallel partial orders without a time series structure, where the observations are linear extensions sampled uniformly from these orders. The observation model used could apply to any partial order and is therefore of interest here. This section describes such observation model with a particular distribution for missing elements in observations and in a setting where more than one partial order emits the observations, precisely there is one partial order for each observation due to the HMM time structure.

Observations $(y_i)_{i=1}^T$ at respective times $(t_i)_{i=1}^T$ are total orders on random subsets $(o_i)_{i=1}^T$ of a fixed set $\{a_1, \ldots, a_n\}$. y_i is sampled uniformly at random from the set of linear extensions of a partial order $P_{t_i}[o_i]$, the restriction of the partial order P_{t_i} to its suborder on o_i:

$$y_i \sim \mathcal{U}[\mathcal{L}(P_{t_i}[o_i])]$$

where $\mathcal{L}(P_{t_i}[o_i])$ denotes the set of linear extensions of partial order $P_{t_i}[o_i]$. The conditional likelihood is

$$p(y_i|P_{t_i}[o_i]) = 1/|\mathcal{L}(P_{t_i}[o_i])| \tag{2.1}$$

if y_i respects $P_{t_i}[o_i]$ and 0 otherwise. In other word, y_i is a linear extension of P_{t_i}. In loose terms, y_i can be viewed as some observed extract from the partial order as it could correspond to a path in its graph (any path would be a valid observation y_i, but y_i could be a total order that is not a path).[1]

This assumes absence of observation errors in the sense that order y_i needs to respect P_i, the conditional likelihood is 0 even if it is due to a single conflict. For this reason, y_i will be referred to as *hard obstacles*. Algorithms for counting $\mathcal{L}(P_{t_i}[o_i])$ were described in Section 1.3.5.

[1] As a remark, the model generalizes to when y is any general partial order that extends P_{t_i}, not just a total order. An order y extends P if its orders do not conflict with those in P.

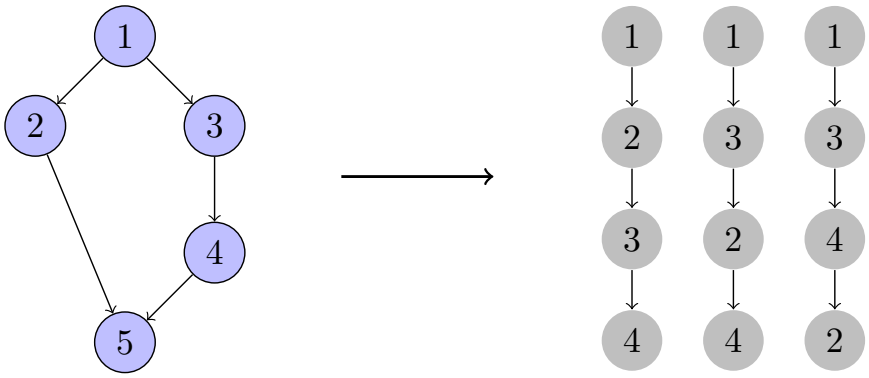

Figure 2.2: The partial order P on $n = 5$ elements and suborders on $\{1, 2, 3, 4\}$ as linear extensions of P.

2.1.2 Observation with errors

The uniform model where a linear extension is sampled from a partial order is an observation model without measurement error in the sense that the total order observed needs to respect all orders of the partial order. The following observation models enlarge the support to include total orders which do not respect the partial order.

It is tempting to smooth the likelihood by assigning a small value if a total order does not extend the partial order. A mixture model would achieve this. Mannila and Meek [2000] uses a two component mixture model with one component taken as the null partial order:

$$y|P \sim \delta p(\cdot|P) + (1 - \delta)p(\cdot|P_0)$$

where $0 < \delta < 1$ is the mixture weight, P is a partial order $P \in \mathcal{P}_n$, P_0 is the null order and $p(\cdot|P)$ is the uniform distribution on linear extensions of P. Since $p(y|P_0) = 1/n_y!$ for all y, where n_y is the length of y, the likelihood is positive for all P. It is computationally simple (for given mixture weights), but arbitrary as a model and does not discriminate between total orders y that have different number of order conflicts with P.

It would be preferable for a model to assign a smaller values as the number of order conflicts increases. This may come at the cost of computational complexity. The queue jump model and its latent variable version describe below achieve this and also have a physical process interpretation.

The concept of measurement error could be confused with that of partial order depth. As the depth of a partial order decreases, the number of orders might decrease

and therefore more linear orders might be observed. In this thesis, depth is a variable of the partial order process and is separated from the observation model.

Previous work Nicholls and Muir Watt [2011] suggests an observation where elements are allowed to jump the queue. Given a partial order P on A, a linear extension follows a process where with probability p an element i is chosen at random from A and with probability $1-p$, the top element of a random linear extension of P is chosen. The second element sampled similarly with P restricted to the remaining $A \setminus i$ elements. The likelihood is given by

$$p(y|P,p) = \prod_{j=1}^{m-1} \left(\frac{p}{m-j+1} + (1-p)\frac{|\mathcal{L}(P(y_{j+1:m})|}{|\mathcal{L}(P(y_{j:m})|} \right),$$

it is positive for $p > 0$ and the uniform model is recovered with $p = 0$.

A shortcoming of this model is the fact that nodes are more likely to receive promotion than demotion, although this can be partially addressed with a mixture with one distribution allowing promotion only and the other demotion only.

Latent variables variant It is possible to formulate a latent variable version of the error model, where y is observed given latent variables Z (rather than $P(Z)$), where elements are more likely to jump the queue if they features scores in Z are higher (rather than elements jumping uniformly at random).

For a given Z, the orders of $P(Z)$ are determined by $a_i \prec a_j$ if and only if $Z_{i,1:K} < Z_{j,1:K}$. In particular, the set of top elements $T(P)$, where $P(Z)$ and Z is given, is the set of elements i such that there is no j such that $Z_{i,1:K} < Z_{j,1:K}$. In the following observation model, element may only jump the queue if they are approximately in the top list: proceeding like in the queue jump model, the subsequent element either in T or among elements j such that $Z_{j,1:K} \approx Z_{T,1:K}$ according to some given metric. In this observation model the queue jumping process makes use of the information in the K feature scores, as opposed to the previous uniformly at random promotion distribution.

2.1.3 Maximum likelihood estimate

In the following, the $\widehat{P}$ is sought for the following likelihood $L(y|P)$: each observation y_k in y is independent and from the uniform on the linear extensions $\mathcal{L}(P[o_k])$, therefore

$$L(y|P) = \prod_{i=1}^{T} \frac{\mathbb{1}\{P[o_i] \uparrow y_i\}}{|\mathcal{L}(P[o_i])|}. \tag{2.2}$$

Observations of full length In the case where the observations y are total orders of "full length" and without missing elements, ie they are on A ($o_i = A, \forall i$), the MLE is easy to determine. Among partial orders that allow y as linear extensions, the intersection corresponds to the partial order of minimum number of linear extensions and is there the MLE.

Proposition 3 (Intersection is the MLE)**.** *If for all i, $o_i = A$, then the intersection of y is the unique partial order P of least number of linear extensions $|\mathcal{L}(P)|$ such that P is extended by y.*

Proof. By contradiction: denoting by Y the intersection of y and P a partial order minimizing the number of linear extensions such that it is extended by y, if $|\mathcal{L}(P)| < |\mathcal{L}(Y)|$ then there exists an order $a_i \prec a_j$ in P not in Y, therefore at least one y_k in y does not extend $a_i \prec a_j$. □

This result is not unlike Beerenwinkel et al. [2007] p.7 where the partial order MLE is shown to be the "largest" partial order compatible with the observations, but for a different observation model and data. However, the result differs in the case when observations can be suborders on $o_i \subset A$. In particular, in that case the MLE for Equation 2.2 is not the intersection any more.

The posterior is $\pi(P|y) \propto L(y|P)p(P)$ with prior $p(P)$. Hows does the MLE $\widehat{P}$ compare to the MAP? Clearly, the MLE $\widehat{P}$ is equal to the MAP $\widehat{P_{\mathcal{U}}}$ when the prior is the uniform on $\mathcal{P}_n$. When y is a single element, a linear order, then the MLE $\widehat{P}$ is y and the posterior $\pi(P|y) \propto \mathbb{1}\{P \uparrow y\}p(P)$. If y has both an element y_k and its reverse order, then both the MLE and the MAP are both the null order. The MAP cannot have more orders (edges) than the MLE $\widehat{P}$, because any additional edges results in a zero likelihood. However it could have less, for example if the prior is the point mass on the null order, or the (K, ρ, n)-dimensional order when the correlation ρ is small enough.

Observations on suborders The intersection of a set of linear orders y on A is the "largest" or most "informative" order on A extended by y: no order relation can be added without violating $\prec_y$. Figure 2.3 illustrates is a generalization to when some linear orders in y are only on subsets of A, defined as:

Definition 1 (Intersection for suborders of variable length)**.** *An intersection of K linear orders $y_k = (o_k, \prec_{y_k}), o_k \subseteq A, k = 1 : K$ is the unique partial order $P = (A, \prec)$ compatible with y_k and with each edge supported by at least one y_k, i.e. $a_i \prec a_j$ iff*

there exists k *and* $(i,j) \in o_k^2$ *such that* $a_i \prec_{y_k} a_j$ *and there are no* k *and* $(i,j) \in o_k^2$ *such that* $a_j \prec_{y_k} a_i$.

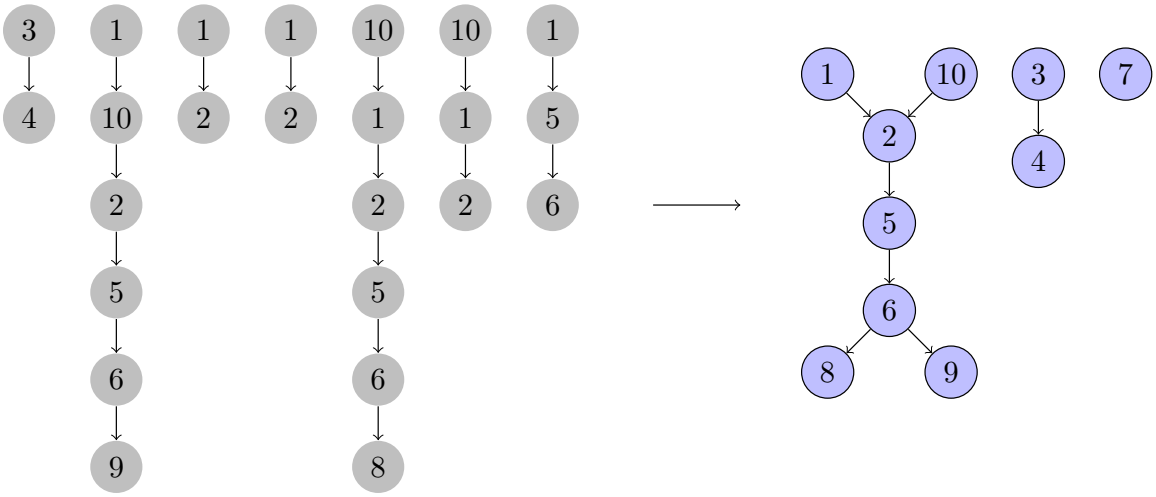

Figure 2.3: Set of suborders (left) and their intersection (right). Orders appearing at least once and without conflict appear in the partial order, for example $4 \prec 3$.

When at least some of the observations are on subsets of A ($\exists i, o_i \subset A$), the intersection of y is defined as the *intersection* in Definition 1. Unfortunately, the intersection is not necessarily the MLE and the MLE may not be unique. For example on Figure 2.3, if P denotes the intersection (right), then inserting the order $1 \prec 4$ to P results in a partial order of higher likelihood.

Without giving an explicit algorithm to determine the MLEs and count them, the following remarks are starting points. It is easy to determine the MLEs if there exists a partition of A, $A = A_1 \cup A_2, \ldots, A_v$, such that each observation y_i is a total order on one of the A_j (ie, there exists j such that $o_i = A_j$). As notation, y^{A_j} is the set of y_i which are total orders on A_j (ie, $o_i = A_j$) and P_j the intersection of y^{A_j} (restricted to A_j). Then in this settings the set of MLEs are the *series* (defined in Section 1.3.5) of the v partial orders P_j, for example $P_1 \otimes \ldots \otimes P_v$. For example, if the partition is of size two, $A = A_1 \cup A_2$, then there are two MLEs: $P = P_1 \otimes P_2$ and $P = P_2 \otimes P_1$. It can be shown by induction that the series $\otimes$ operator is the operation adding edges between disjoint orders P_j into P that minimizes the linear extension count of P.

This can be generalized to when y^{A_j} is the set y_i on subsets only of A_j, ie $o_i \subset A_j$. In this settings, if each y^{A_j} contains at least one observation y_i of full length $|A_j|$, then the previous method to determine MLEs applies by computing P_j as the intersection of y^{A_j} (instead of the standard intersection).

2.2 Partial order process

As reminder, $Z \in \mathcal{M}_{n,K}(\mathbb{R})$ denotes a matrix of n rows and K columns corresponding to elements and features respectively. Z maps to a partial order $P(Z)$ by the rule: $a_i \prec a_j$ if and only if $Z_{i,1:K} < Z_{j,1:K}$. In the (K, ρ, n)-dimensional order model, $Z_{i,1:K} \overset{iid}{\sim} \mathcal{N}_K(0, \Sigma)$ with unit variance and equi-correlation ρ. This is a static model.

In the following, this is extended to a reversible stochastic process in time (Z_t, ρ_t) but with initial distribution at $t = 0$ the static model. It is then discretized at some arbitrary discrete times t_i. This allows for use as transition distributions f_i on the hidden states (Z_{t_i}, ρ_{t_i}) of the Hidden Markov model if t_i are fixed at the times of the observations y_i.

2.2.1 Stochastic process on latent variables

At times $\Phi = (\Phi_1, \Phi_2, \ldots, \Phi_k, \ldots)$ of a Poisson process $\mathcal{P}_t^C$ of rate λ_C a change point occurs where for $k \geq 1$, $(Z_{\Phi_k}, \rho_{\Phi_k})$ is distributed independently of all history. At times Φ_k for $k \geq 1$, the parameter ρ is sampled according to a Beta distribution $\rho_{\Phi_k} \sim \mathcal{B}(\alpha_\rho, \beta_\rho)$ and $\forall i \in \{1, \ldots, n\}$,

$$(Z_{\Phi_k})_{i,1:K} | \rho_{\Phi_k} \sim \mathcal{N}_K(0, \Sigma_{\rho_{\Phi_k}}).$$

At times $\phi = (\phi_1, \phi_2, ...)$ of a Poisson process $\mathcal{P}_t^S$ (the singleton or mutation process) of rate λ_S, the latent variables $(Z_{\Phi_k})_{i,1:K}$ of some element $i \sim \mathcal{U}[1, n]$ are independently renewed at a fixed ρ_{Φ_k}.

Thus, in time intervals between change points $[\Phi_k, \Phi_{k+1}]$, $\rho_t = \rho_{\Phi_k}$ is constant whereas some rows of Z are renewed whenever a singleton event occurs. Figure 2.4 shows the effect of a change point and three singleton events on the latent variables and ρ.

2.2.2 Time discretization

Here, the process on (Z_t, ρ_t) is discretized in time by computing the transition density f between two fixed and arbitrary times $t_i < t_{i+1}$ in $\mathbb{R}$.

In the following notation, $S_{t_{i+1}} = S_{t_{i+1}}(Z_{t_i}, Z_{t_{i+1}}) \subseteq \{1, \ldots, n\}$ is the set of elements whose path has been updated at least once during $[t_i, t_{i+1}]$ and $C_i \in \{0, 1\}$ such that $C_i = 1$ if a change point occurred during $[t_i, t_{i+1}]$, ie there exists a change point time $\Phi_k \in [t_i, t_{i+1}]$, and $C_i = 0$ otherwise.

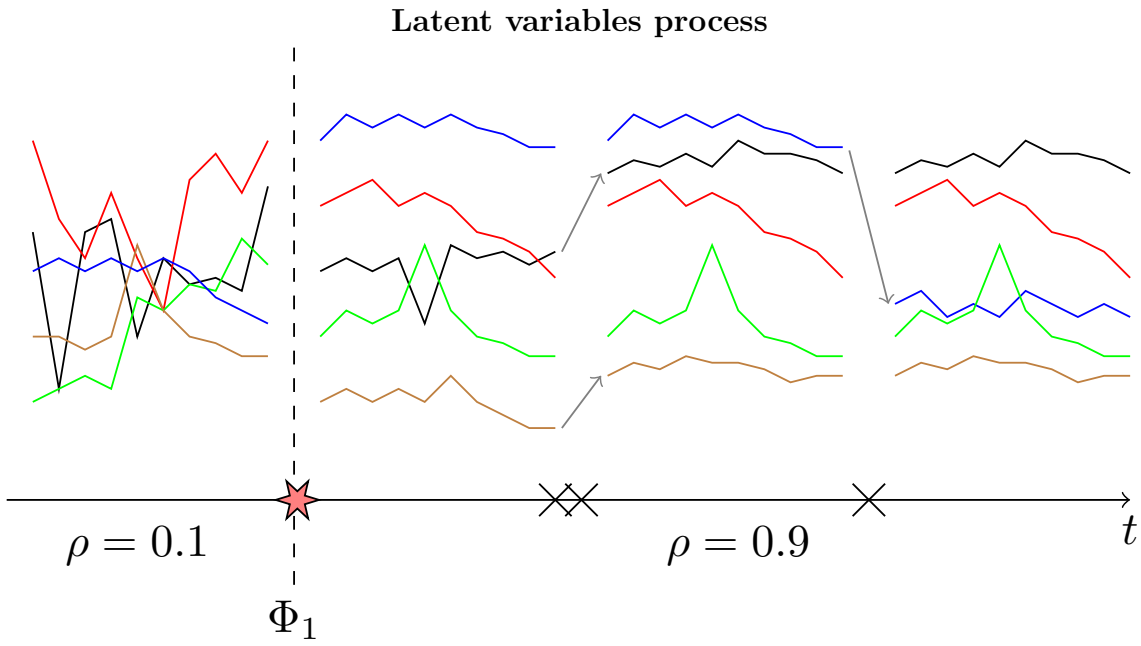

Figure 2.4: The latent variables Z_t are plotted at four time points, one before before a change point Φ_1 (red star) and three after. At each of the four time points, a coloured line represents the features of an element. At the change point, ρ increases from 0.1 to 0.9 and hence the lines look flatter thereafter. At each singleton event (three cross outs), an element's features is either renewed or remains unchanged. Here, three renewals occur in total (each emphasized by a grey arrow pointing to the new values).

The transition distributions when a change point occurs in $[t_i, t_{i+1}]$ are by definition

$$f(\rho_{t_{i+1}}|C_i = 1) = \mathcal{B}(\rho_{t_{i+1}}|\alpha_\rho, \beta_\rho) \tag{2.3}$$

and

$$f(Z_{t_{i+1}}|\rho_{t_{i+1}}, C_i = 1) = \prod_{j=1}^{n} \mathcal{N}_K((Z_{t_{i+1}})_{j,1:K}|0, \Sigma_{\rho_{t_{i+1}}}). \tag{2.4}$$

As initialization of the process, the distribution $\mu(\cdot)$ of the initial sample (Z_{t_0}, ρ_{t_0}) at t_0 is as in equations 2.3 and 2.4. This is the equilibrium of the latent variable process.

When no change point occurred in $[t_i, t_{i+1}]$,

$$f(\rho_{t_{i+1}}|\rho_{t_i}, C_i = 0) = \mathbb{1}\{\rho_{t_{i+1}} = \rho_{t_i}\} \tag{2.5}$$

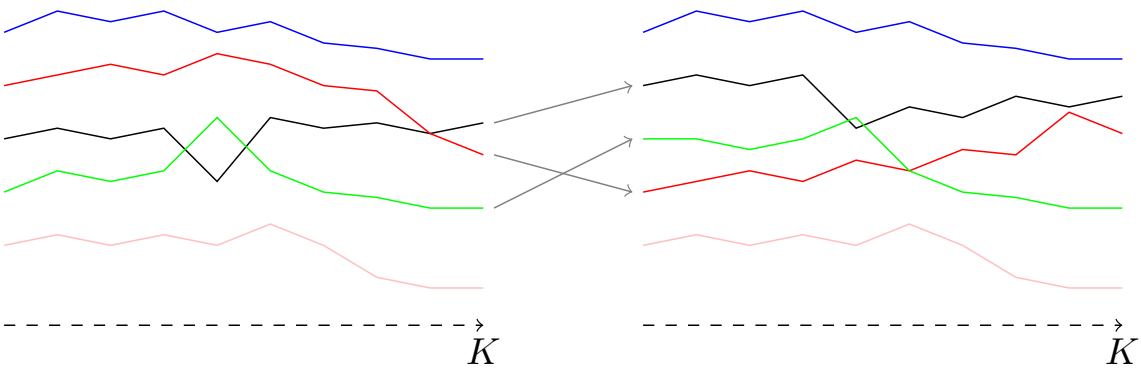

Figure 2.5: Process Z_t with $n = 5$ at two time points between which at least three singleton events occurred (an event may affect the same path multiple times). Both the upper and lower paths remained constant.

and

$$\begin{aligned} & f(Z_{t_{i+1}}|\rho_{t_{i+1}}, Z_{t_i}, C_i = 0, \lambda_S, t_{i+1} - t_i) \\ = \; & f(S_{t_{i+1}}|\lambda_S, t_{i+1} - t_i) \prod_{j \in S_{t_{i+1}}} \mathcal{N}_K(Z_{t_{i+1}})_{j,1:K}|0, \Sigma_{t+1}), \end{aligned} \tag{2.6}$$

where $f(S_{t_{i+1}}|\lambda_S, t_{i+1} - t_i)$ is the probability that the paths of elements $S_{t_{i+1}}$ have been renewed at least once and the product is over normal densities for the renewed paths. Equation 2.6 is a density with respect to the product measure $\prod_j d(Z_{t_{i+1}})_{j,1:K}$ with $j \in S_{t_{i+1}}$. The set $S_{t_{i+1}}$ has a random size and the density is normalized over $\emptyset \cup \mathbb{R}^K \cup \ldots \cup \mathbb{R}^{nK}$. Although of varying dimensions, objects are all distinguishable and therefore the issues of label switching do not arise.

Samples $(Z_{t_{i+1}})_{j,1:K}$ are from a continuous distribution and a renewal of the j-th path occurred almost surely if and only if $(Z_{t_{i+1}})_{j,1:K} \neq (Z_{t_i})_{j,1:K}$. Therefore the set of such j, $S_{t_{i+1}}$, can be found by inspection of Z_{t_i} and $Z_{t_{i+1}}$. Figure 2.5 shows the Z_t process at two times points where the singleton events affected three paths (possibly with replacement). Almost surely no change point occurred because two paths remained equal over time.

By properties of the Poisson processes [2], each path is renewed at least once in $[t_i, t_{i+1}]$ with probability $p_i = 1 - \exp(-\lambda_S(t_{i+1} - t_i)/n)$. With $|S_{t_{i+1}}|$ the number of renewed paths,

$$f(S_{t_{i+1}}|\lambda_S, t_{i+1} - t_i) = p_i^{|S_{t_{i+1}}|}(1 - p_i)^{n - |S_{t_{i+1}}|}. \tag{2.7}$$

[2]The conservation property of the Poisson process allows to split the process into independent Poisson processes. Since $\mathcal{P}_t^S$ affects paths uniformly at random, the split is into n independent Poisson processes of rate λ_S/n.

2.3 Overview for Bayesian inference

This section overviews the model on the parameters of interest

$$\theta := (\Phi, \lambda_C, \lambda_S, (t_i)_{i=1}^T)$$

and

$$X_i := (Z_{t_i}, \rho_{t_i})$$

where $Z_t \in \mathcal{M}_{n,K}(\mathbb{R})$ is a latent variable matrix, $0 < \rho_t < 1$ is features correlation parameter, Φ_k are change point times, λ_C and λ_S are the rate of the change point $\mathcal{P}_t^C$ and singleton $\mathcal{P}_t^S$ Poisson processes respectively and t_i are times of the observations y_i. Parameter Φ is a strictly increasing sequence of (possibly infinite) real values. Some hyper-parameters α_ρ and β_ρ specify the distribution of ρ at change points, $\rho_{\Phi_k} \sim \mathcal{B}(\alpha_\rho, \beta_\rho)$.

Given observations are T total orders y_i sampled independently given $(Z_{t_i})_{i=1}^T$ and the likelihood is the uniform on linear extensions

$$g((y_i)_{i=1}^T | (Z_{t_i})_{i=1}^T, (t_i)_{i=1}^T) = \prod_{i=1}^T \frac{1}{|\mathcal{L}(P_{t_i}[o_i])|}$$

where $P_{t_i} = P_{t_i}(Z_{t_i})$ is the partial order mapped to from latent variables Z_{t_i}. The HMM has hidden state $X_i := X_{t_i} := (Z_i, \rho_i)$, observations $(y_i)_{i=1}^T$ and static model parameters θ. Given the static parameters, the transition distributions f_i on X_i are determined by equations 2.3, 2.4, 2.5 and 2.6:

$$f((X_i)_{i=1}^T | \theta) = \mu(X_1) \prod_{i=2}^T f_i(X_{i+1} | X_i, \theta).$$

where the initialisation μ is a draw from the latent variable prior model.

The remaining parameters of interest are distribution as follows. The prior on rates is two independent exponential distributions for the rates $\lambda_S \sim \mathcal{E}(\Psi_S)$ and $\lambda_C \sim \mathcal{E}(\Psi_C)$ (with $\Psi_C >> \Psi_S$ so that there are more singleton events than change points on average), Φ is the sequence of the times of the Poisson process $\mathcal{P}_t^C(\lambda_C)$, $(t_i)_{i=1}^T$ are sampled independently from a uniform constrained to some possibly overlapping intervals of $\mathbb{R}$ and $t_i \sim \mathcal{U}[L_i, U_i]$ if observations times are known up to an interval.

The HMM is specified for a given set of change point times Φ. From a modelling perspective, it would be equivalent to simulate $\mathcal{P}_t^C$ and Φ as part of the Hidden state, like for $\mathcal{P}_t^S$, and the choice is to some extent arbitrary.

In a Bayesian context, the posterior is

$$\begin{aligned} & \pi((Z_i)_{i=1}^T, (\rho_i)_{i=1}^T, \Phi, \lambda_C, \lambda_S, (t_i)_{i=1}^T | (y_i)_{i=1}^T) \\ \propto \quad & f((Z_i)_{i=1}^T, (\rho_i)_{i=1}^T | (t_i)_{i=1}^T, \Phi, \lambda_S) g((y_i)_{i=1}^T | Z_{1:T}) p(\Phi | \lambda_C) p(\lambda_C) p(\lambda_S) p((t_i)_{i=1}^T). \end{aligned}$$

Re-written more compactly, this is

$$\pi((X_i)_{i=1}^T, (\theta)_{i=1}^T | (y_i)_{i=1}^T) \propto g((y_i)_{i=1}^T | (Z_i)_{i=1}^T) f_i((X_i)_{i=1}^T | \theta) p(\theta).$$

Chapter 3

Inference algorithm for the latent variables process

This chapter shows how to perform inference on the Hidden Markov model parameters θ and hidden state $(X_i)_{i=1}^T$ given the observations $(y_i)_{i=1}^T$. The latent variables Z_{t_i} are discretized at the observation sampling times t_i and are included in the state $X_i = (Z_{t_i}, \rho_{t_i})$. The inference relies on a standard Particle MCMC algorithm specified in the first section. This is an MCMC algorithm on $((X_i)_{i=1}^T, \theta)$ with a particle filter as a proposal distribution on $(X_i)_{i=1}^T$. The second section shows how to improve the estimation variance with a suitable choice of new filtering distributions on $(X_i)_{i=1}^T$. The settings are non-standard for two reasons.

Firstly, the dimension of the hidden state $(X_i)_{i=1}^T$ is high. Each $X_i = (Z_{t_i}, \rho_{t_i})$ includes T latent variables matrices $Z_{t_i} \in \mathcal{M}_{n,K}(\mathbb{R})$. When $K \propto n$ such in applications of Chapter 6, the dimension grows with n^2.[1] In addition, the observations times $(t_i)_{i=1}^T$ are uncertain in some applications. Those are required to determine the HMM (otherwise, the ordered sequence of observations is not defined). They are part of the static HMM parameters $\theta = (\Phi, \lambda_C, \lambda_S, (t_i)_{i=1}^T)$, which dimension increases with T.

Secondly, the observation model has *hard obstacles*: the likelihood is zero when an observation y_i conflicts the state X_i, i.e. the partial order $P(Z_{t_i})$ does not admit y_i as a linear extension. As a consequence, when running the particle filter there is a non-zero probability that the system of particles dies out.

This should convince the reader that the estimation variance of the algorithm could be high depending on the transition distributions. This motivates the work on filtering distributions in the second section, intended to lower the variance by

[1] Fortunately, as will be detailed, the estimation variance may not be a function of Kn due to the correlation ρ between latent variables.

improving the particle proposal distribution and therefore the mixing time of the MCMC algorithm.

3.1 Particle MCMC algorithm

The basis for the inference is the Particle Marginal Metropolis-Hastings (PMMH) algorithm of Andrieu et al. [2010].[2] The particle filter is an importance sampling extension of Gordon et al. [1993] allowing for a generic transition kernel. The first section states a generic PMMH algorithm and its convergence conditions and the second section makes comments more specific to the partial order model.

3.1.1 Particle Marginal Metropolis-Hastings

3.1.1.1 Algorithm statement

As notation, the target posterior distribution is denoted by $\pi((X_i)_{i=1}^T, \theta)$, where $(X_i)_{i=1}^T$ is the hidden state, θ are static parameters and the conditioning on the observations y_i is implicit. It will be useful to refer to the marginals $\pi_\theta((X_i)_{i=1}^T)$ and $\pi(\theta)$ in

$$\pi(X_{1:T}, \theta) = \pi_\theta(X_{1:T})\pi(\theta),$$

and to decompose them as

$$\pi_\theta(X_{1:T}) = \frac{\gamma(X_{1:T}, \theta)}{\gamma(\theta)}$$

and

$$\pi(\theta) = \frac{\gamma(\theta)}{\mathcal{Z}},$$

where $\gamma(\theta)$ and $\mathcal{Z}$ are normalizing constants. Both likelihoods are intractable. Sampling from $\pi_\theta(X_{1:T})$ is also intractable. In a nutshell, PMMH is an exact MCMC algorithm where the proposal distribution on $(X_i)_{i=1}^T$ is a particle filter estimating $\pi_\theta(X_{1:T})$ and its normalizing constant.

Algorithm 4 is a particle filter with target $\pi_\theta(X_{1:T})$ where N samples or *particles* are sampled sequentially in T iterations. At the end of the i-th iteration, the sample $X_{1:i}^k$ represents an importance sample targeting a density $\pi_\theta^i(x_{1:i})$ with importance weight w_i^k. The densities π_θ^i are chosen such that $\pi_\theta^T = \pi_\theta$. While computing the weight w_i^k, the target $\pi_\theta^i(x_{1:i}) \propto \gamma_i(X_i^k, \theta)$ only needs to be known up to a normalizing constant.

[2] In practice, a conditional sampler such as Particle Gibbs with or without backwards sampling may result in better mixing. A comparison is found in Chopin [2012].

At the i-th iteration, the set of particles $(X_i^k)_{k=1}^N$ is resampled according to a distribution r and the particle weights $(W_{i-1}^k)_{k=1}^N$. Each particle is then propagated with a transition kernel M_i^θ and its new weight is computed. The random variable $A_{i-1} \sim r(\cdot|W_{i-1}^{1:N})$ is a mapping from the previous particles indexes to the resampled particle indexes such that the offspring count of particle k at step i is $O_{i-1}^k := \sum_{m=1}^N \mathbb{1}\{A_{i-1}^m = k\}$.

The particle estimates of $\pi_\theta(X_{1:T})$ and its normalizing constant are

$$\hat{\pi}_\theta^N(dx_{1:T}) := \sum_{k=1}^N W_T^k \delta_{X_{1:T}^k}(dx_{1:T})$$

and

$$\hat{\gamma}^N(\theta) := \prod_{i=1}^T \left(\frac{1}{N} \sum_{k=1}^N w_i^k \right). \tag{3.1}$$

Algorithm 5 is Particle Marginal Metropolis-Hastings (PMMH) with target $\pi(X_{1:T}, \theta)$, where q is the proposal on θ and $\hat{\pi}_\theta^N(dx_{1:T})$ is the proposal on $X_{1:T}$. Since $\hat{\pi}_\theta^N(dx_{1:T})$ is a sample from the particle filter, the state of the MCMC algorithm also includes the random variables A.

Algorithm 4 Particle filter for $\pi_\theta(X)$, Gordon et al. [1993]

Sample $X_0^k \overset{iid}{\sim} \mu$ and set $W_0^k = 1$, $k = 1, \ldots, N$
for $i = 1$ to T **do**
  Sample $A_{i-1}^{1:N} \sim r(\cdot|W_{i-1}^{1:N})$
  Sample $X_i^k \overset{iid}{\sim} M_i^\theta(\cdot|X_{i-1}^{A_{i-1}^k})$ and set $X_{1:i}^k = (X_{1:i-1}^{A_{i-1}^k}, X_i^k)$, $k = 1, \ldots, N$
  Compute weights $w_i^k = \dfrac{\gamma_i(X_{1:i}^k, \theta)}{\gamma_{i-1}(X_{1:i-1}^{A_{i-1}^k}, \theta) M_i^\theta(X_i^k | X_{1:i-1}^{A_{i-1}^k})}$
  Normalize weights $W_i^k = w_i^k / \sum_{j=1}^N w_i^j$, $k = 1, \ldots, N$
  Return $\hat{\pi}^N(dx_{1:T})$ and $\hat{\gamma}^N(\theta)$
end for

3.1.1.2 Convergence conditions

Under some conditions listed below, PMMH is an exact MCMC algorithm which converges to its target $\pi(\theta, X)$ in some sense as iteration $m \to \infty$. As notation, the support of the target π_θ^i is

$$S_i^\theta := \{x_{1:i} \mid \pi_\theta^i(x_{1:i}) > 0\}$$

Algorithm 5 PMMH for $\pi(\theta, X)$, Andrieu et al. [2010]

Sample initial values $\theta^{(0)}$, $X_{1:T}^{(0)}$.
for $m \geq 1$ **do**
  Sample $\theta^* \sim q(.|\theta^{(m-1)})$
  Sample $\hat{\pi}_{\theta^*}^N(dx_{1:T})$ and $\hat{\gamma}^N(\theta^*)$ with Algorithm 4
  Sample $X_{1:T}^* \sim \pi_{\theta^*}^N$
  With prob. $1 \wedge \frac{\hat{\gamma}^N(\theta^*)p(\theta^*)}{\hat{\gamma}(\theta^{(m-1)})p(\theta)} \frac{q(\theta^{(m-1)}|\theta^*)}{q(\theta^*|\theta^{(m-1)})}$
    $\theta^{(m)} = \theta^*$
    $X_{1:T}^{(m)} = X_{1:T}^*$
  Otherwise
    $\theta^{(m)} = \theta^{(m-1)}$
    $X_{1:T}^{(m)} = X_{1:T}^{(m-1)}$
end for

and the support of the particle proposal at the i-th iteration is

$$Q_i^\theta := \{x_{1:i} \mid \pi_\theta^{i-1}(x_{1-i})M_\theta^i(x_i|x_{1:i-1}) > 0\}.$$

The following is a set of conditions on the proposal distributions used in the literature to establish various convergence properties:

1. Importance sampling condition: $S_i^\theta \subseteq Q_i^\theta$
2. Unbiased resampling condition: $\mathbb{E}(O_i^k|W_i) = NW_i^k$
3. Technical resampling condition: $r(A_i^k = m|W_i) = W_i^m$
4. Bounded weights: for $x_{1:i} \in S_i$, $w_i(x_{1:i}) \leq C_i$
5. MCMC condition: $q(\theta^*|\theta)$ is irreducible and aperiodic

Under assumption 1 and 2, Pitt et al. [2010] (for HMM settings) and Del Moral [2004] (for more general settings) give two proofs that the particle estimate $\hat{\gamma}^N(\theta)$ is unbiased, i.e. $\mathbb{E}\{\hat{\gamma}^N(\theta)\} = \gamma(\theta)$. In light of Andrieu and Roberts [2009], the unbiased likelihood is sufficient to show convergence of PMMH Algorithm 5. An alternative pursued in Andrieu et al. [2010] is to explicitly show that under assumption 1, 2, 3 and 5, PMMH is a standard MCMC algorithm on an extended state that includes the particle filter variables A.

The Particle Independent Metropolis-Hastings (PIMH) is when θ is fixed and the target is just $\pi_\theta(X)$. Under assumptions 1, 2, 3 and 4, Andrieu et al. [2010] show that PIMH satisfies $\pi^N/q^N \leq \prod_{i=1}^T C_i/\mathcal{Z}$, resulting in uniform geometric ergodicity.

3.1.2 Application to the partial order problem

Having stated the generic PIMH algorithm and convergence conditions, this section is more specific to the partial order problem. In particular, it remains to specify the expression of the conditional likelihood γ and the proposals M_i^θ, r and q. A basic choice of proposals is given here. Alternatives to achieve a lower estimation variance will be the subject of the next section.

3.1.2.1 Proposal choice

Model The model of interest is a HMM with state $X_i = (Z_{t_i}, \rho_{t_i})$, static parameters $\theta = (\Phi_k, \lambda_C, \lambda_S, t_i)$, state transition f_i and observation model g. This corresponds to a normalization constant $\mathcal{Z} = p(y_{1:T})$ and conditional likelihood $\gamma(\theta) = p_\theta(y)p(\theta)$, where $p_\theta(y)$ is the HMM likelihood and $p(\theta)$ is the prior

$$p(\theta) = p(\Phi|\lambda_C)p(\lambda_C)p(\lambda_S)p((t_i)_{i=1}^T).$$

Furthermore,

$$\gamma(X_{1:T}, \theta) = p(\theta)f_0(X_0|\theta)\prod_{i=1}^{T} f_i(X_i|X_{i-1}, \theta)g(y_i|X_i, \theta)$$

and the particle weights for the sequential targets $\pi_\theta^i(dx_{1:i}) = \pi_\theta(dx_{1:i}|y_{1:i})$ are

$$w_i^k = \frac{f_i^k(X_i^k|X_{i-1}^{A_{i-1}^k}, \theta)g(y_i|X_i^k, \theta)}{M_i^\theta(X_i^k|X_{i-1}^{A_{i-1}^k})}. \tag{3.2}$$

Proposals and convergence In the implementation, the MCMC proposal on θ is taken as the model prior, i.e. $q(\theta^*|\theta) = p(\theta^*)$. This satisfies condition 5. Since the particle filter variables are also sampled independently, the PMMH sampler is an independent MCMC sampler. The resampling scheme $A_{i-1} \sim r(\cdot|W_{i-1}^{1:N})$ is based on *systematic resampling* by Carpenter et al. [1999] and satisfies condition 2.

The proposal on $X_i = (Z_{t_i}, \rho_{t_i})$ samples ρ_{t_i} from the model on ρ_t and then Z_{t_i} from a density $q_i(Z_{t_i}|Z_{t_{i-1}}, \rho_{t_i}, \theta)$. The proposal $q_i(Z_{t_i}|Z_{t_{i-1}}, \rho_{t_i}, \theta)$ for Z_{t_i} is the subject on the next section, but it is useful to think about the simple case when the proposal is the model transition density $q_i = f_i$ with no conditioning on the observation y_i. Such proposal satisfies condition 1: given $Z_{t_{i-1}}$, ρ_{t_i} and θ, there is a non-zero probability that y_i respects the partial order of $Z_{t_i} \sim f_i(Z_{t_i}|Z_{t_{i-1}}, \rho_{t_i}, \theta)$. Indeed, almost surely $\lambda_S > 0$ therefore with non-zero probability all paths of Z_{i-1} are updated in which case there is a probability of at least $1/n!$ that y_i respects the partial order $P(Z_i)$.

When $q_i = f_i$, the particle weights are bounded $w_i^k = g(y_i|Z_{t_i}^k) \leq 1$ (when g is the uniform on linear extensions observation model). Since assumption 4 is verified and the MCMC sampler is independent, it is uniformly geometric ergodic.[3] In the next section, the filtering distributions yield weights $w_i^k \leq g(y_i|Z_{t_i}^k)$.

3.1.2.2 Hard obstacles

For the observation model g as the uniform on linear extensions, a linear extension y_i is sampled uniformly at random from the set of linear extensions of $P(Z_{t_i})$. In particular $g(y_i|X_i) = 0$ if y_i is not a linear extension of $P(Z_{t_i})$. Therefore, unless the proposal $q_i(Z_{t_i}|Z_{t_{i-1}}, \rho_{t_i}, \theta)$ is suitably chosen, a sample $Z_{t_i} \sim q_i$ may result in a weight $w_i^k = 0$ (i.e., the inclusion $S_i^\theta \subset Q_i^\theta$ is strict). That is for example the case for the proposal $q_i = f_i$ where the particle $Z_{t_i}^k$ is sampled without conditioning on the observation y_i: the weight $w_i^k = g(y_i|Z_{t_i}^k)$ may be zero. The first step of the particle filter Algorithm 4 at which all particles die out is denoted by

$$\tau^N := \inf\{i \geq 1 : \forall k \in \{1, \ldots, N\}, w_i^k = 0\}.$$

If the event $\tau^N < T$ occurs, then the algorithm as stated is undefined because the weights normalization step $W_i^k = w_i^k / \sum_{j=1}^N w_i^j$ has a division by zero.

What to do if the particle system dies out? The particle filter is an importance sampling scheme with resampling. In pure importance sampling, if N importance weights are zero then the estimate based on those N weights is zero. The fact that this has a non-zero probability of occurring when estimating a non-zero quantity should not call into question the unbiased property of importance sampling estimates. In the case of the particle filter, it is straightforward to implement the following: if $\tau^N < T$ then stop Algorithm 4 and return $\hat{\gamma}^N(\theta) = 0$ and an arbitrary distribution $\hat{\pi}^N(dx_{1:T})$, with as a result the almost sure rejection of $(X_{1:T}^*, \theta^*)$ in Algorithm 5 [4]. Although not explicitly discussed in Andrieu et al. [2010], it is easy to see that such rule does not affect the unbiased property of the estimate $\hat{\gamma}^N(\theta)$ and therefore convergence occurs as stated.

For a given proposal and set of HMM parameters, one would expect the probability of extinction $\tau^N < T$ to decrease with the number of particles N. Del Moral and Doucet [2005] studies particle filtering with hard obstacles for the homogeneous

[3] Furthermore, the geometric rate r_T is bounded by $r_T \leq 1 - \mathcal{Z} / \prod_{i=1}^T C_i \leq 1 - p(y_{1:T})$. For $T = 1$, $p(y_{1:T}) = 1/n!$ when the latent variables correlation $\rho = 1$.

[4] An alternative rule is a rejection sampler where the particle filter is ran until $\tau^N \geq T$. However, the normalization constant of such sampler is intractable and depends on θ.

HMM case. If $\sup_x f(x, H) \leq \exp(-\alpha)$ for some $\alpha > 0$, where H is the set of hard obstacles, it is shown that the probability of extinction is less than $T\exp(-N\alpha)$ and some convergence error bounds are given in terms of α. Here the HMM is not homogeneous, but numerical practice do suggest that the probability of extinction decrease exponentially with N.

3.2 Filtering distributions

This section gives new transition distributions on the latent variables Z_{t_i} with conditioning on the observations y_i and parameters ρ_{t_i} (the latent variables correlation) and θ (the model parameters which determine the rate of change). These latent variables transition distributions will be denoted by $q_i(Z_{t_i}|y_i, Z_{t_{i-1}}, \rho_{t_i}, \theta)$.

They are to serve as transitions for the particle filter Algorithm 4, which purpose is to estimate a sequence of densities π_θ^i conditioned on the observations y_i. That is, the transition $M_i^\theta(X_i|X_{1:i-1})$ on $X_i = (Z_{t_i}, \rho_{t_i})$ is decomposed as

$$M_i^\theta(X_i|X_{1:i-1}) = p_i(\rho_{t_i}|\rho_{t_{i-1}}, \theta)q_i(Z_{t_i}|y_i, Z_{t_{i-1}}, \rho_{t_i}, \theta)$$

where q_i is the distribution on latent variables to optimise and $p_i(\rho_{t_i}|\rho_{t_{i-1}}, \theta)$ is the transition prior on ρ_{t_i} (sampled at events of the change point Poisson process $\mathcal{P}_t^C$). The particle filter variables k and A_{i-1} have been omitted from the notation of $M_i^\theta(X_i^k|X_{1:i-1}^{A_{i-1}^k})$ for clarity. Thus, the particle importance weights Equation 3.2 are

$$w_i = \frac{f_i(Z_{t_i}|Z_{t_{i-1}}, \theta)g_i(y_i|Z_{t_i}, \theta)}{q_i(Z_{t_i}|y_i, Z_{t_{i-1}}, \rho_{t_i}, \theta)}$$

where f_i is transition prior and g_i is the observation model.

The objective is to find q_i to minimise the variance of the weights w_i. This is a standard importance sampling problem and the optimal q_i^* is proportional to $f_i g_i$. Although q_i^* is intractable, the following work will give a choice of q_i which reduces the weight variance when compared to the simple choice of $q_i = f_i$ (in particular, it lowers the probability of extinction $\tau^N < T$) and is equal to the optimal q_i^* in some special cases.

To give an idea of the following work on q_i, it is helpful to recall that the transition prior f_i is the discretization of a singleton Poisson process $\mathcal{P}_t^S$ where between t_{i-1} and t_i some elements are sampled and latent variables for these elements are renewed. Sampling with q_i will proceed in two steps, Section 3.2.2.3 for the random elements S_i

and Section 3.2.2 for the latent variables $(Z_{t_i})_a$ with $a \in S_i$, such that q_i is decomposed as

$$q_i(Z_{t_i}, S_i | y_i, Z_{t_{i-1}}, \rho_{t_i}, \theta) = q_i(Z_{t_i} | y_i, Z_{t_{i-1}}, \rho_{t_i}, \theta, S_i) q_i(S_i | y_i, Z_{t_{i-1}}, \theta).$$

Simulating change points events of $\mathcal{P}_t^S$ is treated separately in Section 3.2.1. Although tractable, these new samplers are presented with statistical rather than computational efficiency in mind. Section 3.2.3 will comment on the computational cost from numerical practice.

3.2.1 Simulating change points

This section studies the case where Z_{t_i} needs to be sampled conditioned on the observation y_i but independently from the past, i.e. not conditioned on a realization $Z_{t_{i-1}}$. This is relevant for sampling at change points: if a change point occurred in $[t_{i-1}, t_i]$ then all latent variables are renewed and the conditioning on $Z_{t_{i-1}}$ can be ignored. This is similar for when the singleton events affect all elements (this occurs with probability $(1 - \exp(-\lambda_S(t_i - t_{i-1})/n))^n$). Since time dependence is not relevant in this section, the simpler notation $Z := Z_{t_i}$ and $y := y_i$ is used.

The target is $p(Z|y) = p(Z)L(y|P(Z))$ where $p(Z)$ is the static (K, ρ, n)-dimensional order model and $L(y|P(Z))$ is the likelihood $L(y|P(Z)) = \mathbb{1}\{P(Z) \uparrow y\}/|\mathcal{L}(P(Z))|$. The first section presents how to sample Z from the posterior (given y) exactly, then the second section discusses non-optimal variants with smaller computation cost of sampling. It is not suitable to use a rejection sampler where Z is sampled a number of times until $P(Z)$ admits y as a linear extension, because the probability of acceptance is low (when $\rho = 1$, it is $1/n!$).

3.2.1.1 Exact sampler

Consider the sampler q on Z as the following procedure: a latent variables matrix $\tilde{Z}$ is sampled from the prior p (not conditioned on y), one linear extension $\tilde{y}$ of $P(\tilde{Z})$ is sampled uniformly at random then the permutation from $\tilde{y}$ to y is applied to the rows of $\tilde{Z}$ (this permutes the elements of the partial order $P(\tilde{Z})$) to yield the output Z. This is illustrated Figure 3.1 where for a given Z and y, $P(Z)$ has only two linear extensions and therefore two possibilities for assigning elements such that $P(Z) \uparrow y$.

Theorem 1. *Let $q(Z|y)$ be the aforementioned sampler on the latent variables $Z \in \mathcal{M}_{n,K}$ given a linear extension y. Then $q = p$, the posterior target distribution.*

Proof. By Bayes' rule,

$$p(Z|y) = \frac{n!p(Z)}{|\mathcal{L}(P(Z))|}\mathbb{1}\{P(Z) \uparrow y\}$$

because $p(y) = 1/n!$. The numerator $n!p(Z)$ can be viewed as the density of sampling a set of latent variables. Furthermore, for a total order y and partial order P on the same set, there exist $|\mathcal{L}(P)|$ permutations of P elements such that $y \in \mathcal{L}(P)$. This can be shown by induction on n. Assuming it holds true for sets of a fixed size n, let y and P denote orders on a set A of size $n+1$. Recall the linear extension count Algorithm 3 as a recursion based on the equation

$$|\mathcal{L}(P)| = \sum_{i \in T} |\mathcal{L}(P_{-i})|$$

where P_{-i} is the suborder on $A \setminus \{i\}$ and T is the set of maximal elements $\mathcal{M}(P)$. By the induction hypothesis, for each $i \in \mathcal{M}(P)$, there are $|\mathcal{L}(P_{-i})|$ permutations on $A \setminus \{i\}$ of P_{-i}'s elements such that $y_{-i} \in \mathcal{L}(P_{-i})$. Thus there at least $|\mathcal{L}(P)|$ valid permutations. Conversely a valid permutation must map the maximal element i of y to $\mathcal{M}(P)$ and there are $|\mathcal{L}(P_{-i})|$ permutations on the remaining n elements, so there are at most $|\mathcal{L}(P)|$ valid permutations. This completes the proof by induction. □

3.2.1.2 Importance sampler

The previous sampler q is statistically optimal as an importance sampler because it is equal to the target. However it requires sampling a linear extension uniformly at random and may therefore be computationally costly. The following gives an example of an alternative importance sampler $\tilde{q}$ which sampling procedure is similar conceptually but faster to sample from. It is described in pseudo-code by Algorithm 6. The first step of $\tilde{q}$ is, as before, sampling $\tilde{Z}$ from the prior p unconditionally. It then samples an element from the maximal set of $\tilde{Z}$ uniformly at random and maps it to the maximal element of y, then repeat on the remaining rows of $\tilde{Z}$ and suborder of y. The mapping is a permutation on the rows of $\tilde{Z}$. The scalar d returned is to help computing $q(Z|y)$. By construction, the sampler has support $\text{supp}(\tilde{q}) = \text{supp}(p)$, its density is

$$q(Z|y) = dn!p(Z)\mathbb{1}\{P(Z) \uparrow y\}$$

and the importance weight is $p(Z|y)/\tilde{q}(Z|y) = 1/d|\mathcal{L}(P(Z))|$. In constrast to q, the sampler $\tilde{q}$ does not require sampling linear extension uniformly at random, but it does require counting linear extensions for computing its importance weights. However, as

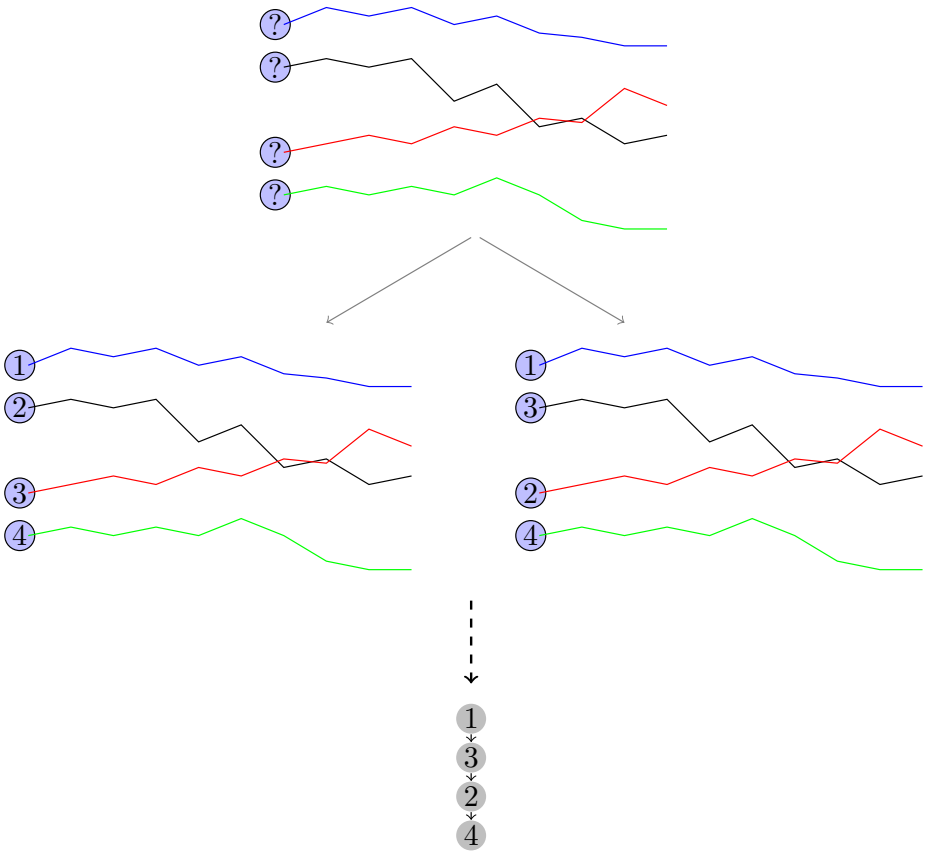

Figure 3.1: An initial set of paths (top) and the set of two possible element permutations such that Z admits y as a linear extension (below).

explained in Section 1.3.5, counting linear extensions can be significantly faster than enumerating them.[5]

3.2.2 Simulating singleton events

This sections turns to the more general case of sampling Z_{t_i} conditioned on the previous latent variables Z_{t_i} and the current observation y_i. This is relevant when the singleton Poisson process $\mathcal{P}_t^S$ has affected some but not all of the latent variables between t_{i-1} and t_i. If y_i is not a linear extension of $P(Z_{t_{i-1}})$, at least one path (the set of K latent variables of an element) requires renewal so that y_i is a linear extension of $P(Z_{t_i})$. If not, the likelihood $g_i(y_i|Z_{t_i}, \theta)$ and the importance weight w_i are zero for the sample Z_{t_i}. This is illustrated on Figure 3.2 where elements 2 and 3 are ordered as $2 \prec 3$ in Z_{t_i} and therefore in conflict with y_{i+1}, but updating their

[5] A more relevant comparison not pursued here would take into account both the weights variance and running time per sample.

Algorithm 6 Sampler $\tilde{q}$ for Z given y

Sample $\tilde{Z}$ from p unconditionally
Call ZY-SAMPLER$(\tilde{Z}, y, 1)$
procedure ZY-PERMUTATION(Z, y, d)
    If $y = \emptyset$, return Z and d
    Compute the maximal set $T(Z)$
    Sample i uniformly at random from $T(Z)$
    Set $d \leftarrow d/|T|$
    Map i to the maximal element of y, j
    Return ZY-PERMUTATION$(Z[-i], y_{-j}, d)$
end procedure

path results in a new sample $Z_{t_{i+1}}$ where they are unordered and therefore agree with y_{i+1}. If the path of only 1 or 4 are renewed, then the conflict with 2 and 3 remains.

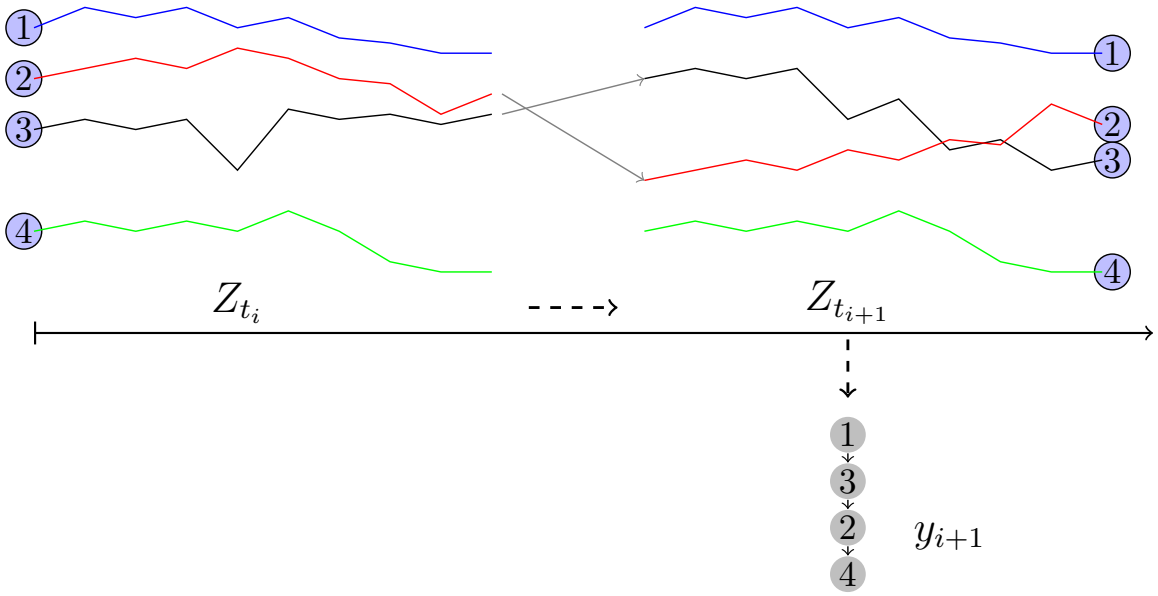

Figure 3.2: Latent variables at times t_i and t_{i+1} with $n = 4$. Variables for elements 2 and 3 are renewed between t_i and t_{i+1}.The sample Z_{t_i} does not admit y_{i+1} as a linear extension (zero likelihood), but the new sample $Z_{t_{i+1}}$ does.

The following simpler notation will be used: $Z := Z_{t_{i-1}}$ is a latent variable matrix for time t_{i-1}, $y := y_i$ is the subsequent observation at time t_i, $\rho := \rho_{t_i}$ is the correlation of the latent variables renewed by time t_i and S is the set of elements which paths are renewed in the transition from $Z = Z_{t_i}$ to $Z' = Z_{t_{i+1}}$.

3.2.2.1 Sampling latent variables after a single mutation

This section seeks a proposal $q(z_a|Z, y, \rho)$ on the K latent variables z_a of some given single element a (there is one element in the set $S = \{a\}$) conditioned on y and Z. Thus, the proposal transition is from Z to Z' where Z' a sample is equal to Z on all but the a-th element row and $Z'_{a,1:K} = z_a$. For now it is taken for granted that a has been suitably sampled such that it is possible to update the variables z_a of a such that conflicts between Z and y are removed. If it has not, the proposal given remains valid but the importance weight will be zero (and thus the corresponding particle is almost surely thrown away). The particle weights are

$$w = \frac{f(z_a|Z, \rho)g(y|Z')}{q(z_a|Z, y, \rho)} \tag{3.3}$$

where f and g are the transition and observation models, with notation for the time index i suppressed for clarity.

In the case of $n = 3$, $y = \{3 \prec 2 \prec 1\}$ and $a = 2$, then z_2 must be sampled such that z_2 lies neither entirely above $Z_{1,1:K}$ nor entirely below $Z_{3,1:K}$. Such constraints on z_2 will be referred to as a Z-constraints. It is sufficient, but not necessary, that at least one of the K features (entries) of z_2 lies between that of 1 and 3 (i.e., there exists k such that $(z_2)_k \in [Z_{3,k}, Z_{1,k}]$). A slight subtlety is for $a = 1$, where z_1 must not lie entirely below $Z_{2,1:K}$ nor $Z_{3,1:K}$: the constraints from below are formed by two rows of Z. In general, the Z-constraints from above above or below are not necessarily a whole row (of some other element). For example, when sampling z_1 it is not sufficient to look at the constraints imposed by the row of 2, the element immediately below in the order in y. If $2 \parallel 3$, i.e. the paths of elements 2 and 3 cross, then it is necessary to take into account both.

It will be shown how to sample $z_a \sim \mathcal{N}_K(0, \Sigma)$ such that z_a satisfies Z-constraints. The covariance Σ is for the equi-correlated case, $\Sigma_{i,i} = 1$ and $\Sigma_{i,j} = \rho$ for $i \neq j$. Although the ideas discussed applies to models with other multivariate distributions $f_{K,\Sigma}$ equipped with a covariance parameter, the tractability of marginals and conditionals of a multivariate normal $\mathcal{N}_K$ is convenient for the sampler discussed.

Sampling z_a such that it satisfies Z-constraints could always be achieved by rejection sampling, proposing and rejecting candidates until the Z-constraints are satisfied. For each proposed sample, it takes $O(nK)$ to test whether such constraints are satisfied by comparing it with the K latent variables of $n - 1$ elements. However the support of the target may be small, especially if the number of elements or if the feature correlation is high (for $\rho = 1$, the support is the interval between some $\mathcal{N}(0, 1)$

order statistics), therefore the rejection rate may be high. This rejection sampler $q^*(z_a|Z, y)$ has density

$$q^*(z_a|Z, y) = \frac{\mathcal{N}_K(z_a|0, \Sigma)\mathbb{1}\{Z' \uparrow y\}}{C_{Z,y,a}}$$

where the indicator $\mathbb{1}\{Z' \uparrow y\} = 1$ if z_a satisfies the Z-constraints (Z' agrees with y) and zero otherwise, and $C_{Z,y,a}$ is a normalizing constant. The sampler q^* is not the optimal particle proposal in z_a minimising the variance of the weights Equation 3.3 because it ignores the likelihood term $|\mathcal{L}(P(Z'))|$. It satisfies the Z-constraints of the likelihood and at least $w > 0$ almost surely, but it ignores the linear extension count weighting.

The following properties of a sampler q with Z-constraints are sought for:

1. Importance sampling condition: $\text{supp}(fg) \subseteq \text{supp}(q)$
2. Small total computational cost of sampling z_a and computing w
3. Small variance of w
4. Previous properties apply even for a single sample

Property 4 stresses that the problem of interest is for a single sample, rather than on average over a large number of samples (for example from MCMC, which is therefore excluded). This is because in the particle filter settings, each of the N particles requires one transition per iteration conditioned on its own Z-constraints and so the targets are different.

The following gives a sampler, the Z-sampler, with the desired properties. It is a rejection-free alternative to q^*. It makes use of the following multivariate normal decomposition,

$$\mathcal{N}_K(x|0, \Sigma) = \mathcal{N}_1(x_1|0, \Sigma_{1,1})\mathcal{N}_{K-1}(x_{2:K}|\rho x_1 \mathbb{1}_{K-1}, \tilde{\Sigma}),$$

where $\tilde{\Sigma} = \Sigma_{2:K,2:K} - \rho^2 \mathbb{1}_{K-1}\mathbb{1}_{K-1}^T$ and for clarity the notation leaves out the dependence of Σ and $\tilde{\Sigma}$ on K and ρ.

Firstly, an auxiliary variable $w \sim \mathcal{N}_{K-1}(0, \tilde{\Sigma})$ is sampled unconditionally. The final sample z_a will be a random transformation of w in order to accommodate the Z-constraints. The set of allowed transformations $x(g)\mathcal{R}^K$ are parameterized by a scalar $g \in I$ where

$$x(g) = (g, w + g\rho 1_{K-1})$$

and I is a set to be determined. The set I is the set of real numbers such that for all $g \in I$, $x(g)$ satisfies the Z-constraints. The value of g is then sampled according to a univariate $\mathcal{N}(0,1)$ conditioned on $g \in I$.

The sampler is outlined in Algorithm 7 and illustrated in Figure 3.3. The set I is a function of Z, y, a and w. It remains to check that its computation is tractable and that $z_a \sim \mathcal{N}(0,1)|\{z_a \in I\}$ is also tractable. Algorithm 8 uses Lemma 1 to compute I (the Lemma is for the special case of only two paths as constraints and the Algorithm intersects the constraints over all paths). It shows that I is an interval of the real line, therefore it is possible to sample $z_a \sim \mathcal{N}(0,1)|\{z_a \in I\}$ by standard techniques even without a rejection sampler.

It takes $\mathcal{O}(Kn)$ to determine I. One would expect that a sampling from a conditional proposal would take at least $\mathcal{O}(Kn)$, because that is also the complexity of testing whether the constraints are satisfied. Otherwise, the main cost is sampling a multivariate normal of size $K-1$ and a truncated univariate normal. This is approximately the same as sampling from a multivariate normal of size K. Looking at the sum of these costs suggests that the sampler has a complexity similar to that of the rejection sampler q^*, except that q^* may reject the sample and therefore have a running time multiple times higher.

Theorem 2 shows that the density is tractable. Importantly, in particle filtering there is no need to compute the $\mathcal{N}_K(0,\Sigma)$ density appearing in the numerator of because it cancels out with the density of f in the particle weight. The importance weight is $w = \int_{I_z} \mathcal{N}(g|0,1)dg/|\mathcal{L}(P(Z))|$ and it can be shown that its variance is lower than that of the weight for the unconditional sampler. The denominator is not a constant as it depends on z through I, therefore $q \neq q^*$ and is not the optimal importance sampler. However, in the limit $\rho \to 1$ it reduces to a univariate truncated normal and it is optimal.[6]

If the output of Algorithm 8 is not an interval (i.e., $I = [I^L, I^U]$ with $I^L > I^U$) then the constraints are impossible to satisfy (for a given w). This happens if there are conflicts between other elements, in which case z_a should not be the only variables to be renewed like assumed in this section.

Lemma 1 (Computing I for two paths constraints)**.** *Let $x(g)$ be the length K vector $x(g) = (g, w + g\rho 1_{K-1})$ and a^L and a^U be two vectors of constraints. The set I such that $x(g)$ does not lie entirely below a^L nor entirely above a^U is the interval $[I^L, I^U]$*

[6]The importance weights

Algorithm 7 Z-sampler for z_a

procedure Z-SAMPLER(Z, y, a, Σ)
  Sample $w \sim \mathcal{N}_{K-1}(0, \tilde{\Sigma})$
  Set $x = x(g)$ as $x_1 = g$ and $x_{2:K} = w + g\rho 1_{K-1}$
  Compute $I = \{g \setminus x(g)$ satisfies Z-constraints$\}$ ▷ See Algorithm 8
  Sample $z_{a,1} \sim \mathcal{N}(0,1)|\{z_{a,1} \in I\}$
  Set $z_{a,2:K} = w + z_{a,1}\rho 1_{K-1}$
  Return z_a
end procedure

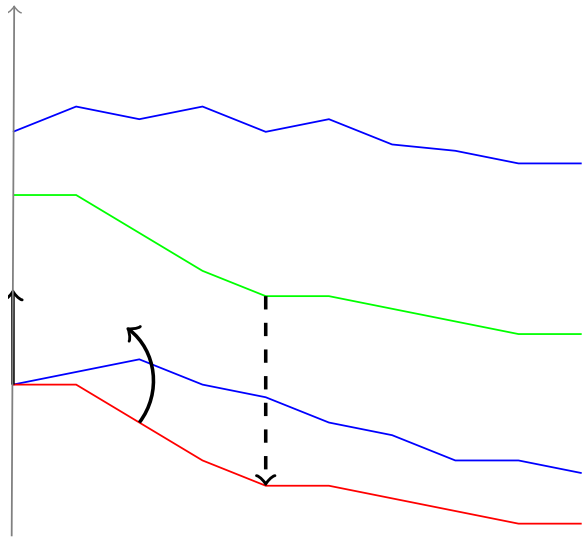

Figure 3.3: The aim is to sample a path with blue lines as Z-constraints. The green path is the initial w sample. The algorithm looks for a transformation of w. The red path is the lower limit for an acceptable solution.

given by

$$I^L = \min\{a_1^L, \frac{1}{\rho} \min_{k\in\{2,\dots,K\}} (a_k^L - w_k)\},$$

and similarly for I^U *with* min *and* a^L *replaced by* max *and* a^U.

Proof. The proof is similar to that of Lemma 2 for a related sampler. □

Theorem 2 (Properties of the Z-sampler)**.** *The density is*

$$q(z) = \frac{\mathcal{N}_K(z|0,\Sigma)\mathbb{1}\{Z' \uparrow y\}}{\int_{I_z} \mathcal{N}(g|0,1)dg} \tag{3.4}$$

and has support $\text{supp}(q) = \text{supp}(fg)$ *and a finite weight variance. In the limit* $\rho \to 1$, *it reduces to* $q = q^*$.

Algorithm 8 Computing I for Algorithm 7

Set $I = [I^L, I^U]$ with $I^L = -\infty$ and $I^U = +\infty$
for b in y **do**
  if $b \prec_y a$ **then**
    Compute the lower bound l with $Z_{b,1:K}$ as constraint ▷ See Lemma 1
    Set $I^L \leftarrow \max(I^L, l)$
  else
    Compute the upper bound u with $Z_{b,1:K}$ as constraint ▷ See Lemma 1
    Set $I^U \leftarrow \min(I^U, u)$
  end if
end for
Return I

Proof. The derivation of the density is similar to that in Proposition 4 for a related sampler. The integration domain I_z in the denominator is an interval that depends on z, Z and y. By construction, a sample from the Z-sampler satisfies Z-constraints, therefore $\text{supp}(q) \subseteq \text{supp}(fg)$. The reciprocal also holds because any $z \in \mathbb{R}^K$ satisfying Z-constraints can be written in terms of a $x(g)$ and w. When $\rho = 1$, both the Z-sampler and q^* both reduce to the same univariate truncated normal.

□

3.2.2.2 Sampling latent variables after multiple mutations

This section gives proposal distributions on Z' given Z and y for the case when there is more than one element which paths are renewed has cardinal ($|S| > 1$). In the previous section with a single element, Z-sampler was introduced as a proposal $q(z_a|Z, y, \rho)$ which have same support as the target (if a is a suitable element). What if now the latent variables z_a for $a \in S$ where each sampled independently with $q(z_a|Z, y, \rho)$? The density would be

$$q(Z'|Z, y, \rho, S) = \prod_{a \in S} q(z_a|Z, y, \rho) \tag{3.5}$$

where $q(z_a|Z, y, \rho)$ is the Z-sampler. Unfortunately, Z' from q may still conflict y because orders between new paths z_a with $a \in S$ may conflict y, even if their orders with paths of other elements agree with y.

The next step is to consider the case when elements in S share the same Z-constraints or *block*. In other words, elements in S are consecutive elements in y. For example if $y = \{a_1 \prec a_2 \prec \ldots \prec a_n\}$, then $S = \{a_k, a_{k+1}, a_{k+2}\}$ is a set of consecutive elements, while $S = \{a_k, a_{k+2}, a_{k+3}\}$ is not. When elements of S are all in the same

block, the situation is similar to that of change points Section 3.2.1: one can sample an initial variable $\tilde{Z}$ from the independent sampler 3.5, form a partial order $P(\tilde{Z})[S]$ on S and then permute its elements S such that it agrees with $y[S]$ (the ordering of S by y). By construction Z' agrees with y. Such block sampler has density

$$q(Z'|Z, y, \rho, S) = |S|! \frac{\prod_{a \in S} q(z_a|Z, y, \rho)}{|\mathcal{L}(P(Z')[S])|}. \tag{3.6}$$

In general, elements in S belong to multiple blocks $B_1, \ldots, B_m$. If each block is sampled independently with the block sampler Equation 3.6, then renewed paths from different blocks could lead to a conflict with y. This is *contamination* between blocks, when a path z_a satisfying the Z-constraints it was sampled for may cross into another block. A solution is to sample each block independently, form the partial with both fixed elements and those to be permuted, then pick a suitable permutation. An example is shown on Figure 3.4, where the three coloured paths are new samples. The blue path satisfies the constraints of both blocks B_1 and B_2. Upon inspection of the linear extension and the latent variables on the Figure, it is possible to determine which Z-constraints were used for each paths (there is only one possible partition into blocks) and the density is $\prod_{i=1}^{2} q(Z'|Z, y, S = B_i)$ with q as in Equation 3.6.

Although it is possible to always output a sample agreeing with y with such approach, sampling is more computationally demanding than the independent sampler Equation 3.5. When $\rho = 1$ (almost surely no paths crossings), the probability of success for the independent sampler is $1/|B_1|! \ldots |B_m|!$, where $B_1, \ldots, B_m$ is the partition of S into blocks and $|B_1| + \ldots + |B_m| = |S|$. And it is only $1/|S|!$ for the sampler on Z' as the transition prior (not conditioned on y).

3.2.2.3 Sampling elements

This section focusses on sampling the sets S of elements which latent variables are to be renewed. It will give a sampler on S with same support as the posterior. Although tractable, a negative result is that the sampler remains computationally intensive. In practice, the simplified version given in Section 3.2.3 will be preferred.

The aim is sampling S given y and Z that would allow Z' to have y as linear extension, depending on how the latent variables of S are sampled. For example on Figure 3.2, S should be any subset of $\{1, 2, 3, 4\}$ that includes 2 or 3 (or both) otherwise the order $3 \prec 2$ will conflict y irrespective of if and how the latent variables of other elements are sampled. In other words, S must satisfy $g(y|Z, S) := \int g(y|Z) f(Z'|Z, S) dZ' > 0$ where $g(y|Z)$ is the likelihood and $f(Z'|Z, S)$

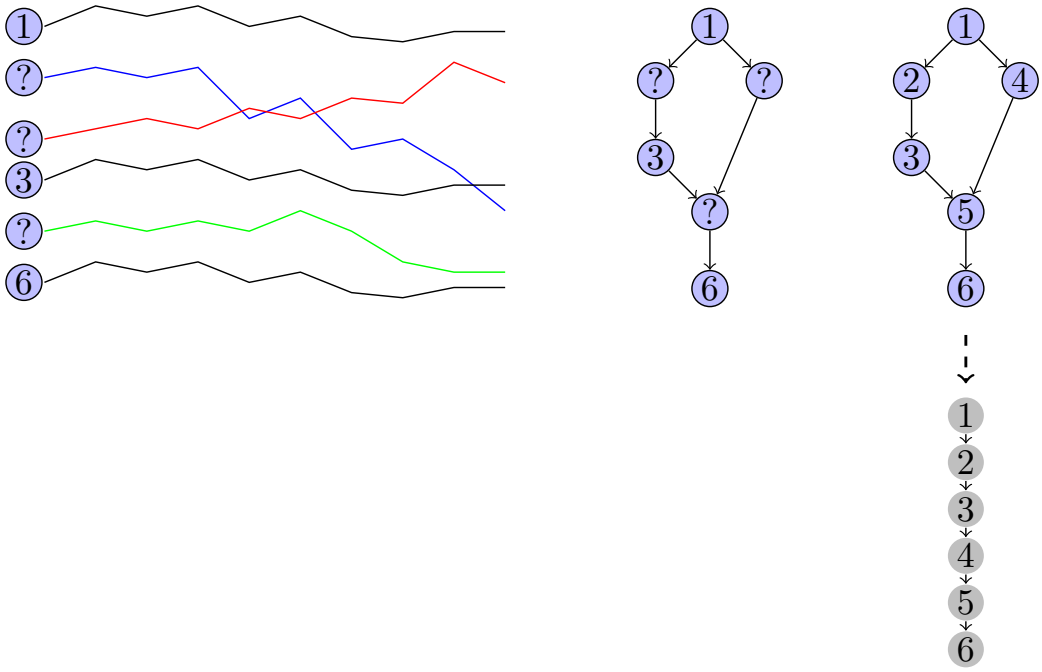

Figure 3.4: (The paths in black are fixed, defining two blocks B_1 (between 1 and 3) and B_2 (between 3 and 6). The coloured paths are new samples to be mapped to elements 2, 4 and 5. On the right, elements are mapped.

is the prior transition conditioned on the latent variables of S (only) being renewed. In the following $\tau = \tau(Z, y)$ denotes the set of such sets S:

$$\tau := \{S \mid g(y|Z, S) > 0\}.$$

If y is already a linear extension of $P(Z)$, then $S = \emptyset \in \tau$ (the converse also holds). And if $S = \emptyset$ is sampled, then $Z = Z'$. It always holds that $S = \{1, \ldots, n\} \in \tau$. However, posterior probability of sampling a large set is small if the mutation rate λ_S of the Poisson process $\mathcal{P}_t^S$ is small. It will therefore be of interest to know the sets in τ which do not have *superfluous* elements. Let τ_m denote the set of sets S such that $g(y|Z, S) > 0$ and that do not admit a strict subset $S' \subset S$ such that $p(y|Z, S') > 0$. Any element of τ is the union of an element of τ_m and of a (possibly empty) subset of $\{1, \ldots, n\}$. If y is already a linear extension of $P(Z)$ then $\tau_m = \emptyset$, but otherwise $|\tau_m| \geq 2$ because any order $a \prec b$ depends on the values of the latent variables of either element a or b.

Consider the following undirected graph of *conflicts* G_τ (a function of y and Z): vertices a and b in $\{1, \ldots, n\}$ are connected if and only if $a \prec b$ in $P(Z)$ and it conflicts y (i.e., $b \prec_y a$ in y and $a \prec_P b$). For an undirected graph in general, a *vertex cover* is a set of vertices such that every edge of the graph is adjacent to at least one vertex of

that set. The set of covers is equal to τ. The posterior on the set of elements to renew S has support equal to τ. A *minimum vertex cover* is a vertex cover of minimum size. The set of minimum vertex covers of G_τ is in general a strict subset of τ_m. A proposal for sampling S from the set of minimum covers only may result in a positive likelihood, but would violate the importance sampling condition of a support at least as large as that of the posterior. However, the support of the posterior on S only includes sets of at least the size of a minimum cover of G_τ.

Algorithm 9 gives a procedure for sampling from a proposal distribution q on S. It is a rejection free algorithm for sampling S like the Poisson process $\mathcal{P}_t^S$ does conditioned on having the same support as the posterior. When sampling Z_{t_i} given y_{t_i} and $Z_{t_{i-1}}$, the parameter $\lambda := \lambda_S(t_{i+1} - t_i)$ should be used. The purpose is to simulate S from the singleton event Poisson process $\mathcal{P}_t^S(\lambda_S)$ conditioned on $S \in \tau$ (the support of the posterior) and on being of a suitable size. Indeed it is desirable to sampled S large enough to be in the posterior support (with the conditioning on n_S) but small enough to have a high posterior probability given the rate λ (with the conditioning on n_S).

Algorithm 9 Simulating the Poisson process conditioned on a cover

Compute the conflicts graph G_τ
if $G_\tau = \emptyset$ **then**
  Return A sample S from the unconditional process $\mathcal{P}_t^S(\lambda)$
end if
Compute the size n_τ of a min cover of G_τ
Sample a Poisson count $n_S \sim \mathcal{P}(\lambda)$ such that $n_S \geq n_\tau$
Sample a uniform $S \sim \mathcal{U}[\tau]$ such that $|S| \leq n_S$
if $n_S > |S|$ **then**
  Sample $\tilde{S}$ as $n_S - |S|$ elements uniformly at random with replacement
else
  Set $\tilde{S} = \emptyset$
end if
Return $S \cup \tilde{S}$

3.2.3 Implementation

Some of the samplers introduced allow for sampling Z' conditioned on it agreeing with y. In that case, almost surely the importance weights are positive and the particles avoid the hard obstacles. However, they are computationally costly. This is a general trade-off: for a fixed number of particles N, optimizing the particle transition

proposal may result in lower variance of the importance weights but may be costly computationally.

In the applications Chapter 6, the typical settings are $n = 10$ elements, a prior on latent variables correlation ρ with a high mean close to 1, and priors on rates λ_S and λ_C such that there is on average one singleton event between two observation times and one change point in total according to the prior. Numerical practice led to the following choice of proposals. When sampling Z', it is first tested whether Z conflicts y. If yes, a set S is sampled such that its size is a Poisson count conditioned on being at least 1 and its elements are sampled uniformly at random from elements ordered in a conflict with y. The latent variables for elements in S are sampled with the Z-sampler. Chapter 4 will feature numerical experiments on a related sampler. The change point process is only simulated conditionally on y with an importance sampler when y is an order on a large number of elements (when the probability that a new Z' sample respects y by chance is small).

Chapter 4

Fast simulation of truncated multivariate normal probabilities

In Chapter 3, we presented a multivariate sampler for the rows of the latent variables matrix Z_t such that the partial order $P(Z_t)$ admits y_t as linear extension. It turns out that with a minor change, such sampler can be used for a better-known problem: estimating truncated multivariate normal probabilities. In this chapter we study the sampler as a novel importance sampling proposal distribution for simulating truncated multivariate normal densities. We state some existing importance samplers, present our importance sampler and some extensions and then compare the samplers on numerical examples.

The aim is unbiased and fast estimation of integrals involving multivariate normal distributions. Many methods have been proposed to treat this classical problem and different samplers are most efficient in different settings. Our sampler has lower estimation variance (for a given computational cost) than competing samplers in some cases when the correlation between variables is high. It uses a sample decomposition into two sets of variables, one unconditional and one as pivot to accommodate the constraints. This is conceptually similar to McFadden's Parabolic Cylinder Function simulator, but with a different decomposition and numerical behaviour.

4.1 Introduction

The aim is unbiased estimation of truncated multivariate normal probabilities A of the form

$$A(n, \mu, \Sigma, S, h) = \int_{x \in S} h(x) \mathcal{N}_n(x|\mu, \Sigma) dx$$

where S is set of constraints and $\mathcal{N}_n(\mu, \Sigma)$ is a multivariate distribution on $\mathbb{R}^n$ with mean μ and covariance Σ.

A classic problem is computing multivariate normal *orthant probabilities*. This corresponds to the semi-finite intervals $x_i \geq 0$ as constraint. In low dimension, some studies have given some deterministic procedures to compute it.

In the following, we take $\mu = 0$ (without loss of generality), $h(x) = 1$ and

$$S = \{x \in \mathbb{R}^n | a_i \leq x_i \leq b_i, i = 1, \ldots, n\}$$

written more compactly as $S = [a, b] \subseteq \mathbb{R}^n$ where $a \in \mathbb{R}^n \cup \{-\infty\}$ and $b \in \mathbb{R}^n \cup \{+\infty\}$. Some other constraints will also be discussed. A special case of interest is the equi-correlated case $\Sigma_{i,i} = 1$ and $\Sigma_{i,j} = \rho, i \neq j$ with high correlation $\rho \approx 1$. We focus on estimating

$$A(n, \Sigma, a, b) = \int_{x \in [a,b]} \mathcal{N}_n(x|0, \Sigma) dx$$

with N independent samples from a *proposal distribution* q as

$$\hat{A}(n, \Sigma, a, b) = \frac{1}{N} \sum_{i=1}^{N} w^i$$

where w^i are the *importance weights* defined by $w^i = \tilde{p}(x^i)/q(x^i)$ and $\tilde{p}(x) = \mathcal{N}_n(x|0, \Sigma)\mathbb{1}\{x \in S\}$. The (intractable) proposal distribution minimizing the estimation variance $\mathbb{V}\{\hat{A}\}$ is

$$q^*(x) = \frac{\mathcal{N}_n(x|0, \Sigma)\mathbb{1}\{x \in S\}}{c} \tag{4.1}$$

where $c = c(n, \Sigma, S)$ is a normalizing constant (and clearly $c = A$).

The following are the desired properties of a proposal q, where "fast" applies to each sample (rather than just on average):

- $\mathbb{E}\{\hat{A}\} = A$ and $\mathbb{V}\{\hat{A}\} < \infty$
- $\mathbb{V}\{\hat{A}\}$ is small
- Sampling $x \sim q$ is fast
- Computing $w(x)$ for a given sample x from q is fast

Condition $\mathbb{V}\{\hat{A}\} < \infty$ are satisfied if and only if $\int_S \tilde{p}^2/q < \infty$. In particular, $S \subseteq \text{supp}(q)$ is required.

MCMC samplers are excluded from this problem, firstly because they estimate A with a bias (that becomes zero only asymptotically) and secondly because getting a single sample x can take time due to the burn-in time. MCMC samplers aim to sample from the target p exactly asymptotically as the running time increases, contrary to samplers presented here which focus on estimating A.

In the following, a new sampler Z-SAMPLER is suggested. It is intended to be advantageous when the n variables are highly correlated and when constraints S are either semi-infinite or of a non-convex type that will be defined. The proposal is equal to the optimal sampler $\tilde{p}$ in some limiting cases, such as when correlations are 1 or when there are no constraints ($S = (-\infty, +\infty)$).

As notation appearing in several sections, for an interval $I \subset \mathbb{R}$, $\Phi(I)$ is the probability of I under a standard normal distribution,

$$\Phi(I) = \int_I \mathcal{N}_1(x|0,1)dx. \tag{4.2}$$

4.2 Existing samplers

Hajivassiliou [1996] reviews some samplers for computing multivariate normal probabilities and some of these are outlined here similarly. The Geweke-Hajivassiliou-Keane (GHK) sampler is one of the simplest and obtained a high score in their numerical experiments. It is a benchmark for the Z-SAMPLER. The Parabolic Cylinder Function (PCF) sampler also obtained a high score, but for estimating a quantity different than A. However, it is conceptually close to the Z-SAMPLER. These samplers are therefore outlined below.

More recent samplers based on MCMC have been studied. However, as explained previously, they are excluded from this study.

4.2.1 Geweke-Hajivassiliou-Keane (GHK)

The GHK (Geweke 1989) sampler transforms n independent $\mathcal{N}(0,1)$ variables η and computes their constraints recursively over the n dimensions. $\Sigma = \Gamma\Gamma^T$ is a Cholesky decomposition of Σ, where Γ is the lower triangular matrix. Writing

$$x = \Gamma\eta,$$

the constraints on η follow a recursion: for $i = 1, \ldots, n$,

$$(a_i - \Gamma_{i,1:n}\eta_i)/\Gamma_{i,i} < \eta_i < (b_i - \Gamma_{i,1:n}\eta_i)/\Gamma_{i,i}$$

where $\Gamma_{i,1:n}$ is the i-th row of Γ (of which only i elements are non-zero).

As notation, these sets are denoted by B_i. Since B_i is an interval, sampling $\eta_i \sim \mathcal{N}(0,1)|\{\eta_i \in B_i\}$ can be done with standard methods (without a rejection sampler). This suggests the estimate

$$\hat{A} = \frac{1}{N}\sum_{i=1}^{N} w(\eta^i)$$

where $w(\eta) = \prod_{j=1}^{n} \Phi(\eta_j)$ and as in Equation 4.2, $\Phi(I)$ denotes the probability of I under a standard normal distribution.

Limiting cases The optimal distribution in the case when there are no constraints ($S = \mathbb{R}^n$) or when there is no correlation ($\Sigma = I$) is a product of independent univariate normal distributions (truncated in the latter case). In both cases, GHK reduces to such optimal distribution and has zero variance ($\hat{A} = A$).

When correlation Σ is equi-correlated with correlation $\rho \to 1$, the optimal distribution reduces to sampling from a univariate normal distribution truncated to $[\max(a), \min(b)]$ (provided that $\min(b) > \max(a)$]). However, GHK proceeds recursively with the first sample sampled in $[a_1, b_1]$. Therefore a GHK sample may have support outside $[\max(a), \min(b)]$, and is therefore not optimal.

4.2.2 Spherical decompositions

The sampler that will be presented seems conceptually similar to a small number of samplers in the literature. They are reviewed here. What they have in common is decomposing a sample x into components of different nature, for example a direction and a distance, each with a different proposal or estimation procedure.

The following requires sampling on the n-sphere uniformly at random. By Knuth [1969], this can be done by sampling a set of standard normal variables and normalizing by the norm.

Deak's decomposition The vector $x \in \mathbb{R}^n$ is decomposed as

$$x = r\Gamma s$$

where Γ is a Cholesky decomposition of $\Sigma = \Gamma\Gamma^T$, $r \in \mathbb{R}$ is a χ^2 distributed variable with n degrees of freedom and $s \in \mathbb{R}^n$ is uniformly distributed on the n-dimensional unit sphere $\mathcal{S}_n$ (not to be confused with S, the set of constraints).

Deak (1980) showed that

$$A = \int_{\mathcal{S}_n} \int_{[L(s),U(s)]} dF_{\chi^2}(r) dF_{\mathcal{S}_n}(s)$$

where $L(s) = \min(r | r \geq 0, a \leq r\Gamma s \leq b)$ and $U(s) = \max(r | r \geq 0, a \leq r\Gamma s \leq b)$.

The $\int_{[L(s),U(s)]} dF_{\chi^2}(r)$ integral is over one dimension and can be expressed in terms of the χ^2 cumulative distribution function. This suggests an estimate of the form

$$\hat{A} = \frac{1}{N} \sum_{i=1}^{N} w(s^i)$$

where $s^i \sim \mathcal{U}[\mathcal{S}_n]$ and

$$w(s) = \int_{[L(s),U(s)]} dF_{\chi^2}(r).$$

McFadden's decomposition (PCF) The Parabolic Cylinder Function (PCF) sampler of McFadden (1989) decomposes x as

$$x = b + rs$$

where $b \in \mathbb{R}^n$ is the upper limit of the constraints $S = [a, b]$, $r \in \mathbb{R}$ is positive and $s \in \mathcal{S}_n$ is in the unit sphere in $\mathbb{R}^n$. The Jacobian of x to $(s_1, \ldots, s_{n-1}, r)$ is r^{n-1}/s_n, where $s_n^2 = 1 - s_1^2 - \ldots - s_{n-1}^2$.

The multivariate density is decomposed as $\mathcal{N}_n(x|0, \Sigma) = q_{b,\Sigma}(s) f(r|s)$ where

$$f_{b,\Sigma}(r|s) = r^{n-1} \exp\{-(r + b^T\Sigma^{-1}s/s^T\Sigma^{-1}s)^2(s^T\Sigma^{-1}s)/2\}/K(s),$$

$K(s)$ is a normalizing constant and

$$q_{b,\Sigma}(s) = K(s)(2\pi)^{-n/2}|\Sigma|^{-1/2} \exp\{-b^T\Sigma^{-1}b/2 + (s^T\Sigma^{-1}b)^2/2s^T\Sigma^{-1}s\}/s_n.$$

When $b = 0$ and $\Sigma = I_{n,n}$, q is the uniform distribution on the unit sphere.

$$A = \int_{s<0} \int_{[0,U(s)]} f_{b,\Sigma}(r|s) q_{b,\Sigma}(s) dr ds,$$

where

$$U(s) = \min_i (a_i - b_i)/s_i.$$

This suggests an estimate of the form

$$\hat{A} = \frac{1}{N} \sum_{i=1}^{N} w(s^i)$$

where s^i is sampled uniformly on intersection of the unit sphere and the negative orthant ($s \sim q(s|0, I)|\{s < 0\}$), and

$$w(s) = \frac{q(s|b, \Sigma)}{2^n q(s|0, I)} \int_{[0,U(s)]} f_{b,\Sigma}(r|s) dr.$$

Here, $q(s|0, I)$ is equal to the uniform distribution on the unit sphere. The scalar integral $\int_{[0,U(s)]} f_{b,\Sigma}(r|s) dr$ can be computed with a recursion over n, due to integration by parts. When $b = 0$, it is proportional to a χ^2 cumulative distribution function.

Deak vs McFadden Both decompositions use a similar spherical transformation. At the cost of increased computation, McFadden improves Deak's decomposition in the following sense. In Deak's decomposition, if for a sample s the variables $L(s)$ and $U(s)$ do not form a positive interval $[L(s), U(s)]$, then the importance weight $w(s) = 0$. In contrast, in McFadden's decomposition, $U(s) > 0$ therefore $[0, U(s)]$ is an interval and weights $w(s)$ always have a positive contribution to $\hat{A}$, which suggests (but does not imply) a lower estimation variance.

4.2.2.1 Limiting cases

Semi-infinite constraints Here, $S = [a, +\infty)$ and $a \in \mathbb{R}^n$. In Deak's decomposition, semi-infinite constraints do not guarantee that $[L(s), U(s)]$ is an interval. In McFadden's decomposition, $[0, U(s)]$ is always an interval and $w(s) > 0$ (unless $\exists i, a_i = b_i$). If $a = -\infty$ then $U(s) = +\infty$. If $b = +\infty$ then the transformation $x = a + rs$ is used instead and similar results follow.

No constraints When there are no constraints ($S = \mathbb{R}^n$), Deak's decomposition is optimal ($\hat{A} = A = 1$). This is not the case for McFadden's decomposition, as should be clear from the following: s is sampled uniformly on the unit sphere (no dependence on Σ) and the importance weight become

$$w(s) = \frac{q(s|b, \Sigma)}{2^n q(s|0, I)}.$$

High correlation We discuss the equi-correlated case with $\rho \approx 1$. In McFadden's decomposition, a only enters the calculations to determine $U(s)$ and sample r given s. The input of the constraints a and b are asymmetric in sampling and importance weight computation. This asymmetry suggests that the sampler may not be optimal as $\rho \to 1$, when the optimal distribution (discussed in the limited cases paragraph of GHK) treats a and b analogously.

Another way to see this non-optimality is that the ratio $q(s|b, \Sigma)/q(s|0, I)$ appearing in the importance weights may exhibit variability (because $\Sigma \neq I$ for the correlated case), which suggests a non-zero variance. The following confirms it even for the case $b = 0$ (a best case for which value of b of the target matches that of the proposal).

When $b = 0$ the mode of $q(s|b = 0, \Sigma)$ is the minimizer of $s^T \Sigma^{-1} s$ with constraint $s^T s = 1$. In the equi-correlated case with unit variance and correlation ρ, this is

$s_i = 1/\sqrt{n}$ (up to a sign). As $\rho \to 1$ for fixed n and $b = 0$, the expression $q(s)f(r|s)$ tends to

$$(2\pi)^{-n/2}|\Sigma|^{-1/2}\sqrt{n}r^{n-1}\exp(-r/2n)\delta_{1/\sqrt{n}}(|s|).$$

This uses the fact that the two eigenvalues of the one's matrix is 0 and n, therefore $s^T\Sigma^{-1}s$ is infinite when $\rho = 1$ unless s is spanned by eigenvectors of Σ^{-1} of the $1/n$ eigenvalue. In such limit, sampling $s \sim q(s|0, I)$ (instead of $s_i = 1/\sqrt{n}$) is therefore not optimal and the ratio $q(s|b = 0, \Sigma)/q(s|0, I)$ may exhibit high variability.

A question is whether PCF could be modified such that this ratio is less variable in the limit $\rho \to \infty$, by sampling s from a distribution other than $q(s|0, I)$. For example, with q as sampling s_i near $1/\sqrt{n}$ and then as previously $r \sim f(r|s)|\{r \in [0, U(s)]\}$. Sampling from q would be tractable (just like simulating the uniform on the n-sphere $q(s|0, I)$). However it has a support $\text{supp}(q) \subset \text{supp}(p)$ which violates importance sampling conditions.

4.3 Semi-finite intervals

This section is on estimating

$$A(n, \Sigma, a, b) = \int_{x \in S} \mathcal{N}_n(x|0, \Sigma)dx$$

for the case where $S = [a, +\infty)$ and $a \in \mathbb{R}^n$ (ie, S is the set of x such that $x_i \geq a_i$). The main ideas of the Z-SAMPLER are presented here. It is a special case of the Z-SAMPLER for finite intervals $S = [a, b]$ presented subsequently. For semi-infinite intervals S and when the correlations in Σ (or its off-diagonal elements) are all of the same sign, the support of the Z-SAMPLER is S (exactly) and in particular every sample x^i output by the sampler has a positive importance weight $w(x^i) > 0$.

4.3.1 Equi-correlated case

To improve readability, the special case of an equi-correlated covariance is presented first. Assuming unit variance $\Sigma_{i,i} = 1$ and identical correlations $\Sigma_{i,j} = \rho, i \neq j$ with $0 < \rho < 1$, we consider sampling $x \sim q$ with the Z-SAMPLER(ρ, a) described in Algorithm 4.3.1 and illustrated on Figure 4.1.

An initial sampled $w \sim \mathcal{N}_{n-1}(0, \tilde{\Sigma})$ is sampled unconditionally. Given w, $x(g)$ defined by

$$x : g \mapsto (g, w + g\rho\mathbb{1}_{n-1})$$

Algorithm 10 Z-SAMPLER(ρ, a), $S = [a, +\infty)$ and equi-correlated

Set $\tilde{\Sigma} = \Sigma_{2:n,2:n} - \rho^2 \mathbb{1}_{n-1}\mathbb{1}_{n-1}^T$
Sample $w \sim \mathcal{N}_{n-1}(0, \tilde{\Sigma})$ ▷ Auxiliary variable
Set $x : g \mapsto (g, w + g\rho\mathbb{1}_{n-1})$ ▷ Function $\mathbb{R} \mapsto \mathbb{R}^n$
Compute $I = \{g \in \mathbb{R} | x(g) \in S\}$ ▷ See Lemma 2
Sample $g \sim \mathcal{N}(0, 1) | \{g \in I\}$
Return $x(g)$, I

is a transformation (translation along dimension 1 and along dimensions 2 to n) parametrized by $g \in \mathbb{R}$. I is the set of admissible g such that $x(g) \in S$. A value of g is then sampled from a (univariate) standard normal distribution such that $g \in I$. Since I is an interval, such sampling is tractable.

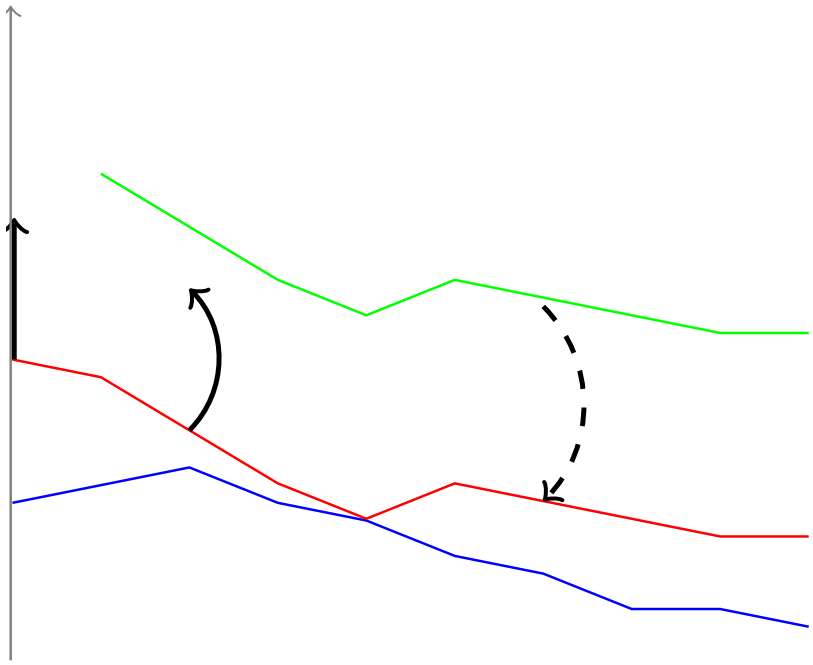

Figure 4.1: The aim is to sample a line above the constraints (lower bound) shown in blue. The green path is the initial w sample. The algorithm looks for a transformation $x(g)$ of w such that $x(g)$ satisfies the constraints. The red path is the lower limit for an acceptable solution.

Lemma 2. *[Computing I] When $0 < \rho < 1$, $I(w)$ is the interval $[L, +\infty[$ where*

$$L = \max\{a_1, \frac{1}{\rho} \max_{i \in \{2,\dots,n\}} (a_i - w_i)\}.$$

Proof. Let B denote union of a_1 and of the $n-1$ values of g such that $x_i(g) = a$ for $i = 2, \dots, n$. By the monotonicity of the n transformations $g \mapsto x_i(g)$ for $i \in \{1, \dots, n\}$, L is the maximum of B. □

Proposition 4 (Properties of Z-SAMPLER(ρ, a) Algorithm 4.3.1)**.** *Let q denote the density of* Z-SAMPLER(ρ, a)*. Its support is* $\text{supp}(q) = S$ *and its probability density function is*

$$q(x) = \frac{\mathcal{N}_n(x|0, \Sigma)\mathbb{1}\{x \in S\}}{c(x)}$$

where $c(x) = \Phi(I(x))$ and $I(x)$ is an interval determined in $\mathcal{O}(n)$ operations.

Proof. The Jacobian of the transformation $(x_1, x_{2:n}) \to (x_1, x_{2:n} - x_1\rho\mathbb{1}_{n-1})$ has a unit determinant and the density of a multivariate normal $x \sim \mathcal{N}_n(0, \Sigma)$ with equi-correlation ρ and equi-variance 1 can be written as

$$\mathcal{N}_n(x|0, \Sigma) = \mathcal{N}_1(x_1|0, 1)\mathcal{N}_{n-1}(x_{2:n}|\rho x_1\mathbb{1}_{n-1}, \tilde{\Sigma})$$

where $\tilde{\Sigma} = \Sigma - \rho^2\mathbb{1}_{n-1}\mathbb{1}_{n-1}^T$.

Lemma 2 gives a formula for $I(w)$ used by the algorithm once w has been sampled. Here, the aim is to determine I for any given $x \in \mathbb{R}^n$ (without observing w). Clearly, this can be done by setting $w = x_{2:n} - \rho x_1\mathbb{1}_{n-1}$ and using the same formula for $I(w)$. In particular, I is an interval of $\mathbb{R}$ and as a first order approximation the number of operations required increases linearly in the number of dimensions n. □

This suggests sampling (x^i, I^i) and then estimating A as

$$\hat{A} = \frac{1}{N}\sum_{i=1}^{N}\Phi(I^i).$$

Limiting cases q has the same support as the target q^* Equation 4.1 but the two distributions differ because $c(x)$ is not a constant, other than in some special cases discussed here. $\mathbb{E}\{c(x)\} = c$ under q, and if $c(x)$ is not constant then there exists some $x \in S$ such that $c(x) < c$ and some $x \in S$ such that $c(x) > c$. When $a_i = -\infty$ for all i (no constraints), then q is a multivariate normal distribution and $q = q^*$. As $\rho \to 1^-$, $L \to \max_{i\in\{1,\ldots,n\}} a_i$ and q approaches q^*.

4.3.2 General covariance case

Here, the covariance Σ is general with variances $\Sigma_{i,i} = \sigma_i^2$ and $\Sigma_{-i,i} = v_i \in \mathbb{R}^{n-1}$ denotes i-th column vector of covariances with diagonal entry σ_i^2 removed. The sampler presented has support S (exactly) if *at least one* of the columns has all positive entries, ie $\exists v_i$, $v_{i,j} > 0$ for all j. If not, with some probability some samples x^i have zero weight $w^i = 0$.

4.3.2.1 When correlations are all positive

To simplify notation we assume that the first column satisfies the all positive correlations assumption, ie $v_1 > 0$. In this case, Algorithm 4.3.1 is generalised to 4.3.2.1 and the support of the sampler remains S (exactly).

Algorithm 11 Z-SAMPLER(Σ, a), $S = [a, +\infty)$ and covariance Σ, $v_1 > 0$

Set $\tilde{\Sigma} = \Sigma_{2:n,2:n} - \frac{1}{\sigma_1^2} v_1 v_1^T$
Sample $w \sim \mathcal{N}_{n-1}(0, \tilde{\Sigma})$ ▷ Auxiliary variable
Set $x : g \mapsto (g, w + \frac{g}{\sigma_1^2} v_1)$ ▷ Function $\mathbb{R} \mapsto \mathbb{R}^n$
Compute $I = \{g \in \mathbb{R} | x(g) \in S\}$ ▷ See Lemma 3
Sample $g \sim \mathcal{N}(0,1) | \{g \in I\}$
Return $x(g)$, I

Lemma 3. *[Computing I] When $v_1 > 0$, $I(w)$ is the interval $[L, +\infty[$ where*

$$L = \max\{a_1, \sigma_1^2 \max_{i \in \{2,\dots,n\}} (\frac{a_i - w_i}{v_{1,i}})\}.$$

Proof. The formula holds for the same reason as in the equi-correlation case Lemma 2, because the assumption $v_1 > 0$ implies the monotonicity of transformations $g \mapsto x_i(g)$ for $i \in \{1, \dots, n\}$. □

4.3.2.2 When correlations are not all positive

Here, there is no assumption that there exists a column v_i with only positive entries. In particular, v_1 has mixed signs. In that case, the set I could be empty. For a given sample $w \sim \mathcal{N}_{n-1}(0, \tilde{\Sigma})$, $I \neq \emptyset$ is equivalent to the existence of $g \in \mathbb{R}$ such that

$$g \geq a_1, w + \frac{g}{\sigma_1^2} v_1 \geq a_{2:n}. \tag{4.3}$$

A necessary and sufficient condition for $I \neq \emptyset$ for all $w \in \mathbb{R}^{n-1}$ and fixed a is that $v_1 > 0$. If $v_1 > 0$, then by monotonicity of $x(g)$ a sufficiently large g results in $x(g) \geq a$. If not, then there exists some w such that $I = \emptyset$. For example, one of the form $w = a_{2:n} - \epsilon \mathbb{1}_{n-1}$. Equation 4.3 becomes $(g, gv_1/\sigma_1^2) \geq (a_1, \epsilon \mathbb{1}_{n-1})$. For ϵ large enough, $g = 0$ is not a solution, yet increasing g decreases one of the entries of gv_1 therefore there is no solution.

If the $v_1 > 0$ condition does not hold, with some probability the algorithm samples $w \sim \mathcal{N}_{n-1}(0, \tilde{\Sigma})$ such that $I = \emptyset$. This is less likely to occur when a has a *hockey stick* shape with $a_1 > 0$ and $a_{2:n} < 0$.

The solution proposed here is to simply follow Algorithm 4.3.2.1 and if $I = \emptyset$ sample $g \sim \mathcal{N}(0,1)|\{g \geq a_1\}$, in which case the importance weight of $x(g)$ is 0 (the sampler has a support larger than S). A similar situation will be encountered in the case of finite interval constraints $S = [a, b]$ in Section 4.4.

4.4 Finite intervals

This section is for the case $S = [a, b]$ where a and b are in $\mathbb{R}^n$. Algorithm 4.3.1 and Algorithm 4.3.2.1 does not apply when intervals are finite, because I may be empty depending on the sample $w \sim \mathcal{N}_{n-1}(0, \tilde{\Sigma})$. For example on Figure 4.2, w is wide therefore $x(g) = (g, w + g\rho\mathbb{1}_{n-1}) \notin S$ for all g and therefore $I = \emptyset$.

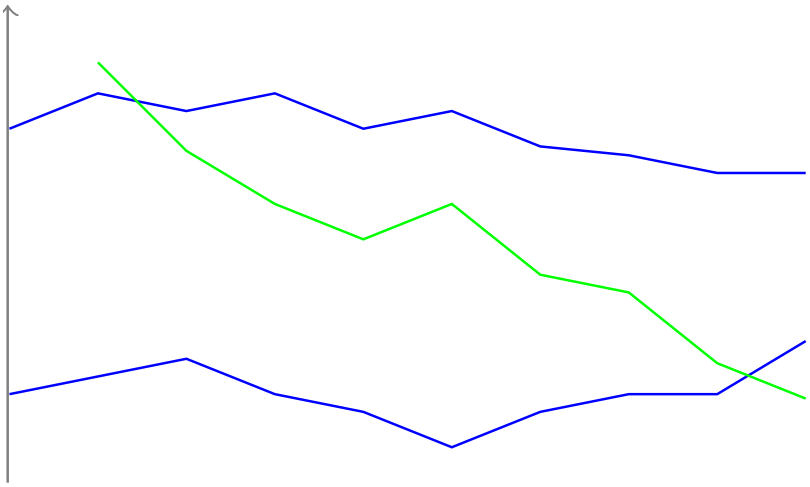

Figure 4.2: The aim is to sample a path between the two blue paths ($S = [a, b]$ constraints). The green path is the initial w sample. For such w, there does not exist a transformation $x(g) = (g, w + g\rho\mathbb{1}_{n-1})$ such that $x(g) \in S$.

4.4.1 Independent sampling

In this solution, Algorithm 4.3.1 undergoes a small modification: $w \sim \mathcal{N}_{n-1}(0, \tilde{\Sigma})$ is sampled independently as previously, if $I \neq \emptyset$ then the algorithm follows as before, and otherwise $g \sim \mathcal{N}(0,1)|\{g \in [a_1, b_1]\}$ and the output is $x(g)$ and $w = 0$. In the latter case, the importance weight is $w = 0$ because by construction there does not exist g such that $x(g) \in S$. The support includes S and the algorithm remains unbiased. This differs from the semi-infinite interval case $S = [a, +\infty)$, where for all w there exists g such that $x(g) \in I$.

Such proposal q has a support including S and has density

$$q(x) = \left(\frac{\mathbb{1}\{I(w) \neq \emptyset, x_1 \in I(w)\}}{\Phi(I(w))} + \frac{\mathbb{1}\{I(w) = \emptyset, x_1 \in [a_1, b_1]\}}{\Phi([a_1, b_1])} \right) \mathcal{N}_n(x|0, \Sigma)$$

where $w = x_{2:n} - x_1 \rho \mathbb{1}_{n-1}$. This uses the fact that $\mathcal{N}_n(x|0, \Sigma) = \mathcal{N}(x_1|0, 1)\mathcal{N}_{n-1}(w|0, \tilde{\Sigma})$. The first term corresponds to the original Z-SAMPLER(ρ, a) Algorithm 4.3.1. [1] This shows how to compute the density for an arbitrary $x \in \mathbb{R}^n$. However if one samples x with the Z-SAMPLER following the procedure, then by construction it is known which of the two indicator terms is positive and what is the value of I when the procedure ends and outputs a sample x.

For Algorithm 4.3.1 and in the case where $\rho \approx 1$, and both $\max a < 0$ and $\min b > 0$ are large in absolute value (ie, 0_n fits comfortably in S), with high probability $I \neq \emptyset$. If $\min b$ is much greater than $\max a$ then the estimation variance remains similar to that of Z-SAMPLER(ρ, a).

4.4.2 Conditional sampling

The approach in this section is to sample w from a conditional distribution, which may reduce the probability of $I(w) = \emptyset$ and increase that of a sampler x respecting the constraints $x \in S$. A bad idea would be to sample $w \sim \mathcal{N}_{n-1}(0, \tilde{\Sigma})$ conditionally on $I \neq \emptyset$, as this is a step back towards the original problem of simulating truncated multivariate normal probabilities. One difference is that the w sampling problem is smaller than the original one (dimension $n-1$ and n respectively), and this is a motivation for Section 4.5.3.

For any set $B \subseteq \mathbb{R}^n$, CONDSAMPLER(n, Σ, B) will denote any sampler suitable as importance sampling proposal targeting the distribution $\mathcal{N}_n(0, \Sigma)$ truncated to B, for example GHK. Its support is assumed to be B, but the validity of the algorithm discussed below will remain if it is larger.

Z-SAMPLER(ρ, a, b) is Algorithm 4.4.2. It differs from Z-SAMPLER(ρ, a) Algorithm 4.3.1 in that w is sampled from a conditional distribution, CONDSAMPLER$(n-1, \tilde{\Sigma}, \tilde{a}, \tilde{b})$, where

$$\tilde{a} = a_{2:n} - b_1 \rho \mathbb{1}_{n-1},$$

$$\tilde{b} = b_{2:n} - a_1 \rho \mathbb{1}_{n-1}.$$

[1] As a remark, for any x in the form $x = (g, w + g\rho\mathbb{1}_{n-1})$ where $g \in \mathbb{R}$ and $w \in \mathbb{R}^{n-1}$, $x \notin S \implies I(w) = \emptyset$ (by definition of I) but not necessarily the converse. This is why we need to compute I and the indicator functions, it is not sufficient to check whether $x \in S$.

As before $\tilde{\Sigma} = \Sigma_{2:n,2:n} - \rho^2 \mathbb{1}_{n-1}\mathbb{1}_{n-1}^T$. The set I is similar to before but with an upper bound: $I = [L, U]$ where

$$L = \max\{a_1, \frac{1}{\rho} \max_{i\in\{2,\dots,n\}} (a_i - w_i)\}$$

and

$$U = \min\{b_1, \frac{1}{\rho} \min_{i\in\{2,\dots,n\}} (b_i - w_i)\}.$$

Algorithm 12 Z-SAMPLER(ρ, a, b), $S = [a, b]$ and equi-correlated

Set $\tilde{\Sigma} = \Sigma_{2:n,2:n} - \rho^2 \mathbb{1}_{n-1}\mathbb{1}_{n-1}^T$
Set $\tilde{a} = a_{2:n} - b_1\rho\mathbb{1}_{n-1}$
Set $\tilde{b} = b_{2:n} - a_1\rho\mathbb{1}_{n-1}$
Sample $w \sim$ CONDSAMPLER$(n-1, \tilde{\Sigma}, \tilde{a}, \tilde{b})$ ▷ Auxiliary variable
Set $x : g \mapsto (g, w + g\rho\mathbb{1}_{n-1})$
Compute $I = \{g \in \mathbb{R} | x(g) \in S\}$
if $I = \emptyset$ **then** ▷ No solution for w
  Sample $g \sim \mathcal{N}(0,1)|\{g \in [a_1, b_1]\}$
else
  Sample $g \sim \mathcal{N}(0,1)|\{g \in I\}$
end if
Return $x(g)$, I

The reader may wonder why the auxiliary variable is constrained to $[\tilde{a}, \tilde{b}]$, rather than $[a_{2:n}, b_{2:n}]$ for example. A motivation is that $[\tilde{a}, \tilde{b}]$ is neither too small nor too large. If it is too large, then $I = \emptyset$ is more likely. If it is too small, then w is flat and there there may be a mismatch between the proposal and the target. The set $[\tilde{a}, \tilde{b}]$ is the smallest interval such that Z-SAMPLER(ρ, a, b) has support on at least S.

In the case when the constraints S are such that $b_i = +\infty$ for $i \geq 2$ and other constraints are finite (ie, semi-infinite $[a_i, +\infty)$ for $i \geq 2$ and also a finite interval $[a_1, b_1]$ along the first dimension), then the support of Z-SAMPLER(ρ, a, b) is S and no more (and $I(w) \neq \emptyset$ for all samples w).

Proposition 5 (Properties of Z-SAMPLER(ρ, a, b) Algorithm 4.4.2)**.** *Let q denoted the density of* Z-SAMPLER(ρ, a, b)*, given an auxiliary sampler* CONDSAMPLER$(n-1, \tilde{\Sigma}, \tilde{a}, \tilde{b})$ *with support $[\tilde{a}, \tilde{b}]$. Then* $\text{supp}(q) \supseteq S$ *and*

$$\begin{aligned} q(x) &= \frac{\mathcal{N}(x_1|0,1)\tilde{q}(w|n-1,\tilde{\Sigma},\tilde{a},\tilde{b})}{c(w)} \mathbb{1}\{I(w) \neq \emptyset\} \\ &+ \frac{\mathcal{N}_n(x|0,\Sigma)}{\Phi([a_1,b_1])} \mathbb{1}\{I(w) = \emptyset\} \end{aligned}$$

where $\tilde{q}$ is the density of the CondSampler*, $w = x_{2:n} - x_1\rho\mathbb{1}_{n-1}$, $c(w) = \Phi(I(w))$ and $I(w)$ is an interval determined in $\mathcal{O}(n)$ operations.*

In particular, the importance weight evaluated at a sample x from q is

$$w(x) = \frac{\mathcal{N}_{n-1}(w|0,\tilde{\Sigma})}{\tilde{q}(w|n-1,\tilde{\Sigma},\tilde{a},\tilde{b})}c(w)\mathbb{1}\{x \in S\}.$$

Proof. The image of $x = (g, w + g\rho\mathbb{1}_{n-1})$ over $w \in [\tilde{a}, \tilde{b}]$ and $g \in [a_1, b_1]$ is

$$L = (a_1, a_{2:n} - \rho(b_1 - a_1))$$

and

$$U = (b_1, b_{2:n} + \rho(b_1 - a_1)).$$

Since $b_1 > a_1$, it follows that $S = [a, b] \subseteq \text{supp}(q)$.

The expression given for q holds for an arbitrary $x \in \mathbb{R}^n$ and follows from inspection of the algorithm. The expression of the weight $w(x)$ evaluated at a sample x is either $w(x) = 0$ if $I(w) = \emptyset$ for the particular auxiliary variable w sampled in the procedure, or $w(x) > 0$ if $I(w) \neq \emptyset$ in which case the sample x satisfies $x \in S$.

□

Example: translated GHK As an example, CondSampler is the GHK sampler described Section 4.2.1. w is sampled given $(n-1, \tilde{\Sigma}, \tilde{a}, \tilde{b})$. In particular, its support is $[\tilde{a}, \tilde{b}]$ and the Cholesky decomposition used is of the form $\tilde{\Sigma} = \Gamma\Gamma^T$. In this case, the Z-Sampler(ρ, a, b) is some translation of a GHK sample. The importance weights for estimating A are in the form of a product of univariate standard normal probabilities

$$w(x) = \Phi(I(x))w(\eta)\mathbb{1}\{x \in S\}$$

where $w(\eta) = \prod_{j=1}^{n-1}\Phi(\eta_j)$, $w = x_{2:n} - x_1\rho\mathbb{1}_{n-1}$ and $w = \Gamma\eta$.

4.5 Extensions

The following suggests some extensions of the Z-sampler for reducing the estimation variance $\mathbb{V}\{\hat{A}\}$. We focus on the case of semi-infinite interval constraints $S = [a, +\infty)$. We sometimes use the equi-correlated case for clarity of presentation, however the generalisation to a more general covariance Σ applies straightforwardly.

The *pivot* choice extension helps to pick the pivot column less arbitrarily. In the algorithms described so far, the pivot is fixed as the first column. The extension is

expected to reduce variance especially when a has a *hockey stick* like shape (one entry a_i much greater than the others).

The *sign correction* extension discusses an alternative distribution for the auxiliary variable w. The extension is expected to reduce variance especially for *flat* constants, ie entries in a are all the same approximately.

The *multiple pivots* extension is for when there are more than one pivot column. However, numerical results show that the variance reduction is less obvious than in the previous two extensions, therefore the discussion exists more for conceptual purposes.

4.5.1 Pivot choice

As a reminder, the Z-SAMPLER(ρ, a) samples $w \sim \mathcal{N}_{n-1}(0, \tilde{\Sigma})$ unconditionally and then a transformation is sampled from the family

$$x : g \mapsto (g, w + g\rho\mathbb{1}_{n-1}).$$

The choice of the first column as a *pivot* is arbitrary. Denoting

$$w^g = w + g\rho\mathbb{1}_{n-1},$$

the same algorithm with an arbitrary pivot column i corresponds to the transformation

$$x(g) = (w_1^g, \ldots, w_{i-1}^g, g, w_i^\rho, \ldots, w_{n-1}^g)$$

and lower bound L^i for the admissible set of g as

$$L^i(x) = \max(a_i, x_i + \frac{1}{\rho}\max_{j\neq i}(a_j - x_j)).$$

In the case where $a_1 = \epsilon$ and $a_{j\geq 2} = 0$, the following shows how to pick the pivot. If the pivot is 1, then

$$L^1 = \max\{\epsilon, -\frac{1}{\rho}\max w\}$$

and if it is $i \geq 2$,

$$L^i = \max\{0, \frac{1}{\rho}(\epsilon - w_1), -\frac{1}{\rho}\max_{j\neq 1}(w_j)\}.$$

As n increases, $\max w$ and $\max_{j\neq 1}(w_j)$ increase in expectation. In the limit $n \to \infty$, L^1 and L^i have a distribution concentrating around ϵ and $\max\{0, \frac{1}{\rho}(\epsilon - w_1)\}$ respectively. This suggests less variability for L^1 and less estimation variance $\mathbb{V}\{\hat{A}\}$ when n is large, and therefore the pivot should be column 1.

For a large g and column 1 as pivot, $x(g) \approx (g, g\rho\mathbb{1}_{n-1})$ is a *hockey stick* shaped vector similar to the shape of the constraints $a_1 = \epsilon$ and $a_{j\geq 2} = 0$. This gives intuition

that the pivot should indeed correspond to the column i of the largest a_i if it is large: if a_i is large and in a low probability area of the zero mean normal distribution, then a good value for a sample $x(g) \geq a$ should *stick* to a rather than sit even further away.

Aside on the variance For a fixed ρ and the notation $c^*_{a,n} = \int \mathcal{N}_K(0, \Sigma)$ (equal to A, the quantity of interest), the denominator $c(x)$ of the density of the Z-SAMPLER(ρ, a) given in Proposition 4.

$$c(x) = \int_{L(x)}^{\max a_i} \mathcal{N}(y|0,1)dy + \int_{\max a_i}^{+\infty} \mathcal{N}(y|0,1)dy$$

where $L(x) = \max(a_1, x_1 + \frac{1}{\rho}\max_{i\geq 2}(a_i - x_i))$. Denoting $c_{x,a,n}$ and $c_{a,n}$ the first and second integral respectively,

$$\mathbb{V}\{N\hat{A}\} = (c^2_{a,n} - (c^*_{a,n})^2) + \mathbb{E}\{c^2_{x,a,n} + 2c_{a,n}c_{x,a,n}\}. \tag{4.4}$$

Both the $c^2_{a,n} - (c^*_{a,n})^2$ and expectation terms tend to 0 as $\rho \to 1$. The term $c^2_{a,n} - (c^*_{a,n})^2$ shrinks as the dimension collapses from n to 1. The expectation shrinks because $c_{x,a,n}$ does in expectation. $\mathbb{V}\{\hat{A}\}$ is reduced if $c_{x,a,n}$ is on average negative (up to a certain point), ie $L(x) > \max a_i$. When ρ is large, the variance of $c_{x,a,n}$ is small and the optimal value for $\mathbb{E}c_{x,a,n}$ is approximated by $-c_{a,n}$.

By Jensen's inequality and in the limit $n \to \infty$, $\mathbb{E}\{L^i\} \geq \epsilon/\rho$, therefore for large n, $\mathbb{E}\{L^i\} > \mathbb{E}\{L^1\}$. The recommendation to take the column of the largest a_i as pivot may at first contradict the intuition given by Equation 4.4, which suggests a reduction in variance $\mathbb{V}\{\hat{A}\}$ if $L(x)$ is on average large enough such that $c_{x,a,n}$ is negative. However, numerical results will confirm the $\max a_i$ pivot recommendation.

4.5.2 Sign correction

The choice of the distribution of the auxiliary variable as $w \sim \mathcal{N}_{n-1}(0, \tilde{\Sigma})$ is not a necessity. Such distribution is not the marginal of the optimal distribution q^* of Equation 4.1. Any other distribution could be chosen subject to the usual importance sampling restrictions including a tractable density. The choice of $w \sim \mathcal{N}_{n-1}(0, \tilde{\Sigma})$ was partly motivated by the fact that it results in simple importance weights, and more of this will be discussed in the section comparing the computational costs of competing samplers.

An alternative distribution for w is defined by $\tilde{w} \sim \mathcal{N}_{n-1}(0, \tilde{\Sigma})$ and $w = \tau\tilde{w}$ where $\tau \in \mathbb{R}$ is a scalar that could depend on $\tilde{w}$, a and Σ.

We consider the case when the constraints are flat, ie $S = [a, +\infty)$ and $a_i = a_1$ is constant. The lower bound

$$L = \max\{a_1, \sigma_1^2 \max_{i\in\{2,\dots,n\}} (\frac{a_1 - w_i}{v_{1,i}})\}.$$

is a constant in the (unlikely) event that all w_i are positive, suggesting a low estimation variance $\mathbb{V}\{\hat{A}\}$. This motivates the choice of $w = \tau\tilde{w}$ with $\tau < 0$ if the $\tilde{w}_i$ tend to be negative. For example, $\tau = \text{sgn}(\sum_i w_i)$. This is equivalent to sampling w from

$$q_{\tau,+}(w) = 2\mathcal{N}_{n-1}(w|0, \tilde{\Sigma})\mathbb{1}\{w^T\mathbb{1} > 0\}.$$

This is intuitive for the constraints considered here, $S = [a, +\infty)$, because $w^T\mathbb{1} > 0$ means that the auxiliary variable w is already oriented towards the positive quadrant once sampled and requires less transformation (a large value of g required for $x(g) \in S$ to hold).

To ensure that the support is (at least) S, w can be sampled from the following mixture distribution: $w \sim \delta q_{\tau,+} + (1-\delta)\delta q_{\tau,-}$ where $0 < \delta < 1$ and $q_{\tau,-}$ defined is as $q_{\tau,+}$ but with the indicator function replaced by $\mathbb{1}\{w^T\mathbb{1} > 0\}$. The importance weight becomes

$$w(x) = \frac{c(x)}{2\delta\mathbb{1}\{w^T\mathbb{1} > 0\} + 2(1-\delta)\mathbb{1}\{w^T\mathbb{1} < 0\}}.$$

When $\delta = 1/2$, this is equivalent to sampling $w \sim \mathcal{N}_{n-1}(0, \tilde{\Sigma})$. If $\delta = 1/2 + \epsilon$ for $\epsilon > 0$ small enough, the discussion above suggests a reduction in variance. For ϵ too large, L may exhibit less variability but the importance weights w may increase in variance due to a mismatch between the proposal and the target. δ is therefore analogous to the smoothing parameter in regression and similar caution is required when fixing its value.

4.5.3 Multiple pivots

Writing $x(g) = (0, w) + g(1, \rho\mathbb{1}_{n-1})$, the transformation $x(g)$ has one degree of freedom (for a given w). The Z-SAMPLER can be generalized to when the g has dimension 2 by using the same properties of multivariate normal distributions, here with marginal on a state of dimension 2:

$$\mathcal{N}_n(x|0, \Sigma) = \mathcal{N}_2(x_{1:2}|0, 1)\mathcal{N}_{n-2}(x_{3:n}|\tilde{\mu}, \tilde{\Sigma})$$

where in the equi-correlated case it can be shown that

$$\mu = \frac{\rho}{1+\rho}(x_1 + x_2)\mathbb{1}_{n-2}$$

and

$$\tilde{\Sigma} = \Sigma_{3:n,3:n} - \frac{2\rho^2}{1+\rho}\mathbb{1}_{n-2}.$$

Defining $x(g) = g_1(1, 0, \frac{\rho}{1+\rho}\mathbb{1}_{n-2}) + g_2(0, 1, \frac{\rho}{1+\rho}\mathbb{1}_{n-2})$ for $g = (g_1, g_2)$, the algorithm samples $w \sim \mathcal{N}_2(0|\tilde{\Sigma})$ and computes the set I of g such that $x(g) \geq a$. The density has the same form as before, namely $\mathcal{N}_n(x|0, \Sigma)/c(x)$, but $c(x)$ is now an integral over two dimensions:

$$c(x) = \int_I \mathcal{N}_2(z|0, \Sigma_{1:2,1:2})dz$$

where $\Sigma_{1:2,1:2} = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$.

This could be continued in higher dimensions. Extending this to many dimensions defeats the purpose of the original problem, because the algorithm requires sampling from truncated multivariate normal distributions and computing truncated multivariate normal probabilities $c(x)$. In low dimensions, this remains tractable with deterministic numerical algorithms. For example, Chopin [2012] is recent work on exact simulation of truncated multivariate normal distributions suitable in low dimensions.

More degrees of freedom for $x(g)$ means that the Z-SAMPLER may be suitable for a wider range of constraints. The connection between the shape of $x(g)$ and of constraints a was discussed in the pivot choice extension: if a is like a hockey stick, then $x(g)$ should follow a similar shape with the pivot along the first dimension. This suggests sorting the columns (permuting the dimensions of the state space) in decreasing order of a values, and then using the first columns as the pivots when $x(g)$ has several degrees of freedom.

However, numerical experiments on hockey sticks a of various slopes suggest that the pivot choice extension already captures most the variance reduction. A transformation with higher degrees of freedom is therefore not pursued in the rest of this study.

4.6 Some non-convex constraints

The Z-SAMPLER can be extended to when S is a non-convex set of the form C_1: given a and b, $x \in C_1$ if neither $x > b$ nor $x < a$, ie the path x must lie neither entirely above path b nor entirely below path a. In that case, Algorithm 4.3.1 can be used similar but with a slightly different set I of admissible g such that $x(g) \in S$. In

particular, whatever the auxiliary variable w, there exists a solution ($I \neq \emptyset$): I is the interval $[L, U]$ where

$$L = \min\{a_1, \frac{1}{\rho} \min_{i \in \{2,\dots,n\}} (a_i - w_i)\},$$

and similarly for I_u with min and a replaced by max and b respectively.

A related but different constraint denoted by C_2 is the following: $x \in S_2$ if at least one of the n intervals $[a, b]$ is visited, ie $\exists i, x_i \in [a_i, b_i]$ ($[a, b]$ is satisfied along at least one dimension). An algorithm computing I for such constraints is to compute the intervals of admissible values of g along each dimension separately and return the union. In general, I is the union of up to n dis-joint intervals.

As comparison, for fixed a and b, denoting $S = [a, b]$ the original convex constraints,

$$S \subseteq C_2 \subseteq C_1.$$

Clearly, spherical decompositions $x = r\Gamma s$ as in Section 4.2.2 are less appropriate for these constraints. For example, once $s \in \mathbb{R}^n$ is sampled on the unit sphere, it may take a large value of r such that $x \in C_1$. In that case, x is in a low probability area of $\mathcal{N}_n(0, \Sigma)$.

It is not clear whether there exists an extension of GHK (with a different recursion for η_i) for C_1 constraints.

4.7 Comparison

This section compares the Z-Sampler to GHK and spherical decompositions (Deak and McFadden) in theory and in practice.

4.7.1 Comments

Conceptual differences The Z-Sampler $x = (k, w + k\rho \mathbb{1}_{n-1})$ is conceptually close to the spherical decompositions $x = r\Gamma s$ (Deak) and $x = b + rs$ (McFadden). It decomposes a variable x into an auxiliary variable $w \in \mathbb{R}^{n-1}$ and a pivot variable $k \in \mathbb{R}$. w is sampled first, then g is sampled given w. In contrast, GHK is more sequential in nature and the distributions of variables across dimensions of x are similar.

The Z-Sampler and the spherical transformations correspond to addition (translation) and multiplication respectively. This implies the following differences. The Z-Sampler appears suited for semi-infinite constraints $S = [a, +\infty)$ and the non-convex constraints discussed in Section 4.6. GHK and McFadden's decomposition

appears suited for finite $S = [a, b]$ and semi-infinite $S = [a, +\infty)$. "Suited" here means that the support is S (no more).

However, a support larger than S does not necessarily imply a higher estimation variance $\mathbb{V}\hat{A}$. The Z-Sampler may have a small estimation variance for high correlations, because it reduces to the optimal proposal when $\rho = 1$ contrary to GHK and McFadden's decomposition. GHK and McFadden's decomposition appear more suited when $\Sigma = I$ as discussed in their review Section 4.2.

Computational cost A comparison between samplers needs to take into account the computational cost (rather than $\mathbb{V}\hat{A}$ for a fixed number of samples N). Contrary to GHK, the Z-Sampler requires only one cumulative normal distribution function evaluation. Another attractive feature is that sampling is also less sequential. McFadden's importance weights computation require the evaluation of a the normalizing constant $K(s)$ and matrix multiplications for each of the N samples separately (and matrix inversion as initial cost). Spherical decompositions also require an integral computation over the scaling variable r that might be more involved than a normal cdf evaluation. Sampling r to output a sample x (not required for merely computing $\hat{A}$) means simulating from the truncation of a potentially non standard univariate distribution.

4.7.2 Results

Python is used to compute the standard deviation of the estimation error of the samplers when the dimension is $n = 10$, the average is over $N = 10^5$ samples, there is equi-correlation ρ and the constraints are of the form $S = [-a, a]$. Parameters varied are ρ and a. The samplers are the naive unconditional sampler $H = \mathcal{N}_n(0, \Sigma)$, GHK, the Z-sampler and its conditional version (translated GHK). Python/numpy is used to *vectorize* the random variable generation whenever possible, for example in GHK the variable η_{i+1} along dimension $i + 1$ is sampled given η_i for all N samples at the same time. Because of the sequential nature of GHK, less vectorization is possible than with the Z-sampler.

Figure 4.3 shows that from $\rho \approx 0.7$, the Z-sampler may be preferred to GHK depending on the constraint width $S = [-w, w]$, and that from $\rho \approx 0.77$ it is uniformly preferred. The conditional version shows little advantage.

Figure 4.4 uses the same settings except that the standard deviation is multiplied by the square root of the running time. When adjusting for the running time, the

plot shows more preference to the Z-SAMPLER uniformly from $\rho \approx 0.7$. It appears that GHK does worse than the unconditional sampler at high-correlation.

Figure 4.5 compared the original Z-SAMPLER, its pivot extension and its sign correction extension with mixture weights $\delta = 0.55$ and $\delta = 0.6$ ($\delta = 1/2$ correspond to the original Z-SAMPLER). The constraints are of the form $S = [a + v, +\infty)$, where $v = (-1, -1 + \frac{2}{n-1}, \ldots, 1)$. In the pivot extension, the pivot dimension corresponds to $\max a_i$. Here, the columns are sorted such that the algorithm to using constraints $S = [a - v, +\infty)$.

The plot shows that the pivot extension is uniformly better, and that a sign correction is advised unless correlation is high. This is expected, because when correlation increases the samples are more flat and the set I has less variability, yet the introduction of the mixture distribution still contributes to the variance. A question is whether combining the pivot and sign correction extensions offers further benefits. From results not shown here, it appears that the variance can indeed be further decreased except at the correlation range when the sign correction extension increases the variance.

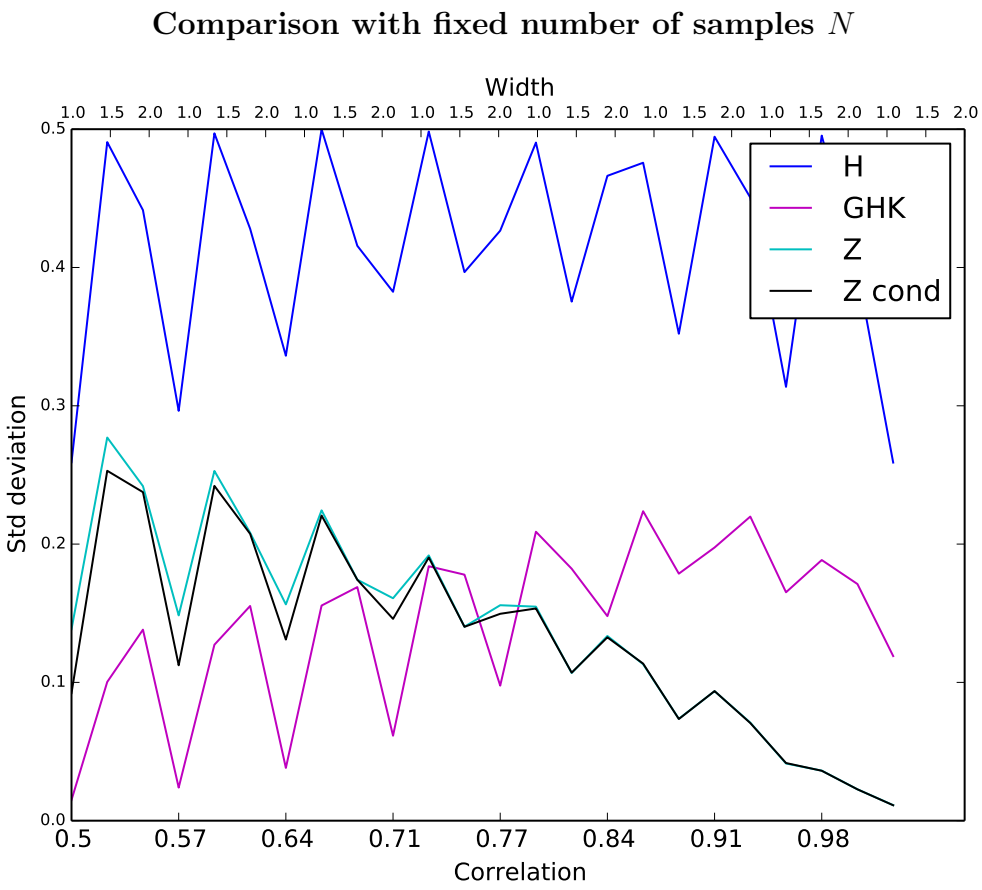

Figure 4.3: Standard deviation of the estimation error for the unconditional sampler H, GHK, Z-SAMPLER Z and its conditional version. Settings are $n = 10$, finite constraints $S = [-a, a]$ and equi-correlation ρ. Parameters varied are the width a (top axis) and the correlation ρ (bottom axis).

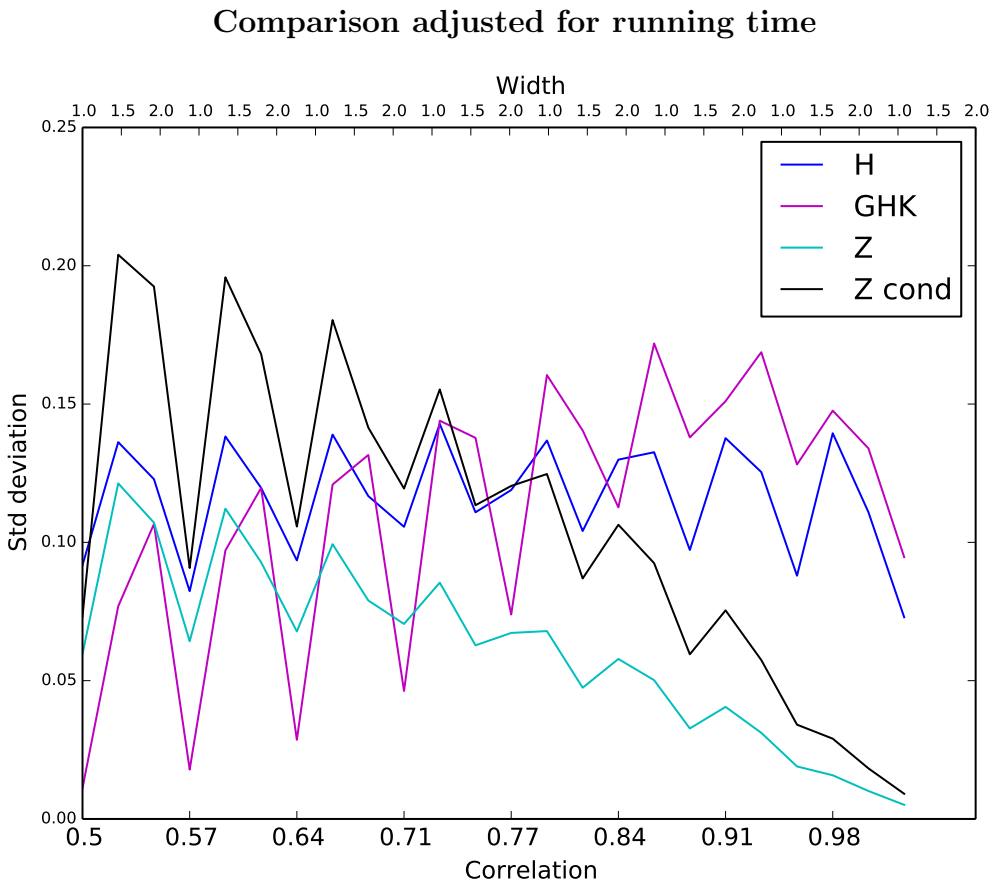

Figure 4.4: Settings are as in Figure 4.3, but the standard deviation is adjusted for the running time. Parameters varied are the width a (top axis) and the correlation ρ (bottom axis).

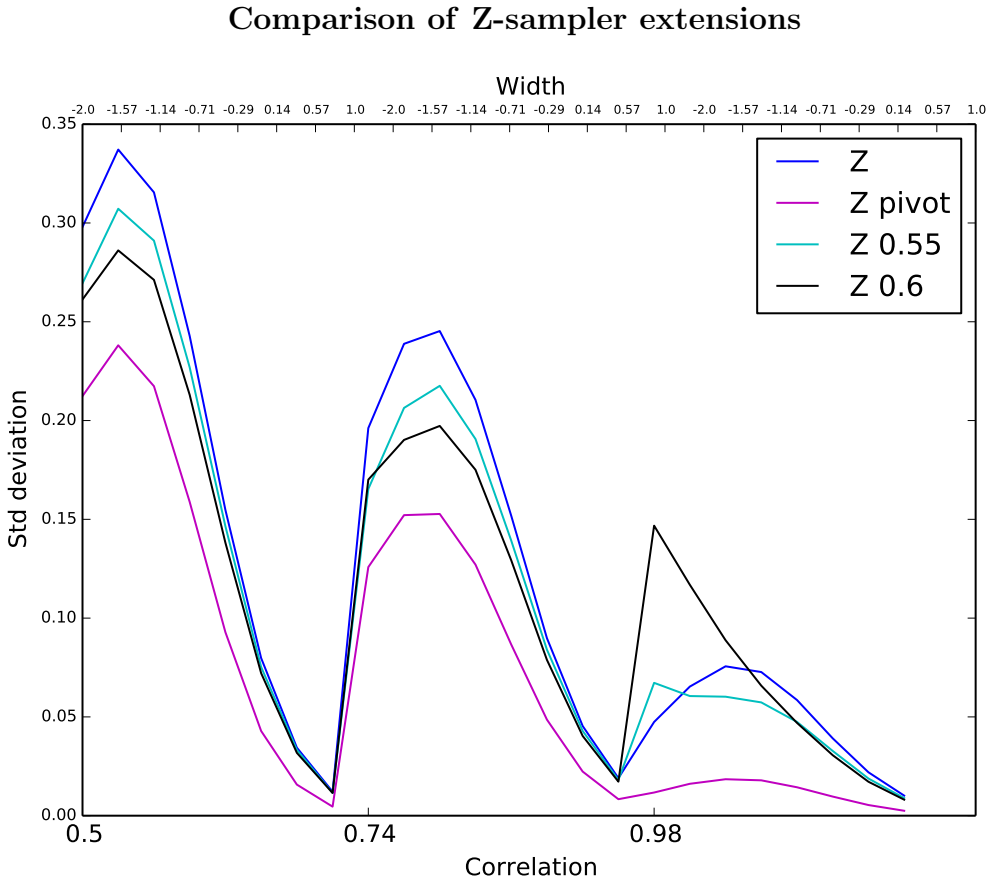

Figure 4.5: Comparison of the original Z-SAMPLER, its pivot extension and its sign correction extension with mixture weights 0.55 and 0.6. The constraints are of the form $S = [a + v, +\infty)$, where $v = (-1, -1 + \frac{2}{n-1}, \ldots, 1)$. For the pivot extension, $S = [a - v, +\infty)$ instead. Parameters varied are a and ρ.

Chapter 5

Numerical algorithms for the latent variables

This chapter presents algorithms that make use of the latent variable structure of the (K, ρ, n)-dimensional order model to reduce the computational complexity of two computations commonly required in the applications. The first is computing the orders from the latent variables (as an adjacency matrix), to determine the partial order induced by the latent variables. The second is computing the maximal set of a partial order, for example when counting the number of linear extensions (this is required for the recursion Alg.:3).

The algorithms presented exploit the fact that the K latent variables are correlated. The numerical results presented here on toy examples show encouraging results when the number of elements n is large enough. In the applications discussed in this thesis, n is not as large and these more evolved algorithms are less useful, nevertheless the reader may be interested to learn ways to make use of the latent variable structure in computations if only for future applications to larger scale problems.

The motivation for this chapter on numerical algorithms comes from the high running time that was required to perform inference in the applications. Particle MCMC was implemented in C++ with the GNU Scientific Library of Galassi et al. for manipulating arrays and generating random numbers. Profiling the simulations (measurement of total CPU time spent on each function) revealed the following top three most costly functions: generating the adjacency matrix M of the partial order $P(Z)$ given a realization of the latent variables Z, counting linear extensions and simulating multivariate normal random variables. This chapter aims to reduce the cost of the first two functions with algorithms making use of the latent variables structure of the partial order model implemented.

5.1 Quicksort like adjacency matrix computation

In the following, a partial order P is represented by its adjacency matrix M where $M_{i,j} = 1$ if and only if $j \prec i$. Given a realization of the latent variables Z, the order $j \prec i$ occurs if $Z_{j,1:K} < Z_{i,1:K}$. The purpose of this section is to efficiently compute M given Z. A simple procedure is to determine whether $Z_{j,1:K} < Z_{i,1:K}$ for all $n(n-1)/2$ element pairs. This would take $\mathcal{O}(Kn^2)$ and will be referred to as StdClosure(Z). It returns the adjacency matrix M of the transitive closure of $P(Z)$ (all orders implied by transitivity are included in M).

The following algorithm is motivated by the following. When $K = 1$, computing M is equivalent to sorting on the real line. In this case, Quicksort may be preferred to StdClosure(Z) as it has an expected number of comparisons in $\mathcal{O}(n \log(n))$ rather than $\mathcal{O}(n^2)$. If $K > 1$, then the standard Quicksort algorithm does not apply. However, if latent variables are correlated such as in the (K, ρ, n)-dimensional order model, then the intuition is that computing $P(Z)$ might be just as easy as sorting on the real line: there are now K lines, but they are correlated. The algorithm presented here is a Quicksort like recursion.

In standard Quicksort, each recursion takes a set as input and splits it into two smaller and disjoint ones. The algorithm presented here is a similar recursion, but the input set is not guaranteed to split into disjoint sets. This suggests a higher running time as a function of n than with Quickosort, which is expected as the array Z to be sorted is multi-dimensional rather than a line. However, sets are more likely to be disjoint if the latent variables are correlated.

5.1.1 Presentation of the algorithm

A pseudo-code of the algorithm is Algorithm 13. It is a recursion with Z, M and S as input. Input Z is the latent variables matrix (for reading only) and M is the adjacency matrix to be filled in. Input S is a set of elements between which some edges are to be determined, and it is split into two smaller but not necessarily disjoint sets.

An element called "pivot" $p \in S$ is picked and the latent variables Z are used to determine the order between p and the other elements $x \in S \setminus p$. This determines the edge matrix entries $M_{x,p}$ and $M_{p,x}$ for $x \in S \setminus p$. This also partitions S into four sets: the pivot p, the sets S_T and S_B of elements with order high and lower than that of p respectively, and the set S_U of elements unordered with p. The procedures then

calls itself recursively twice with S replaced by $S_T \cup S_U$ and $S_B \cup S_U$. The pivot p is discarded from future order computations.

The following propositions show that the matrix M returned (when the recursion ceases to run) is an adjacency matrix of the partial order $P(Z)$ and furthermore that a simple extension of the algorithm ensures that M is the (unique) adjacency matrix of the transitive closure of $P(Z)$. Numerically it is easy to check such claims by comparing the adjacency matrix M returned with the output of the simpler algorithm StdClosure(Z), which returns the transitive closure matrix.

Algorithm 13 Quicksort like computation of $P(Z)$

procedure QuicksortPZ(Z, M, S) ▷ S is a set of active indexes
  if $|S| = 1$ **then**
    Return
  end if
  Pick a pivot p in S ▷ Like in Quicksort
  Compute the sets S_T, S_B, S_U of elements x in $S \setminus \{p\}$ such that $p \prec x$, $x \prec p$ or $x \parallel p$ respectively.
  For $x \in S_T$, set $M_{x,p} = 1$
  For $x \in S_B$, set $M_{p,x} = 1$
  Call QuicksortPZ$(Z, M, S_T \cup S_U)$ and QuicksortPZ$(Z, M, S_B \cup S_U)$
end procedure

Proposition 6 (Validity of computing $P(Z)$ with Algorithm 13)**.** *With input $M = 0$, Z (the latent variables of interest) and $S = \{1, \ldots, n\}$, the algorithm returns an adjacency matrix M (not necessarily transitively closed) of $P(Z)$.*

Proof. By induction on $n = |S|$. If $|S| = n + 1$, then both $S_T \cup S_U$ and $S_B \cup S_U$ have cardinal $\leq n$ because one element (the pivot) is removed from the S. By induction, the algorithm returns in finite time with $M[S_T \cup S_U]$ and $M[S_B \cup S_U]$ as adjacency matrices of the corresponding sub-orders of P.

By edges added to M, the pivot is ordered with respect to all other edges. By transitivity, S_B is ordered with respect to S_T. By induction, each of $S_T \cup S_U$ and $S_B \cup S_U$ are ordered. Thus S_U, S_B, S_T and $\{p\}$ are mutually ordered and so is S. □

Proposition 7 (Computing the transitive closure based on Algorithm 13)**.** *The following extension is considered: having set $M_{x,p} = 1$ for $x \in S_T$ and $M_{p,x} = 1$ for $x \in S_B$, in addition the algorithm sets $M_{y,z} = 1$ for all $y \in S_T$ and $z \in S_B$. Such extension returns an adjacency matrix M which is the transitive closure of $P(Z)$.*

Proof. This follows from the fact that any two elements i and j with $j \prec i$ in $P(Z)$ cannot be separated into some disjoint sets $S_T \cup S_U$ and $S_B \cup S_U$ without setting $M_{j,i} = 1$, whatever the order of the pivot p with i and j. □

It remains to specify how to pick a pivot $p \in S$, if not arbitrarily. If p corresponds to a latent variables $Z_{p,1:K}$ unordered with that of every other element in S, then $S_T = S_B = \emptyset$ and the subsequent function calls have identical inputs. One of them would be superfluous. (The code implemented in the numerical application has a test to avoid calling both when this happens.) A better situation is when p is a more ordered variable, then it may split the set into two smaller sets. This suggests to pick a pivot in S_T or S_B, rather than in S_U, at the next function call. The code implemented uses an arbitrary element of S_T or S_B (the first referenced in memory) as pivot.

The algorithm essentially reduces to Quicksort when all sets S_U are empty, in which case the expect number of comparisons is $\mathcal{O}(n \log(n))$ for a fixed K. In the worst case the unordered sets are large, $S_U = S$, and the algorithm reduces to StdClosure(Z) (and in practice may have higher running time due to the recursion overhead). The algorithm is therefore expected to be more useful when latent variables are more correlated.

As a remark, with a minor modification the algorithm can also return the adjacency matrix of the transitive reduction of $P(Z)$. This can be done by maintaining the transitive reduction as orders are added to M, which can be achieved for an extra $\mathcal{O}(n)$ at each level of the recursion. This leaves the complexity unchanged as $\mathcal{O}(n \log(n))$ in the case of one dimension ($K = 1$) or unit latent variables correlation. This is to be compared to a complexity of $\mathcal{O}(n^3)$ for a naive algorithm looping through all elements pairs for potential order insertions.

5.1.2 Numerical application

As application, random latent variables Z are sampled with correlation ρ and the adjacency matrices M of the transitive closure of $P(Z)$ are computed. StdClosure(Z) is used as benchmark and as a test for correctness (the output should be same). The code is written in C++ with some basic optimization for speed. For example it avoids unnecessary allocation (and de-allocation) of new memory space to hold S_T, S_B and S_U at each step of the recursion by passing a suitable address of a pre-allocated memory space to each new function call so that concurrent calls use non-overlapping subspaces.

The following experiments have three parameters: the number of elements n in the partial order (represented by a transitively closed directed graph, stored in an n by n adjacency matrix), the number of latent variables K for each n elements and the correlation of latent variables ρ. One would expect the Quicksort like algorithm to outperform StdClosure(Z) when ρ is high enough with n and K are fixed.

For small partial orders, such as when $n = 10$ and $K = 10$, it takes microseconds to compute the transitive closure in both cases and Algorithm 13 is faster when the correlation is greater than $\rho \approx 0.97$ (when approximately 25% of element pairs are unordered).

For $n = 100$ and $K = 100$, Algorithm 13 is faster for values greater than $\rho \approx 0.993$ (approx. 15% of pairs are unordered). For $n = 1000$ and $K = 500$, it can take a second to compute the transitive closure and the algorithm is faster from $\rho \approx 0.9996$ (approx. 5% of pairs are unordered).

5.2 Maximal set computation

Given the latent variables matrix Z, the aim is to compute the maximal set $\mathcal{M}(P)$ of $P(Z)$. A motivation for computing the maximal set efficiently is that counting linear extensions with Alg.:3 requires it frequently (at every step of the recursion).

A simple way to proceed is to first compute the adjacency matrix M and then, for each element, test for the existence of any ingoing edges by reading M. Such an algorithm will be referred to as StdMaximalSet(M). The algorithm presented aims to reduce its computational complexity by exploiting the correlated latent variable structure. The idea is to focus computations on elements with high latent variable values in Z, which are more likely to belong to the maximal set.

More precisely, in recursive steps the algorithm shrinks a set S of candidates, an overset of the maximal set, while adding maximal set elements discovered to the solution set. It stops when there are no candidates in S left, at which point all elements of the maximal set have been found.

The key is to shrink the S candidates fast to avoid unnecessarily inspecting elements which do not belong to the maximal set. This is done by focussing on elements which have high latent variable values: the matrix Z is first sorted and auxiliary variable denoted by $s(Z)$ stores the element ranks for each of the K latent variable features. At the first recursion step, the algorithm reads the top ranked elements off $s(Z)$ to determine the top ranked elements, T_1, and also elements unordered with which they are unordered S'. The maximal set is the union of T_1 and of the maximal

set of the partial order on S' (if not empty), hence the subsequent recursion takes $S = S'$ as candidate set. Concatenating $T_1 \cup T_2 \cup \ldots$, the maximal set is determined step by step, hopefully before reading through all of $s(Z)$. If latent variables are correlated, one would expect T_1 to already contain a large fraction of the maximal set.

5.2.1 Presentation of the algorithm

Given a matrix of latent variables $Z \in \mathcal{M}_{n,K}(\mathbb{R})$, the k-th column $Z_{1:n,k}$ can be sorted into a list of elements in ranked order. Such order is one linear extension of $P(Z)$. Intersecting the K columns orders yields the partial order P. Figure 5.1 illustrates the relation between the Z matrix, the sorted column orders and the implied partial order. Although it is not required to explicitly sort Z in order to determine the adjacency matrix M of $P(Z)$, the algorithm will make use of the sorted version, denoted by $s(Z)$. For example, for $n = 3$ if the first column of Z is $Z_{1:3,1} = (2.3, -1.1, 4.0)^T$ then $s(Z)_{1:3,1} = (3, 1, 2)^T$.

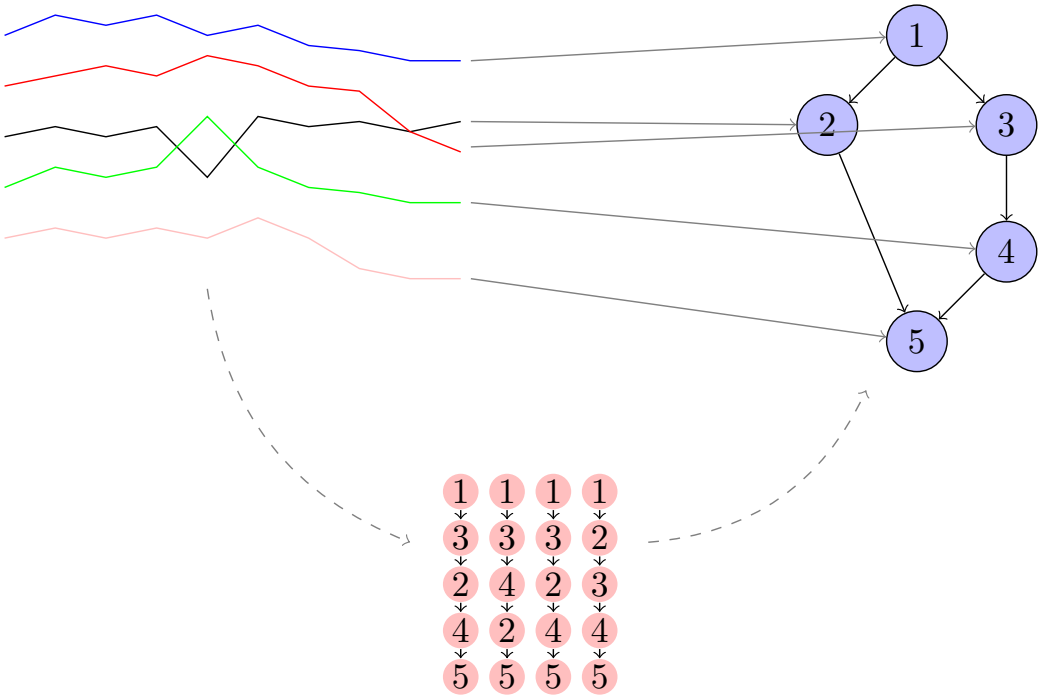

Figure 5.1: The Z matrix (left), the sorted $s(Z)$ matrix (below) and the implied partial order (right).

Definition 2 (Auxiliary variable $s(Z)$)**.** *For a latent variable matrix $Z \in \mathcal{M}_{n,K}$, $s(Z) \in \mathcal{M}_{n,K}$ is the matrix where column k is the list of the n elements appearing in*

descending order from sorting according to the features values of the k latent variable feature, $Z_{1:n,k}$.

Computing $s(Z)$ can always be achieved in $\mathcal{O}(Kn\log(n))$ by sorting the K columns independently. An alternative not pursued here is to use the dependence across the columns to reduce the expected complexity. One may suspect that a simple modification of the Quicksort like algorithm Algorithm 13 would be useful to achieve this: once S_T, S_B and S_U are determined, only the orders of elements in S_U with respect to the pivot p may differ across columns.

The purpose of $s(Z)$ is to enable the fast discovery of the maximal set elements. For example, reading the K values of the first row yields a subset $T_1 = s(Z)_{1,1:K}$ (of cardinal $\leq K$) of the maximal set T. As latent variables correlation increases, so does the probability of $T_1 = T$. Algorithm 14 is a recursive computation of $\mathcal{M}(P)$ such that the successive outputs increase in size: T_1, $T_1 \cup T_2$, $T_1 \cup T_2 \cup \ldots$, until $T_1 \cup \ldots \cup T_k = T$ for some k. The input sets S are oversets of the maximal set. The set S' is the set of all elements of S not in T_i which are unordered with all elements of T_i. As the recursion continues, S shrinks in size and the remaining number of candidates to inspect decreases, until there are no more candidates.

Algorithm 14 explicitly requires $s(Z)$ but not the adjacency matrix M as input: computing the orders of all elements pairs may not be required to determine the maximal set. However, if M has already been computed, then it can be used to determine the set S' and avoid re-computing orders from the latent variables Z. Since M can take $\mathcal{O}(Kn^2)$ to compute (or less in expectation if one implements the Quicksort like algorithm Algorithm 13), then computing $s(Z)$ only instead may be advantageous.

Algorithm 15 is a simpler and non-recursive algorithm displayed to clarify the main ideas of Algorithm 14. It computes a subset T_1 of the maximal set before using the adjacency matrix M (only its restriction $M[S]$ to S needs to be inspected) to compute the rest of the maximal set. This captures some of the benefits of Algorithm 14: as remarked, when latent variables are highly correlated it is expected that T_1 contains a large fraction of the maximal set, leaving only a small matrix $M[S]$ to inspect for determining the remainder.

Proposition 8 (Validity of computing $\mathcal{M}(P)$ with Algorithm 14)**.** *With input* $s(Z)$, $S = \{1, \ldots, n\}$ *and* $i = 1$, *the algorithm returns the maximal set* $\mathcal{M}(P)$ *of* $P(Z)$.

Proof. By abuse of notation, the code uses the $\cup$ and $\cap$ set notations to represent joining and intersecting arrays even though the arrays are not sets. The algorithm

Algorithm 14 Computing the set of maximal set $\mathcal{M}(P)$ given $s(Z)$

procedure MAXIMALSET($s(Z)$, S, i)
  Compute $T_i = s(Z)_{i,1:K} \cap S$
  if $T_i = \emptyset$ **then**
    Return MAXIMALSET$(s(Z), S, i+1)$
  end if
  Compute $S' = \{x \in S \setminus T_i \mid x \parallel y, \forall y \in T_i\}$
  if $|S'| \leq 1$ **then**
    Return $T_i \cup S'$
  end if
  Return $T_i \cup$ MAXIMALSET$(s(Z), S', i+1)$
end procedure

Algorithm 15 A simple alternative to Algorithm 14

procedure MAXIMALSET($s(Z)$, M)
  Compute $T_1 = s(Z)_{1,1:K}$
  if $|T_1| = 1$ **then**
    Return T_1
  end if
  Compute $S = \{x \in \{1, \ldots, n\} \setminus T_1 \mid x \parallel y, \forall y \in T_1\}$
  if $|S| \leq 1$ **then**
    Return $T_1 \cup S$
  end if
  Return $T_1 \cup$ STDMAXIMALSET$(M[S])$
end procedure

is a recursion reading rows of $s(Z)$ step by step until all maximal set elements are found. An element a does not appear in the output if and only if it is removed as a candidate: at some step i, it belongs to S but is dropped from S'. Equivalently, at step i, a is ordered with at least one element $b \in T_i$. Since a is not in the output, a first appears in a row j of $s(Z)$ such that $j > i$. Therefore a is dropped if and only if $a \prec b$ for some element b.

□

The running time depends on the implementation. The code Algorithm 14 is implemented with some basic optimisation for speed. For example some auxiliary arrays are pre-allocated in memory for computing set intersections and unions and for saving orders already computed in previous steps. With such set-up, the complexity is determined in large part by the number of orders to be computed (once). To determine these orders, the approximate number of comparisons required is $Kn^2/2$ for StdMaximalSet and $t(n-t)K + s^2K/2$ for algorithm Algorithm 15 (the non-recursive version is easier to analyse), where $t = |T_1|$ and $s = |S| \leq n - t$. In the worst case of $t = n/2$, Algorithm 15 is still favourable, even after accounting for the $\mathcal{O}(Kn\log(n))$ comparisons in computing $s(Z)$. However, the algorithm is intended to benefit from the latent variables correlation where t and s may be much smaller. In the best case of $\rho = 1$, $t = 1$ and $s = 0$ and the comparison cost is essentially that of computing $s(Z)$.

5.2.2 Numerical application

As numerical application, random latent variables matrices Z are sampled and for each the maximal set of $P(Z)$ is computed with StdMaximalSet and the recursion Algorithm 14. The average running time to compute the maximal set is compared. The running time includes computing M for StdMaximalSet and $s(Z)$ for the recursion. For $K = 10$ and $n = 10$, both algorithms have a similar running time of approx 10^{-5} but the recursion is slower when ρ is small. With $n = 100$, both have similar running time of 10^{-4} at small ρ, but the recursion is three times faster when $\rho > 0.6$. When $n = 1000$, the recursion is faster by an order of magnitude when $\rho > 0.6$. When K is increased, the conclusion remains: the recursion is clearly faster at high ρ. Similar experiments without M and $s(Z)$ as given (pre-computation not taken not accounted for in the running time) shows that the recursion is slower.

As a conclusion, the recursion Algorithm 14 may have lower running time (depending on n, K and ρ) than the simpler algorithm StdMaximalSet as it avoids

the full computation of the adjacency matrix M. The benefits appear clearly when n and ρ are large enough for fixed K (in the simulations considered, this is when $n > 100$, $\rho > 0.6$ and $K = 10$).

Chapter 6

Applications

This chapter turns to the applications of the partial order model to time series of rankings. Since a ranking can be interpreted as a total order, the partial order model can be used as a ranking model, an alternative to one-dimensional models such as Plackett-Luce. There are at least two situations where it is natural to model rankings as linear extensions from a partial order. One is when not all elements are comparable for conceptual reasons. For example, in the bishop dataset analysed in this chapter, it is known from historical accounts that social status is influenced by several factors (geography, network, wealth, etc). For any two bishops, the order seems undetermined unless one of them ranks higher on all these factors. Such social hierarchy therefore seems amenable to modelling with the latent variables Z. The application of statistics to understand the bishop data presented here is novel.

Another application of interest for partial order models is statistical averaging: the average partial order smooths the times series of total orders. Two elements with inconsistent order have no order in the average partial order: the orders are *averaged out*. The partial order removes the more uncertain orders. As in standard time series smoothing, this is a function of time: relative to a given time frequency of interest, some orders appear stable and others unstable. However, the partial order model accomplishes more than merely smoothing as the model may *learn* orders between elements even if they never appeared together in any observation.

Pairwise comparison data (two players per match) is a popular applications of ranking models. Although the partial order model and inference algorithm developed in this thesis would apply to pairwise rank data as a special case (the likelihood becomes trivial), a more natural application of the model is multi-competitor data where orders between more than two elements appear in the observations. Some recent work based on the Plackett-Luce model is applied to data with both a time series structure and observations with orders between more than two elements, for

example in sport competition rankings with skiing in Glickman [2015] and golf in Baker and Mchale [2014].

6.1 Bayesian inference for partial orders

This section gives Bayesian methodology for inference on the Z_t process of Chapter 2, to be applied to the datasets in the rest of this chapter. It is sampled based inference made possible with the posterior samples obtained from running the Particle MCMC algorithm of Chapter 3. For any fixed t, the posterior samples allow to estimate the posterior probability of some order, e.g. of $\{a_1 \prec a_2\}$ or $\{a_1 \prec a_2 \prec a_3 \prec a_4\}$, the probability of a particular element a_i ranking first, or the mean partial order depth and with standard deviations. The following sections show how to summarize the posterior partial order samples and perform some Bayesian model choice and model checking.

6.1.1 Consensus order

Section 2.1.3 showed that the maximum likelihood partial order estimate for uniform linear extensions static model is an intersection of the total orders observation. Another statistic of interest here is the consensus or summary order of a set of partial orders. For a given threshold τ, the consensus order P based on M partial orders $(P^{(m)})_{m=1}^M$ will be defined as the order formed by all orders appearing in proportion of at least τ in these M partial orders. If τ is less than 50%, then P may be cyclic (not a partial order).

In the time series applications explored, some latent variables $(Z_{t_i}^{(m)})_{m=1}^M, i = 1:T$ are sampled from the posterior at (near) equilibrium of Particle MCMC. The set of M of latent variables samples $(Z_{t_i}^{(m)})_{m=1}^M$ corresponds to observation y_i. It may be of greater interest to summarize posterior orders at a fixed and chosen time t rather than at the observation times t_i, which may themselves be modelled as random. This can be achieved at little extra cost by simulating $Z_t^{(m)}$ given $Z_{t_i}^{(m)}$.[1]

[1] This following algorithm shows how. It follows from the discretization of the latent variable process in Equation 2.6. After finding the interval of consecutive times such that $t \in [t_i^{(m)}, t_j^{(m)}]$, set $Z_t^{(m)} := Z_{t_i}^{(m)}$. Inspection of $Z_{t_i}^{(m)}$ and $Z_{t_j}^{(m)}$ determines the list L of element paths (matrix rows) updated at least once in $[t_i^{(m)}, t_j^{(m)}]$. For each $l \in L$, a Poisson random variable n_S with rate $(t_j^{(m)}) - t_i^{(m)})\lambda_S/n$ is sampled conditioned on $n_S \geq 1$. If it is $n_S > 1$, row l in $Z_t^{(m)}$ is re-sampled, otherwise if $n_S = 1$ then it is set to row l of $Z_{t_j}^{(m)}$.

6.1.2 Model choice

The latent variables process undergoes mutations from a Poisson process $\mathcal{P}(\lambda_S)$ and change points from a Poisson process $\mathcal{P}(\lambda_C)$. It may be of interest to test if the change point process is a useful part of the model. The model with and without the change point process is denoted by M_1 and M_0 respectively. The Bayes factor is $p(y|M_0)/p(y|M_1) = p(M_0|y)p(M_1)/p(M_1|y)p(M_0)$, where $p(M_i)$ are prior probabilities and $p(M_i|y)$ are likelihoods.

The likelihoods can be estimated with the Particle MCMC output by counting the number of samples with and without change points. By the law of total probability, the prior probability of no change points is $p(M_0) = \Psi_C/(\Delta T + \Psi_C)$, where Ψ_C is the parameter for the prior $\lambda_C \sim \mathcal{E}(\Psi_C)$ and ΔT is a timespan of interest. In the following, $\Delta T := t_{(T)} - t_{(1)}$ is the timespan of the observations (ordered as $t_{(1)} < \ldots < t_{(T)}$) if their times are fixed. If as in some applications the times are modelled as random, it will be defined as the maximum timespan which has prior support.

6.1.3 Model tests

This section turns to model checking via posterior predictive tests on synthetic data shown on Figure 6.1. The first test is symmetric with $n = 6$ elements and 10 identical observations equal to $y = \{6 \prec 5 \ldots \prec 1\}$. Observation times are unknown. The prior on i-th observation time is a uniform distribution on $[i, i+1[$. There are $K = n$ latent variables with path correlation distribution as $\rho \sim B(1, 1/6)$. The prior on rates $\lambda_S \sim \mathcal{E}(1)$ and $\lambda_C \sim \mathcal{E}(10)$ is such that on average there is one mutation per one time unit and one change point per ten time units.

Inspection of the consensus partial orders with threshold $\tau > 0.01$ at a few time points in $[1, 11]$ shows orders to y. Furthermore, the posterior on change point times (change point samples conditioned on there being at least one change point) is flat in $[1, 10[$ and a peak in $[10, 11[$. This is expected for identical observations: there should be a lower posterior probability of a change point between the observations. With a Bayes factor of 1.2, the change point model is barely worth mentioning.

The second test uses similar settings but is antisymmetric with an order reversal at the midpoint: observation y is repeated in the first half of the interval, then the reverse (transpose) of y is repeated in the second half. Near the midpoint the consensus order with threshold $\tau > 80\%$ is flat and the posterior on change points has a mode. The consensus order is y before the midpoint and then reverses. This shows evidence of time series smoothing: posterior samples just after the midpoint depend

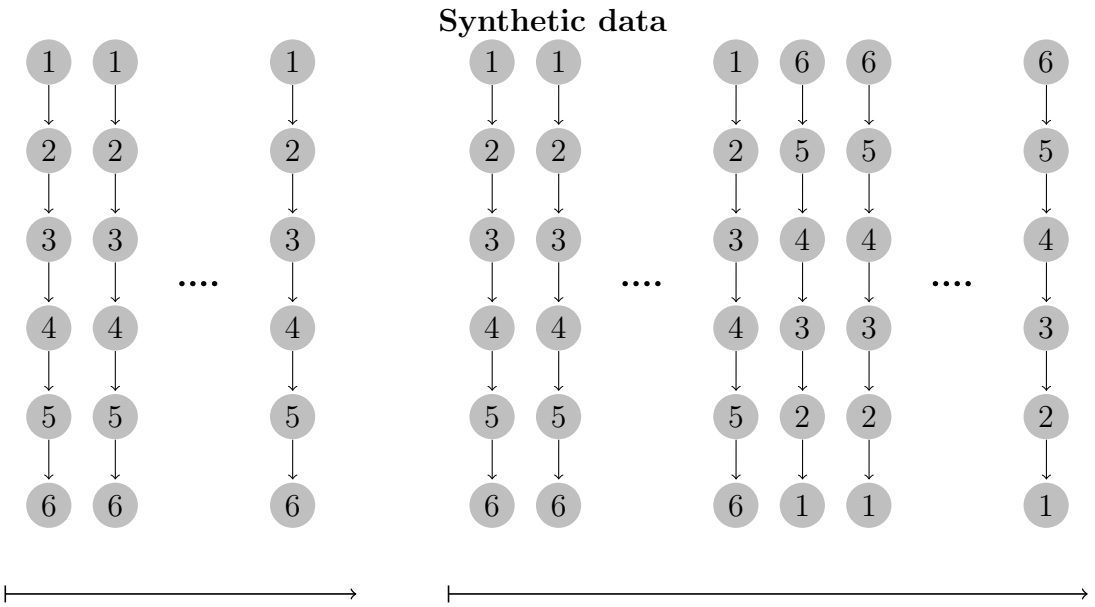

Figure 6.1: Synthetic data for the symmetry (left) and antisymmetry (right) tests.

on all future observations in the second half of the time interval. With a Bayes factor of 0.25, there is substantial evidence in favour of the change point model.

6.2 Bishop hierarchy

6.2.1 Introduction to the data

The data of interest is a set of witness lists of royal charters provided by Dr David Johnson, St. Peter's College, Oxford, and Nicholas Karn, University of Southampton. Witness lists have been used to study medieval politics e.g. in Westervelt [2008], however little statistical work has been performed on them. For each charter, the names of individuals present are recorded in the order of signing, which is a function of their social hierarchy. The number of witnesses in each list is variable and the exact dates are missing, however lower and upper bounds on the calendar year is given. Modelled as a time series of rankings with stochastic times and numbers of elements at each observation, the data lends itself to the partial order model described in this thesis.

The following analysis focusses on the period 1110-1155 corresponding to 327 witness lists as observations involving 67 bishops and other noble titles (a list is included if its date uncertainty range is included in that period). Figure 6.2 shows partial orders summaries at five year intervals. The three partial orders representing 1125-1140 have a lower depth than those before 1125 and after 1140. Does this mean that the

bishop hierarchy was less strict in 1125-1140? The number of observations intersected in each window varies and this could affect the depth artificially. Inspection of the data does show that more observations fall in the three five years windows covering that period, however a new plot (not shown) arbitrarily limiting the number of intersected observations reveals similar partial orders. The time series partial order model might help to determine whether the hierarchy depth really collapsed: there will be no need to assume arbitrarily that the partial order is constant in each of the intervals.

6.2.2 Bishop of London in 1120-30

Inspecting the data for unusual changes in the hierarchy reveals that the bishop of London seems to lose precedence in the 1120-30 decade. The role of the bishop of London in the twelfth century is studied in Johnson [2013], from which the following context is borrowed. By the end of the eleventh century, the bishop of London had risen in status. Him and the bishop of Winchester were then positioned just after the archbishops of York and Canterbury. The origins of their precedence is debated, but both bishops made claims to archbishop status in the twelfth century, and this could have been seen as a threat by some. Perhaps it is no coincidence that the bishops made claims just when the archbishop of Canterbury was on bad terms with the king.

As observations, $T = 46$ orders on $n = 10$ bishops are selected from the 1120-30 decade. Figure 6.4 displays the first and last few observations (in a chronological order determined by the mean of the posterior on times). Table C.1 gives the bishops' names. Latent variables parameters are $K = n$ and $\rho \sim B(1, 1/6)$. This is such that the prior depth on partial orders samples is nearly flat. The rates are distributed $\lambda_S \sim \mathcal{E}(1)$ and $\lambda_C \sim \mathcal{E}(10)$, so that the prior mean is one mutation occurs each year and about one change point per decade.

The Particle MCMC Algorithm 5 is ran with one million samples and the acceptance rate is on average approximately 1%. MCMC trace plots and histograms for rates λ_S and λ_C are given in Appendix B. The Monte Carlo error is noticeable on the histograms, however histograms with different chains all suggest a posterior with mode near 0 for λ_C (with mean less than 0.1) and above 1 for λ_S (with mean greater than 1). This suggests data explain by mutations rather than by change points. The posterior probability of observing any change point in the decade is 22% and Figure 6.4 shows the location of three posterior modes. The Bayes factor is above 4.6, substantial strength of evidence in favour of the model M_0 without change points.

Bishops data

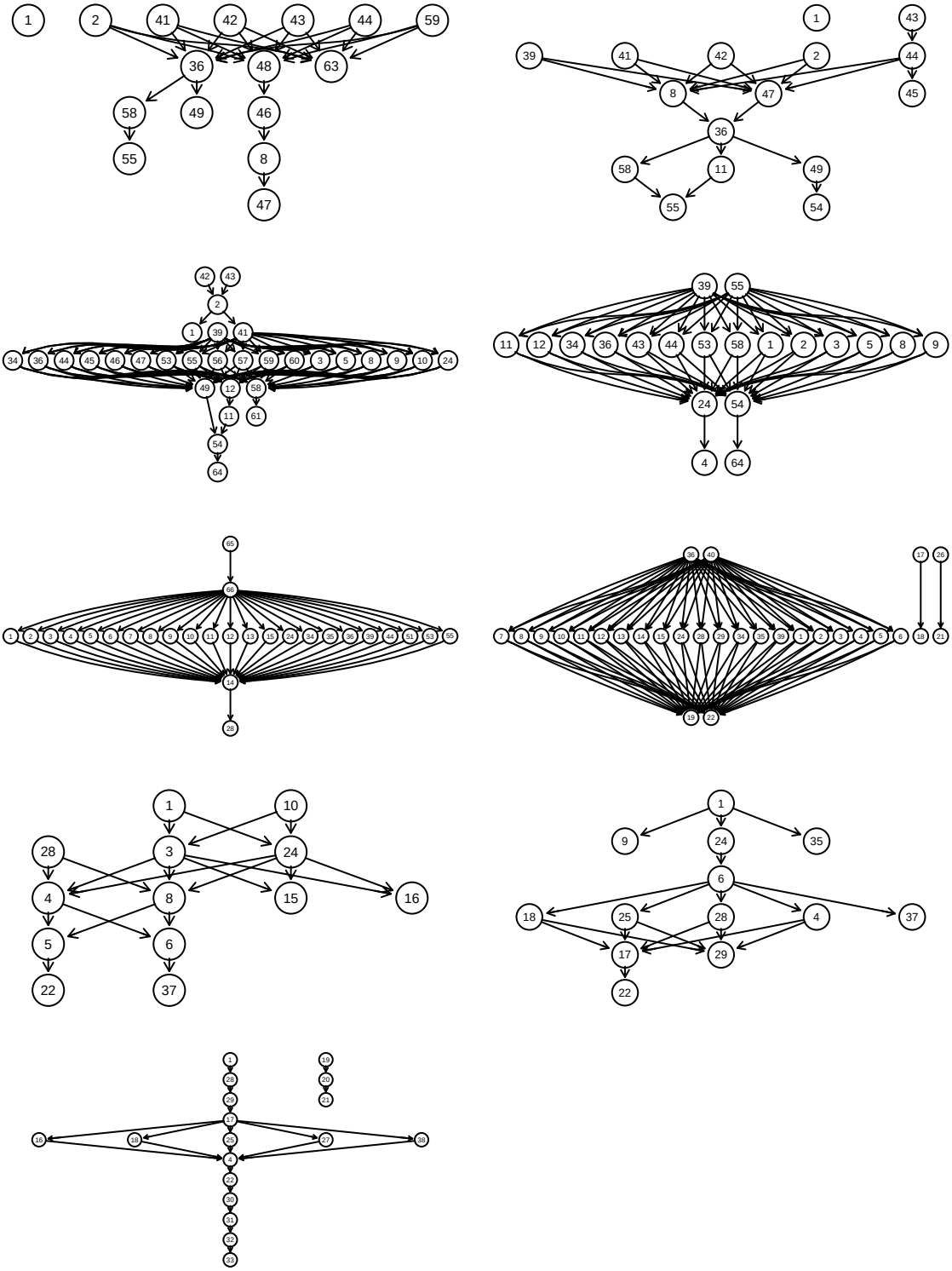

Figure 6.2: Data visualization with a sequence of partial orders in five year rolling windows starting at year 1110. Each order is the intersection of all observations which time is known to be included in the window. Unordered (isolated) elements are removed for clarity.

Figure 6.3 shows the posterior depth distributions at fixed times $t \in \{1123, 1125, 1127\}$. They are more peaked than the prior, with a mode at depth 5. This confirms that the data exhibits some ordering, albeit not a total ordering. This agrees with the consensus orders on Figure 6.4, which have have depths of 4 and 5. In contrast, the MLE has low depth at the end of the decade. Thus, posterior analysis implies that the loss in hierarchy is less dramatic than what was initially suggested by the exploratory data analysis.

Figure 6.4 shows the posterior consensus orders at $t \in \{1123, 1125, 1127\}$, the first and last seven observations (in a chronological order determined by the mean of the posterior on times) and their respective intersections. The consensus orders have no isolated (completely unordered) groups and as one would expect the bishop of Winchester (element 1) has a high precedence throughout. The intersections are less credible, as they show the bishop of Winchester as unordered with a few other bishops. This simply reflects the fact that the dynamic partial order model can learn orders over time even if they are not always observed.

The consensus orders shows that the bishop of London (element 10) loses precedence by $t = 1125$, especially after considering the fact that the consensus orders (at threshold 50%) show an otherwise deep hierarchy (depth 4 and 5). The bishops of Lincoln (element 9) and St David's (element 6) appear to gain in precedence in the intersections. However, the consensus orders suggest a more stable precedence, similar to that of several other bishops. Only the precedence of the bishop of London seems to undergo a major change. This agrees with a model explained by mutations rather than by change points.

6.2.3 Bishop of Salisbury in 1130-40

Another decade with potential upheaval is 1130-40. Roger of Salisbury was an influential bishop until King Stephen seized his castles in 1139, the year of his death.[2] The partial order model will be used to learn the impact of King Stephen's accession in 1135 on the bishops' ordering.

As observations, $T = 36$ orders $n = 9$ bishops are selected from the 1130-40 decade. Table C.2 gives the bishops' names. The prior distributions are the same as for the 1120-30 analysis. The posterior on change points is close to its prior, whereas the posterior on mutation rates has a mode somewhere above 1. At near 50%, the posterior probability of at least one change point is higher. However with a Bayes

[2]From "Roger" in the Encyclopdia Britannica, 11th ed, Vol. XXIII. Encyclopdia Britannica Co. (New York), 1911.

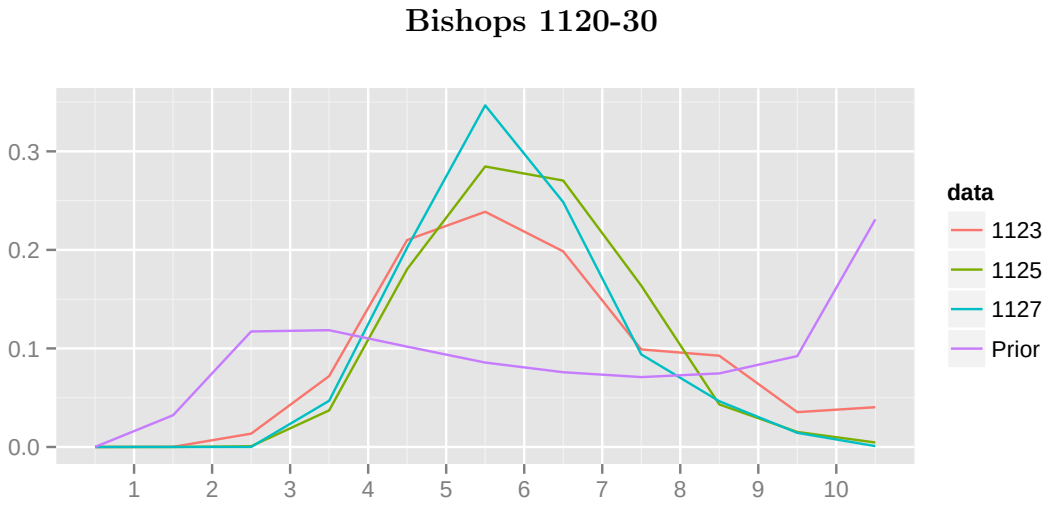

Figure 6.3: Empirical densities on partial order depth for the prior and posterior at times $t \in \{1123, 1125, 1127\}$.

factor close to 1, there is little evidence for (or against) a model with change points. Figure 6.5 shows similar posteriors on depth for the times specified, with a mode at depth 4. The results are thus similar to those for 1120-30, but there is more evidence for change points and mutations.

This agrees with the posterior consensus orders shown on Figure 6.6, which show several bishops changing precedence: the bishop of Ely (element 7) gains precedence over time, while the bishops of St David's and Hereford (elements 4 and 9) lose precedence. The rise of the bishop of Ely could be explained by the fact that the title holder was the nephew of the bishop of Salisbury. The bishop of Salisbury (element 1) does not seem to lose precedence other than when compared to the bishop of Winchester (element 8). To conclude, the data hardly shows forewarning signs of Roger's fall at the end of the decade. His nephew rises in precedence while the bishop of St David's falls, but posterior analysis shows that a change point is barely necessary to explain the evolution.

6.3 Golf tournaments

Baker and Mchale [2014] studied golf tournament scores to infer who is the greatest golfer. The analysis uses an extended Plackett-Luce model with time-varying strengths (latent, one-dimensional variables) that allows for comparison between players of different eras. The following presents a partial order version of the analysis.

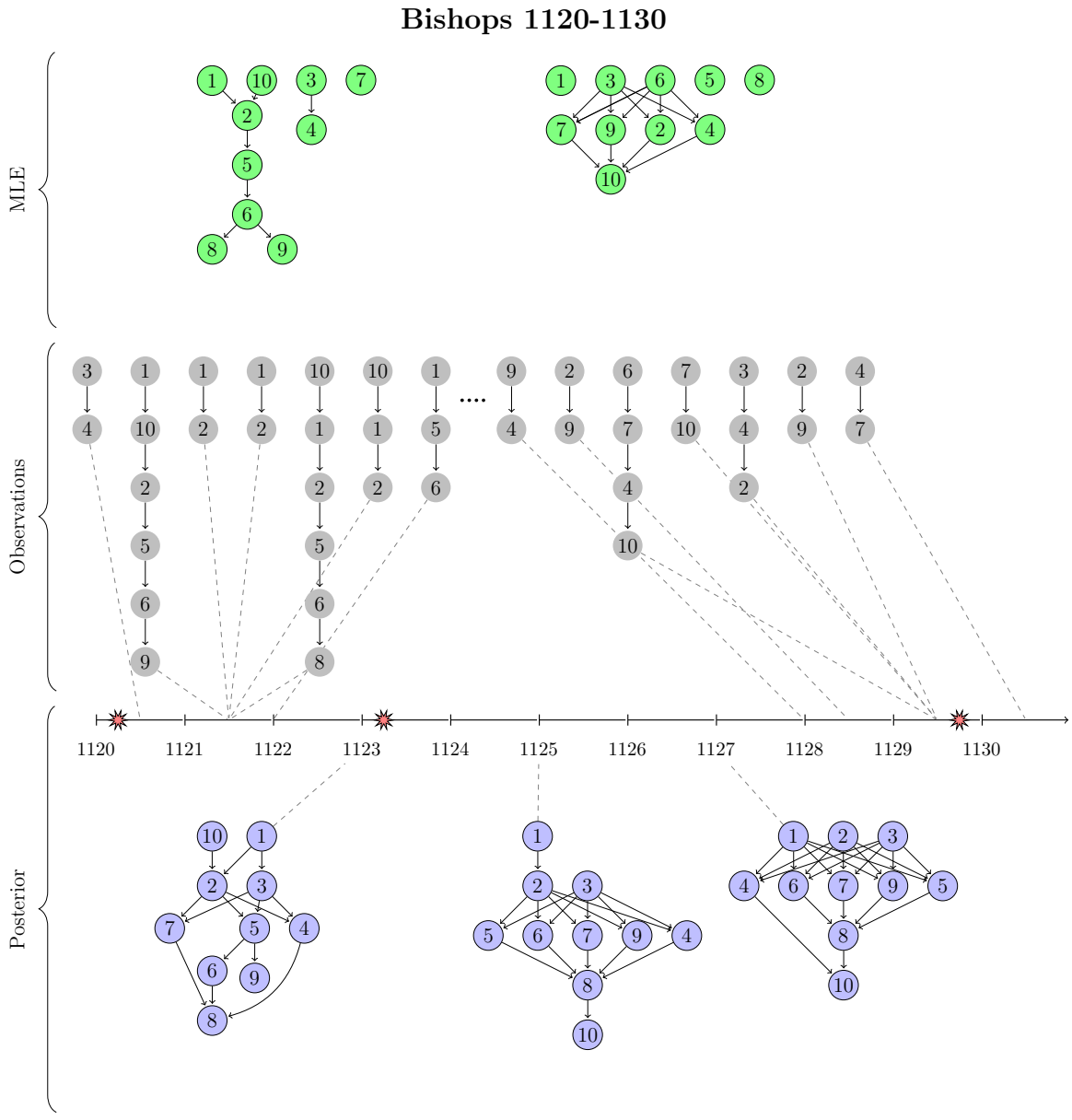

Figure 6.4: Middle: first and last seven linear extensions when ordered by the posterior means of the 46 observation times. Above: MLEs of the first and last seven observations. Below: posterior consensus partial orders at $t \in \{1123, 1125, 1127\}$ with threshold 50%. Each red star is for a change point posterior mode.

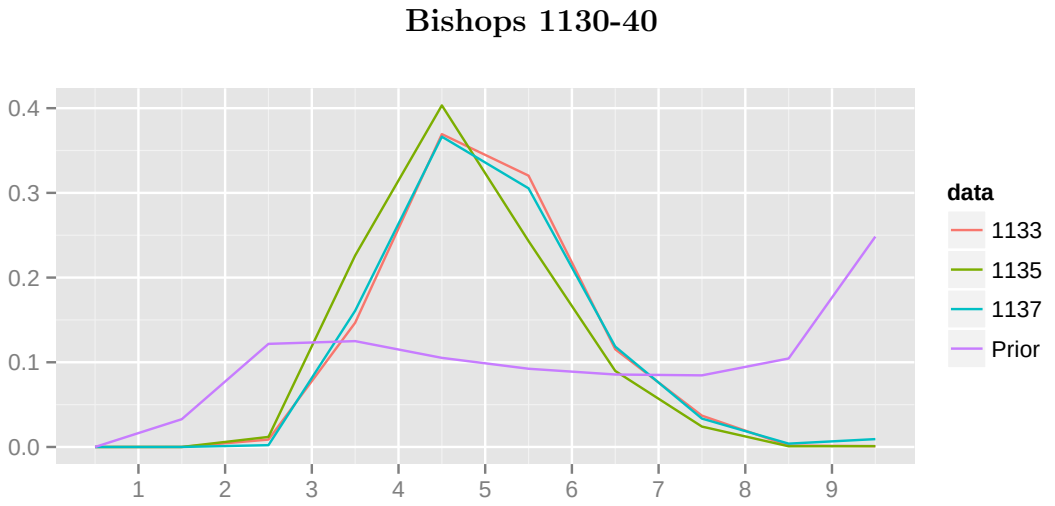

Figure 6.5: Empirical densities on partial order depth for the prior and posterior at times $t \in \{1133, 1135, 1137\}$.

The dataset is a set of scores from the US Masters golf tournament [3]. The following focusses on a set of $n = 10$ players and $T = 33$ tournament years where at least three of these players have participated in the tournament and no ties occurred. Table C.3 gives the players' names. The year range is 1955-2002. The scores plotted on Figure 6.7 show that no player uniformly dominates any other. The top average scores belong to Ben Hogen, Lee Trevino and Sam Snead. The purpose of the following analysis is to smooth the rankings with partial orders to reveal the significant orders. This differs from the bishops data where some orders observed are stable. Here, the average posterior partial order has lower depth and a lower threshold τ is needed to represent a posterior consensus order with the same depth.[4]

As previously, latent variables parameters are $K = n$ and $\rho \sim B(1, 1/6)$, rates are distributed as $\lambda_S \sim \mathcal{E}(1)$ and $\lambda_C \sim \mathcal{E}(10)$. The MCMC plots are found in Appendix B. There posterior probability of a change point is approximately $1/2$ (over the whole year range) and the Bayes factor is 4.9, substantial evidence for a model without change points. The posterior consensus orders at $t \in \{1960, 1980, 2000\}$ have depth distributions with mass mostly on depths 2 and 3 (not shown).

Baker and Mchale [2014] proposed to define the best player as having one of the following properties: maximum strength ever achieved, maximum strength during a 5-

[3] Retrieved from http://www.stat.ufl.edu/ winner/data/mastersy.dat.

[4] In such situation, perhaps more meaningful alternatives to the consensus order can be sought for summarizing the posterior partial orders. For example, the order defined by $a \prec b$ if the posterior probability of $a \prec b$ is more than that of $b \prec a$ by some amount.

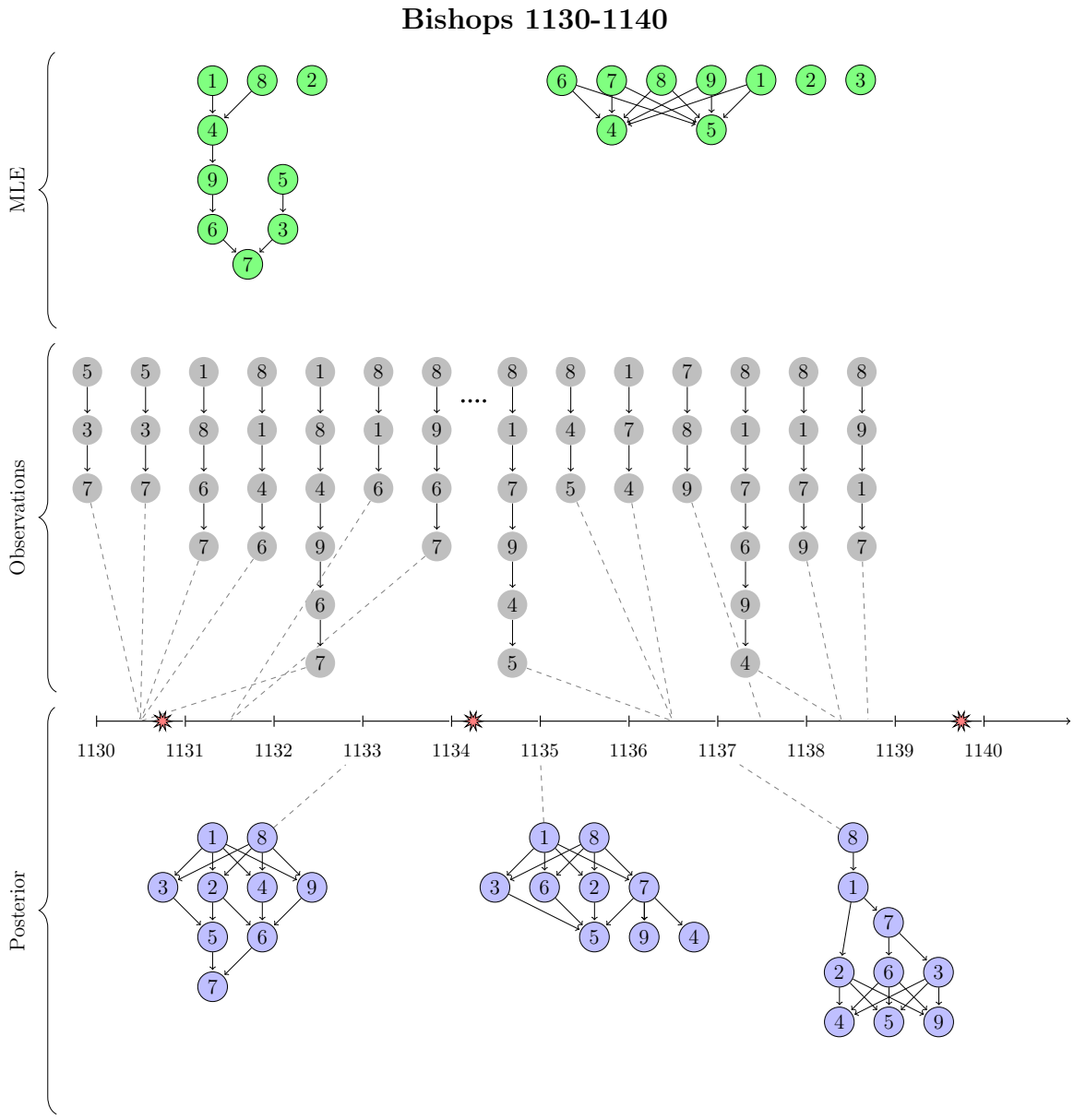

Figure 6.6: Middle: first and last seven linear extensions when ordered by the posterior means of the 36 observation times. Above: MLEs of the first and last seven observations. Below: posterior consensus partial orders at $t \in \{1133, 1135, 1137\}$ with threshold 50%. Each red star is for a change point posterior mode.

year period and total strength. These strengths are one-dimensional latent variables, but it would be possible to define similar metrics for the Z_t process, where the strength would correspond to the latent variables of a fixed feature $k \in \{1, \dots, K\}$ or to the average over the K features. An alternative making more use of partial order structure is to look at players in the maximal set (the top elements) of the posterior partial orders especially if they have higher depth, i.e. top players in the years when the hierarchy is deep. Figure 6.8 shows the consensus partial order with a low threshold of $\tau = 10\%$. They allow to compare golfers of different eras, for example the mid-century golfer Ben Hogan (element 2) with the contemporary Tiger Woods (element 5). With a threshold of 20%, the consensus orders are almost flat but Sam Snead (element 4) appears as the only player with at least one winning order on all consensus orders. This is coherent with the fact that he obtained the highest average score in this dataset. Furthermore, Figure 6.8 shows that Ben Hogan is at the top of the posterior consensus order in 1960, Sam Snead, Lee Trevino and Arnold Palmer at the top in 1980, and Gary Player at the top in 2000.

In comparison, Baker and Mchale [2014] concludes that the overall top player is Ben Hogan, and that the top era specific players were Sam Snead, Ben Hogan, Arnold Palmer, Jack Nicklaus and Tiger Woods. Therefore, the conclusions disagree for Jack Niclaus and Tiger Woods: they are at lower ranks of the consensus order at all three times shown (consistent with the scores in the raw data 6.7). The difference can be explained by the data used by Baker and Mchale [2014], which includes more than US Masters tournament results and with proportionally more titles won by Nicklaus and Hogan.

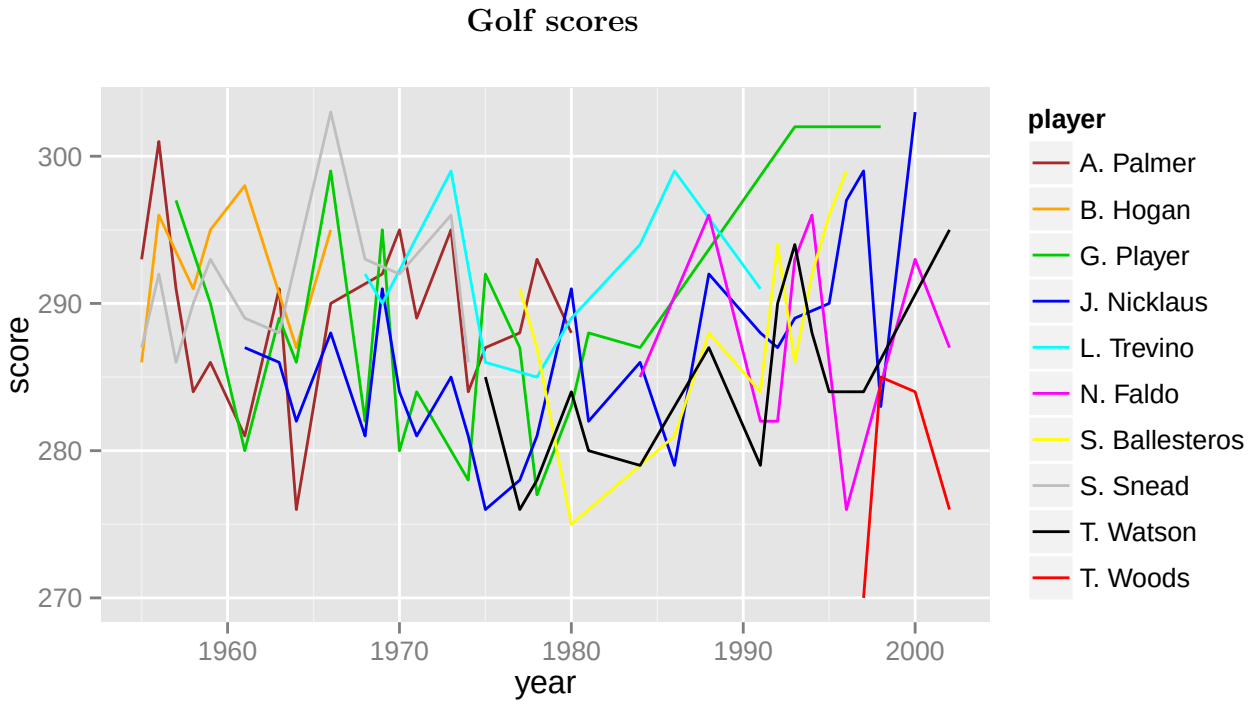

Figure 6.7: US Masters golf tournaments scores for 10 players.

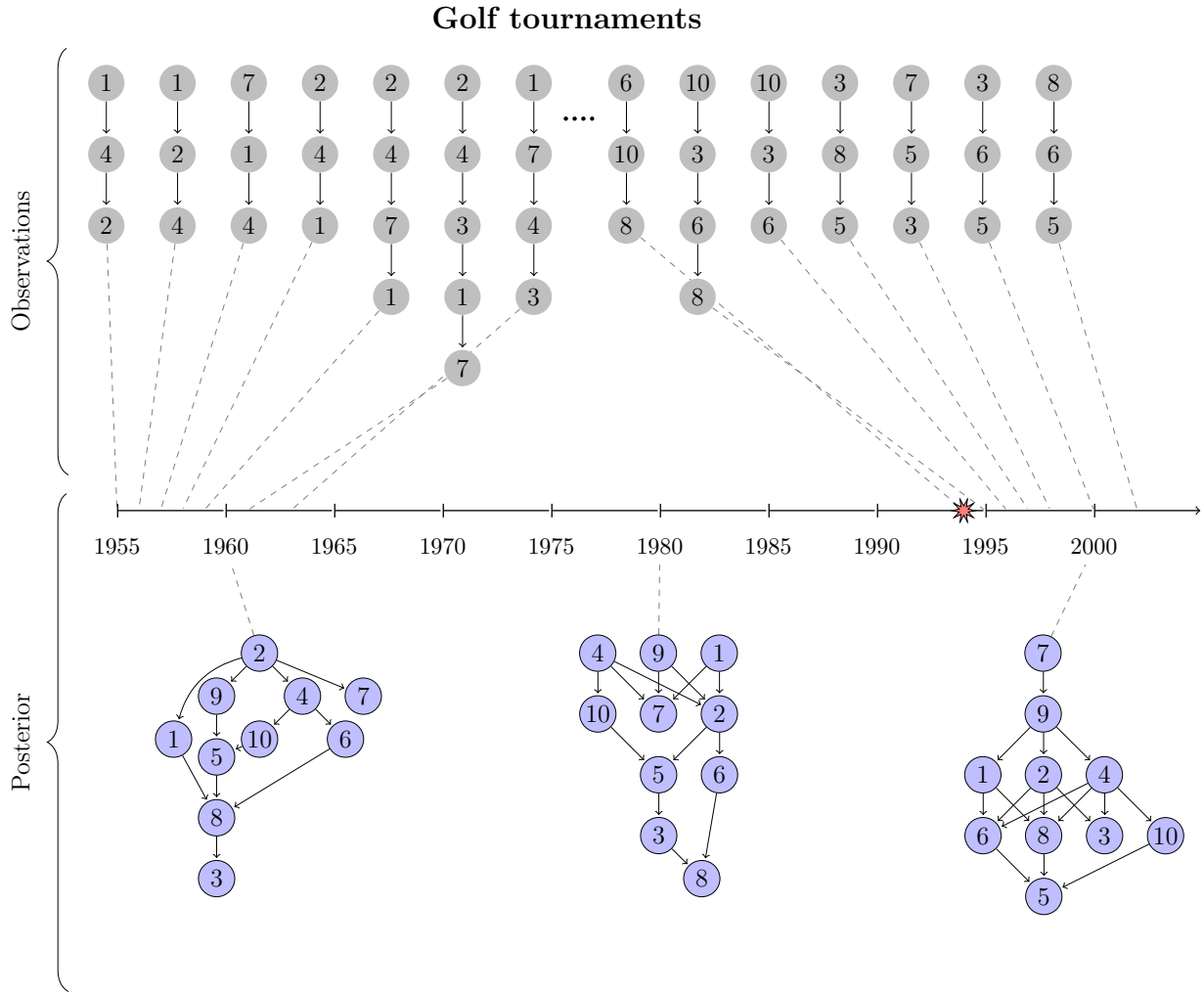

Figure 6.8: Above: first and last seven linear extensions of the 33 observations. Below: posterior consensus partial orders at $t \in \{1960, 1980, 2000\}$ with threshold 10%. The red star is for a change point posterior mode.

Chapter 7

Conclusion

Chapter 1 surveyed three main partial order models: the uniform on partial orders, the $G_{n,p}$-graph order model and the (K, ρ, n)-dimensional order model. The former two have a restrictive structure while the (K, ρ, n)-dimensional order model is general and has a parameter for controlling the mean partial order depth and is therefore our model of choice.

In Chapter 2, the model is extended to a continuous time model on latent variables Z_t. In applications, we would like to use as ranking model a Hidden Markov model where observed orders are modelled as random linear extensions of suborders of an unobserved partial order process at a sequence of uncertain sampling times. Since the Z_t model is tractable and can be discretized at the observation sampling times, it is convenient as a model for the hidden state. The aim is performing inference on the latent variables and other parameters such as the mutation and change point rates. The model parameters are the static parameters $\theta := (\Phi, \lambda_C, \lambda_S, (t_i)_{i=1}^T)$ and the hidden states of the HMM, $X_i := (Z_{t_i}, \rho_{t_i})$.

Performing such inference is intractable, but Particle MCMC applies. The dimension of the hidden state is high and the likelihood is a hard obstacle model, which results in a slow mixing time. We therefore suggest some particle filter proposals that condition on the observations y. It is possible to sample such that the hard obstacle are avoided with the samplers provided, but at a significant computational cost. However, it is possible to sample latent variables with little extra cost with the Z-sampler. If the right latent variables to renew are selected, then the Z-sampler can renew them with same support as the target (and is asymptotically optimal when the correlation is 1).

Chapter 4 generalizes the Z-sampler, as it turns out that sampling the latent variables with constraints is related to the orthant probability estimation problem. We show that our importance sampler is competitive in the case where correlation is high.

Chapter 5 suggests some new numerical algorithms on latent variables. Although not immediately relevant to the applications in the last chapter, we show that their at large n on toy examples.

Chapter 6 performs partial order modelling of two real datasets, one on the social ordering between 12th century bishops and the other on the ranks of the 1955-2002 US Masters golf tournaments. The numerical inference uses the Particle MCMC algorithm developed. We are able to show partial order point statistics such as the posterior consensus order and to plot the posterior depth distribution. These numerical results help to clarify whether the hierarchy's depth has changed over time and to test whether the abrupt order changes seen to occur between some elements are caused by mutations events or change points.

Appendix A

Search of controllable depth

This section gives two transition distributions for searching through partial orders with some control over partial order depth. A *search* distribution is defined here as any distribution with some location parameter. Depending on applications, a single step transition needs to have either a tractable density or a fast sampler. Irreducibility is required. If it is asymmetric and used as a proposal in MCMC for example, then its density needs to be computed in the acceptance probability. However, in some optimization schemes, sampling a set of candidates fast is more important.

This section first describes some distributions found in the literature, usually work where authors applied an MCMC or optimization algorithm requiring a distribution with local moves. Typically, these are rudimentary single edge add or delete operations on DAGs or partial orders.

This section then suggests new alternative search distributions. They have location and scale parameters, but their moves are not *local* in the same way as edge add or delete operations. For example one sampler ensures that at least one linear extension remains in common after one transition, but this does not mean that only a small number of edges are updated. They have a scale parameter for adjusting some distance between partial order samples, with as possible application preventing local optima.

Their main purpose is that the depth is controlled: the distributions have a parameter in which the expected depth is monotonic. In particular, it is possible to ensure that successive partial orders have high depth.

One would expect that search distribution with an additional feature such a control over depth, the computational cost increases. However, this is not always obvious because some of the single edge distributions in the literature resort to an enumeration over graphs or to computing a transitive closure or reduction, at each iteration.

Another motivation for these distributions is they could serve as discrete time models, they illustrate a preliminary effort in the quest for a continuous-time process such as that of Section 2.2.

Previous work on transition distributions The MCMC algorithm Algorithm 1 for the uniform on partial orders $\mathcal{U}[\mathcal{P}_n]$ uses a proposal on DAGs, where a single edge is added or delete at each transition. A single transition may or may not affect the corresponding partial order (equivalently, the transitive closure).

Mannila and Meek [2000] constructs a transition distribution for series-parallel partial orders making use of their binary construction tree representation, where each leaf (corresponding to an element of the partial order) is at the left or right of a parent (series or parallel operators). At each transition a parent-leaf sub-tree changes location, leaf-parent position and parent type. Given a series-parallel order, a single transition yields $\mathcal{O}(n^2)$ possible orders and any order can be reached after a number of transitions.

Froehlich et al. [2007] describes a transition distribution on transitively closed DAGs for use in a Stochastic Annealing algorithm. At each step, either an edge is added to the (already transitively closed) DAG, or an edge is deleted from its transitive reduction before computing the transitive closure.

Sakoparnig and Beerenwinkel [2012] describes a MCMC proposal distribution on partial orders (and some other application specific parameters). At each step an edge is added or deleted to the partial order. These moves are asymmetric and computing the density involves an enumeration. To prevent local optima, some variants update more edges per step and are included in a mixture proposal.

Case and Brayton [2007] studies an algorithm for maintaining transitive reduction for deterministic edge add and delete operations on directed graph, without storing other graphs such the transitive closure.

A.1 Transition with latent variables

This section suggests some extensions of the (K, ρ, n)-dimensional model as transition distributions. This a first step towards a continuous-time model, because a transition distribution can be viewed as some discrete-time model.

Here, transitions are on latent variables and therefore computing the induced partial order (if it is of interest) is part of the computational cost. Figure A.1 illustrates a transition on the Z matrix where one of the paths is updated. Below are

the sorted the K columns of the Z matrices. This suggests a choice on whether to update to Z matrix (continuous variables) or a set of total orders (discrete variables). From a modelling perspective, both are similar, however there are some differences computationally. Transition distributions for both versions are presented.

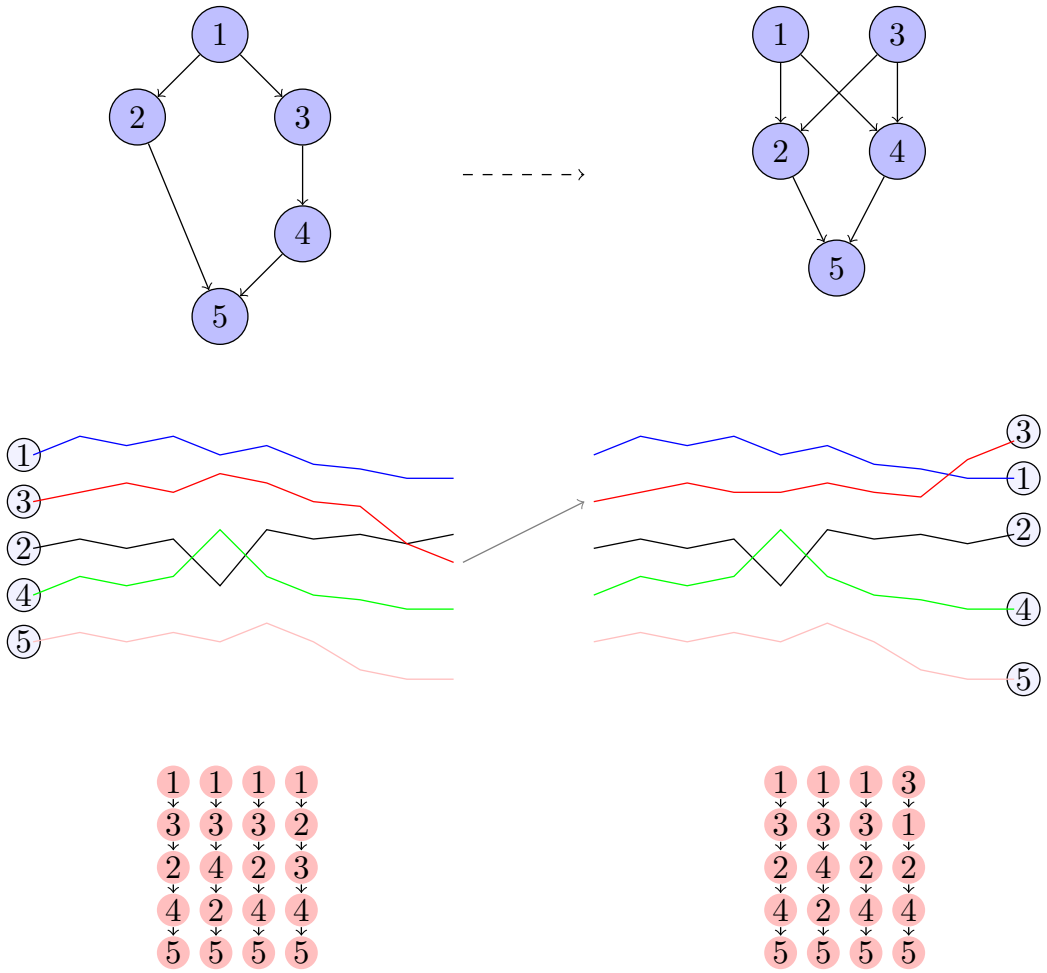

Figure A.1: The Z matrix (middle) with a single path update, the sorted matrix (below) and the implied partial order transition (above).

Continuous variables A latent variable $Z \in \mathcal{M}_{n,K}$ has some of its entries updated for a transition. On Figure A.2, the whole path of a single element is updated, while other paths remain fixed. As a transition, if element i has its path z^i updated independently, for example with $z^i \sim \mathcal{N}_K(0, \Sigma)$, then successive partial orders $P(Z_t)$ remain of controllable depth, where t is the transition number. An example of a situation where this may not occur is with the path update $z^i_{t+1}|z^i_t \sim \mathcal{N}_K(z^i_t, \Sigma)$, where the distribution depends on z^i_t and not other paths $z^j, j \neq i$. As $t \to \infty$,

almost surely the paths diverge from each other and depth of $P(Z_t)$ is n. With $z^i \sim \mathcal{N}_K(0, \Sigma)$, the transition is symmetric and irreducible.

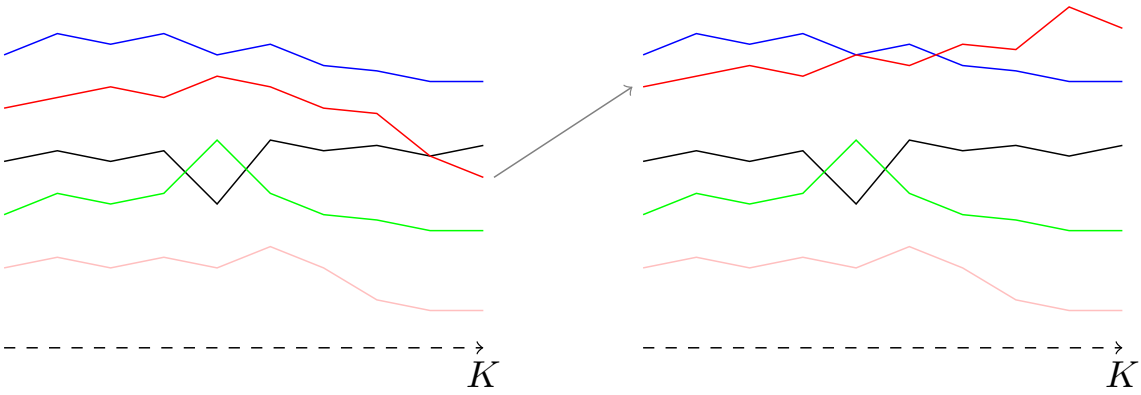

Figure A.2: A transition applied to Z with a single path updated (red).

Permutation based As shown on Figure A.1, a closely related alternative to the transition on the Z variables is the discrete state model consisting of a sequence of K total orders $L_{1:K}$, like in the original (K, n)-dimensional order model. The intersection of these K orders is a partial order. The following is an example of a symmetric and irreducible transition distribution with controllable depth: for a transition from $L_{1:K,t}$ to $L_{1:K,t+1}$, the first order $L_{1,t+1}$ in the sequence $L_{1:K}$ is sampled given $L_{1,t}$ and independently from the other $K-1$ orders according to permutation transposition. The remaining $K-1$ orders are sampled independently given $L_{1,t+1}$, also with a transposition process.

A.2 Transition with shared linear extensions

This section gives search distributions where consecutive partial orders have one or more linear extensions in common. Here, the shared linear extensions serve as a concept of proximity. This is an alternative to standard transition distributions where "local" means that a small number of edges are updated. Contrary to those in the previous section, they are not based on some latent variables.

A first attempt For two partial orders P and Q in $\mathcal{P}_n$ on n elements, a distance between between the linear extensions sets $\mathcal{L}(P)$ and $\mathcal{L}(Q)$ could be used to define a distance $d(P, Q)$ between P and Q. After all, there is a bijection between P and $\mathcal{L}(P)$ (P is the intersection of its linear extensions).

A linear extension can be interpreted as a permutation, therefore a distance on permutations such as the minimum number of adjacent transpositions would also apply between elements of $\mathcal{L}(P)$ and $\mathcal{L}(Q)$. Such distance induces a Hausdorff distance between $\mathcal{L}(P)$ and $\mathcal{L}(Q)$. This would allow to assign a small distance between two partial orders even if a small number of order conflicts ($a_i \prec_P a_j$ and $a_j \prec_Q a_i$ for a few (i, j) pairs) exist between P and Q

An alternative distance is to count the number of linear extension in the set difference between $\mathcal{L}(P)$ and $\mathcal{L}(Q)$. For a given P, if $\mathcal{L}(Q)$ increases in size, this tends to increase such distance. Therefore it distances would tend to be smaller between partial orders of higher depth. However if it is normalized by $|\mathcal{L}(P)| + |\mathcal{L}(Q)|$, such quantity is 1 if there exists any order conflict between P and Q and 0 if $P = Q$.

For a given distance $d(P, Q)$, merely specifying a Mallows distribution of the form

$$p(P|Q, \lambda) = K(Q, \lambda) \exp(-\lambda d(P, Q))$$

is not necessarily useful as a search proposal, because sampling from it fast is not obvious and computing $K(Q, \lambda)$ exactly may require enumerating partial orders.

The following distribution involves sharing linear extensions: given Q, K linear extensions are sampled from the uniform on $\mathcal{L}(Q)$, K' total orders are sampled uniformly at random and then P is taken to be the intersection of these $K + K'$ total orders. If these are sampled independently, the transition probability is

$$p(P|Q) = \frac{\mathbb{1}\{|\mathcal{L}(P) \cap \mathcal{L}(Q)| \geq K\}}{n!^{K'}}.$$

If P is simulated by the process described, then the numerator is 1 and the denominator is a constant, therefore the density is tractable. However, the fact that K' independent total orders are added to the intersection means that P may have low depth even if Q has high depth. The following transition distribution retains the concept of shared linear extension, but gives more control on depth.

From a linear order to a partial order Given a total order y and the adjacency matrix M of its transitively reduced DAG, Binomial distributed numbers $N_d \sim \mathcal{B}(p_d, n-1)$ of edges are deleted from M and $N_a \sim \mathcal{B}(p_a, \frac{(n-1)(n-2)}{2})$ of edges compatible with y are added to M uniformly at random. There are a most $n-1$ edges to delete from M and $\frac{(n-1)(n-2)}{2}$ edges compatible with y that can be added. If M is represented with y as basis, this corresponds to adding or deleting edges in the

upper triangular entries. For example with $n = 5$, $y = \{a_5 \prec \ldots \prec a_1\}$, $N_d = 1$ and $N_a = 3$,

$$M = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

P is then the partial order corresponding to the transitively closure of M, its probability is

$$p(P|y, p_d, p_a) = \frac{\mathcal{B}(N_d|p_d, n-1)}{\binom{n-1}{N_d}} \sum_{M \in T_{adj}} \frac{\mathcal{B}(N_a|p_a, (n-1)(n-2)/2)}{\binom{(n-1)(n-2)/2}{N_a}}, \tag{A.1}$$

where $T_{adj}(P)$ is the set of adjacency matrices M such that the transitive closure of M is P and $N_a(M)$ is the number of edges in M excluding those from y. y serves as a location parameter, whereas p_d controls the scale and both p_a and p_d affect the depth of P. The parameters p_a and p_d control the depth of P.

Equation A.1 can be simplified to a closed-form expression (without a sum) as a function of the number of edges in the transitive closure and in the transitive reduction of the DAG of P, N_c and N_r respectively. P corresponds to a total of $2^{N_c - N_r}$ DAGs, a fact used by the uniform $\mathcal{U}[\mathcal{P}_n]$ sampler Algorithm 1. There are $\binom{N_c - N_r}{k - N_r}$ of these DAGs with k edges.

Computing N_c and N_r for a given P is the main computational cost for computing $p(P|y, p_d, p_a)$. Computational cost for simulating it involves Binomial distributions, a total of $\mathcal{O}(n^2)$ Bernoulli random variables.

Shared linear extension This can be extended to a distribution on P given a partial order Q by sampling a linear extension $y \sim \mathcal{U}[\mathcal{L}(Q)]$ and then sampling P based on y as described with $P \sim p(\cdot|y, p_d, p_a)$. A realization is shown Figure A.3. Starting with any partial order Q and given probabilities $0 < p_a < 1$ and $0 < p_d < 1$, it is possible to reach any partial order in $\mathcal{P}_n$ within two steps: all edges are deleted in the first step, resulting in the null order, from which an arbitrary total order y is sampled, yet every partial order P can be sampled from $p(P|y, p_d, p_a)$ with suitable y (a partial order is represented by an upper-triangular adjacency matrix in the right basis).

Computing the density of the resulting distribution

$$p(P|Q, p_d, p_a) = \frac{1}{|\mathcal{L}(Q)|} \sum_{y \in \mathcal{L}(P \cap Q)} p(P|y, p_d, p_a)$$

involves a sum over linear extensions common to both P and Q, which is small if P and Q have high depth. Linear extensions common to two partial orders P and Q can be enumerated as the set of linear extensions from the partial order with the unions of the edges (for an algorithm, the adjacency matrices are summed).

An alternative model to control the computational cost associated with such sum and of sampling y is to sample a linear extension y from Q non-uniformly. There exist standard algorithms to return a single, arbitrary linear extension from a given partial order.

An alternative to ensure symmetry (and therefore avoid computing densities in MCMC) is to fix N_a and N_d. However, the result is not irreducible any more.

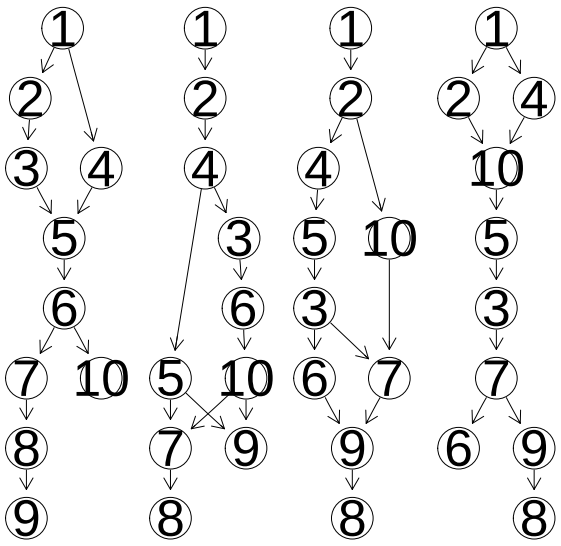

Figure A.3: Random walk on $\mathcal{P}_{10}$ with $p(P; P_0)$ from left to right with $p_d = 1/4$ and $p_a = 1/2$.

Appendix B

MCMC plots

The following shows MCMC traces plots and histograms for rates λ_C and λ_S for the applications 6. Iterations were stopped when the effective sample size reached several thousands.

Bishops 1120-30

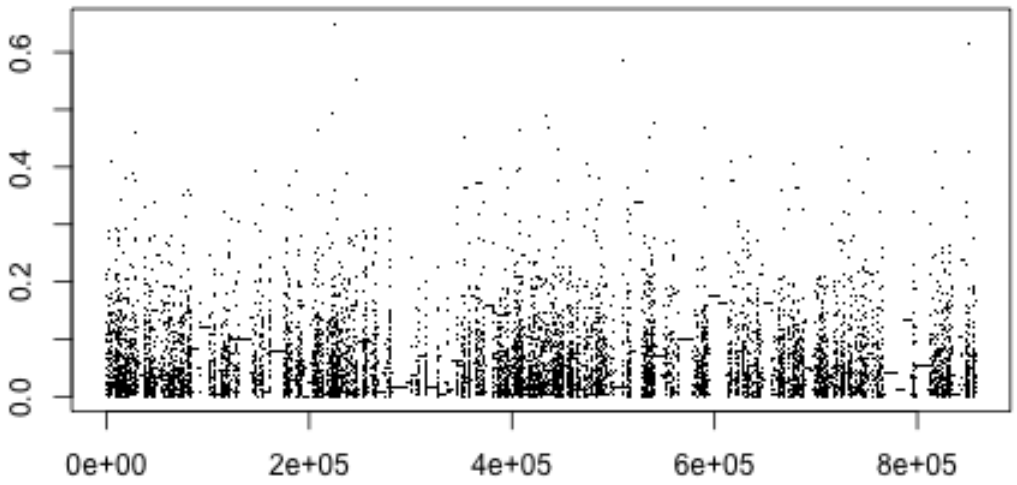

Figure B.1: MCMC trace plot for the change point rate λ_C.

Bishops 1120-30

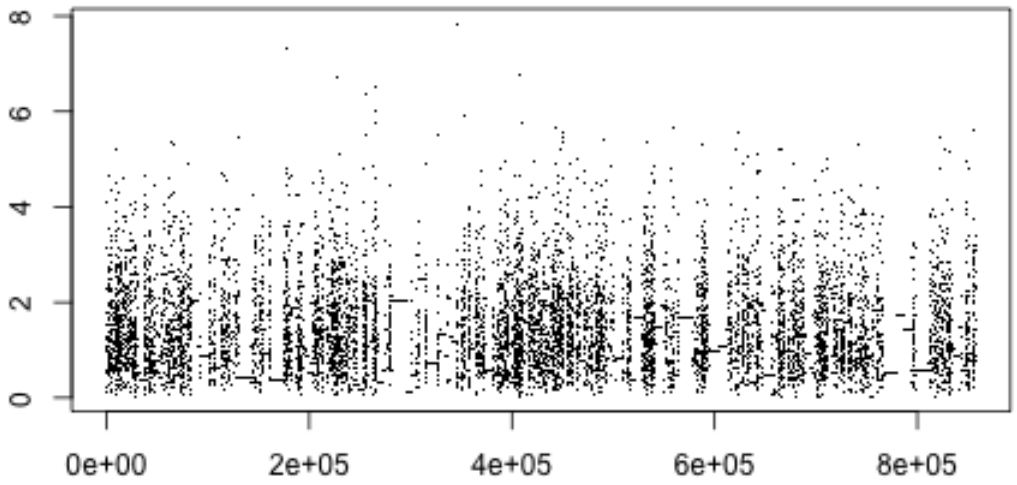

Figure B.2: MCMC trace plot for the mutation rate λ_S.

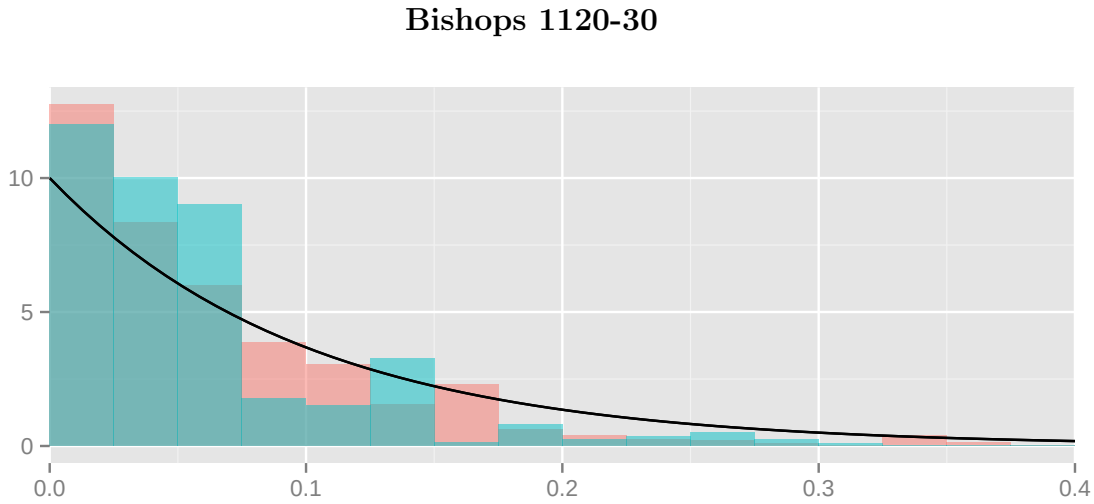

Figure B.3: Empirical posterior densities based on the complete MCMC chain (red) and the last 20% iterations (blue) for the change point rate λ_C with a curve of its prior density $\lambda_C \sim \mathcal{E}(10)$ (black).

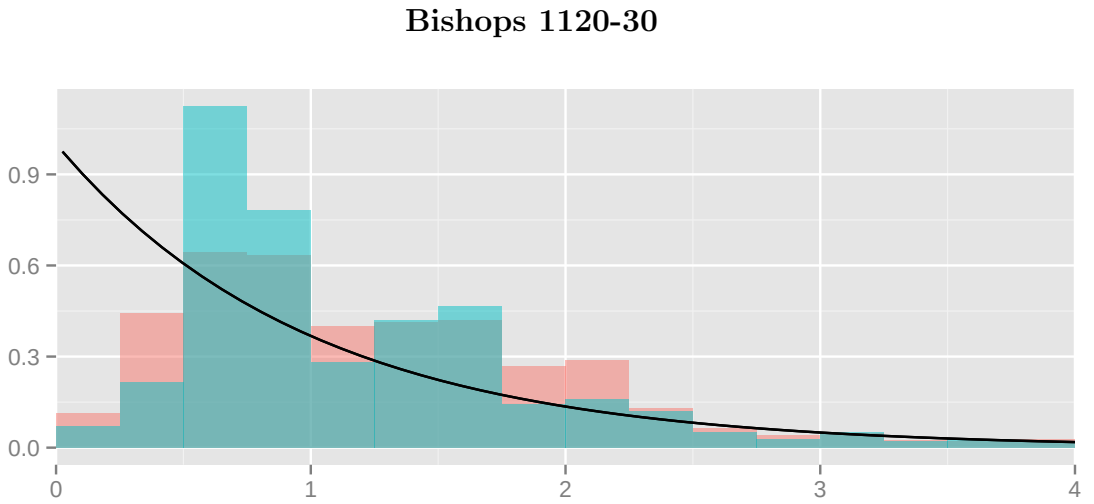

Figure B.4: Empirical posterior densities based on the complete MCMC chain (red) and the last 20% iterations (blue) for the change point rate λ_S with a curve of its prior density $\lambda_S \sim \mathcal{E}(1)$ (black).

Bishops 1130-40

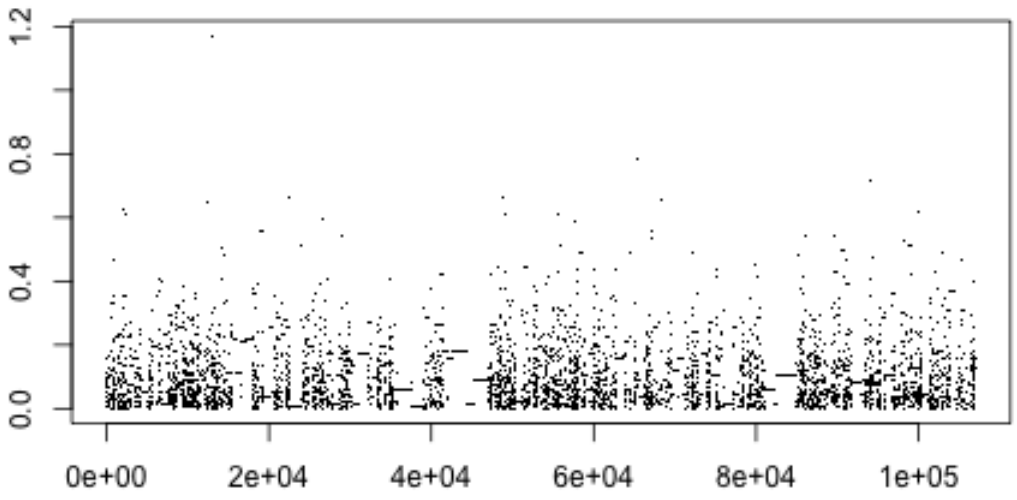

Figure B.5: MCMC trace plot for the change point rate λ_C.

Bishops 1130-40

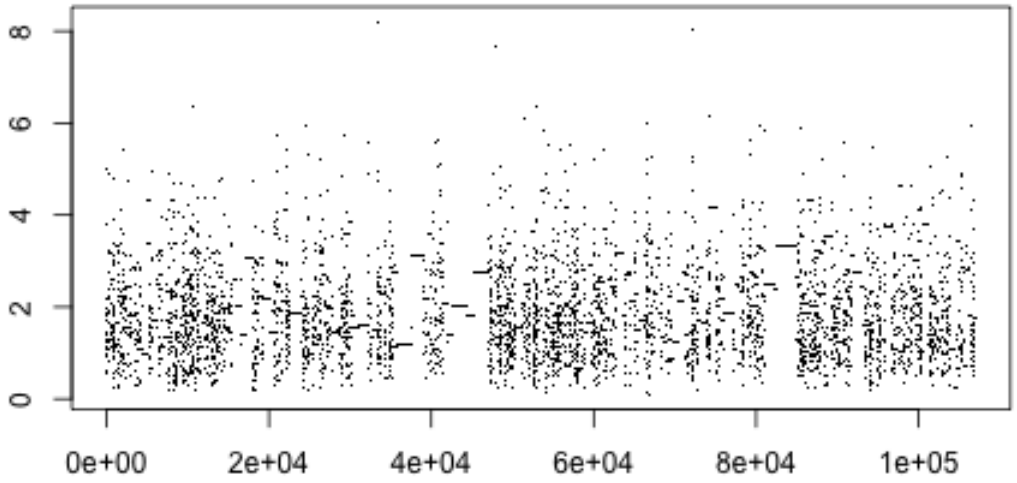

Figure B.6: MCMC trace plot for the mutation rate λ_S.

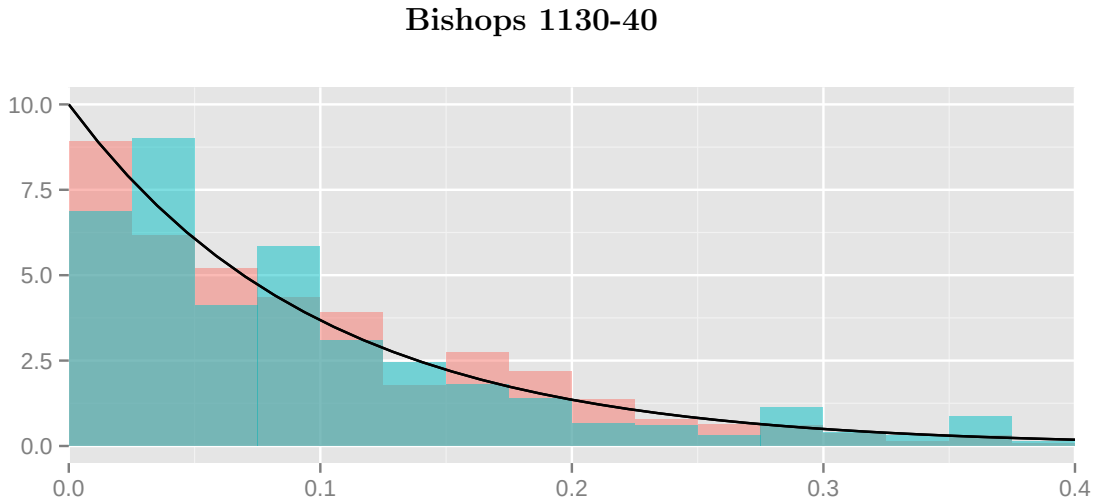

Figure B.7: Empirical posterior densities based on the complete MCMC chain (red) and the last 20% iterations (blue) for the change point rate λ_C with a curve of its prior density $\lambda_C \sim \mathcal{E}(10)$ (black).

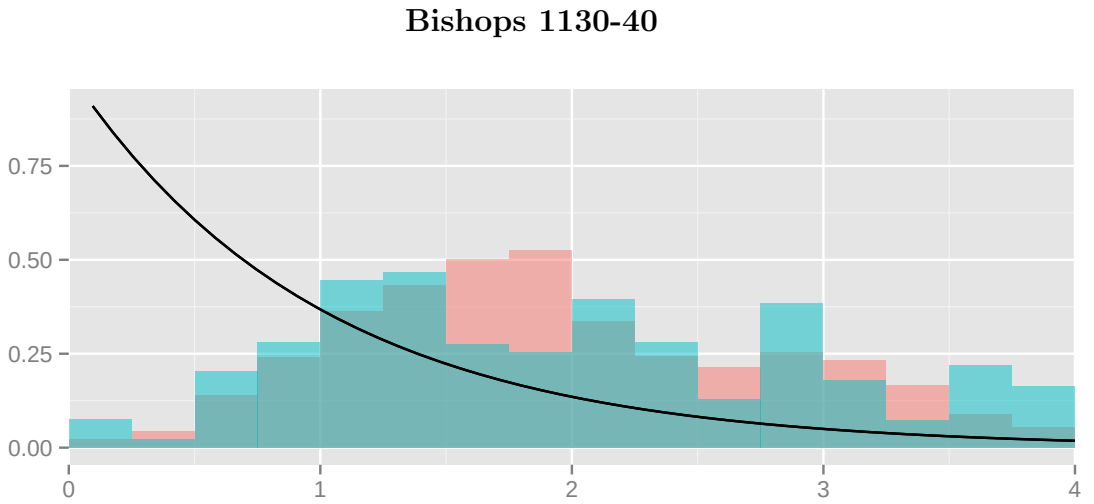

Figure B.8: Empirical posterior densities based on the complete MCMC chain (red) and the last 20% iterations (blue) for the change point rate λ_S with a curve of its prior density $\lambda_S \sim \mathcal{E}(1)$ (black).

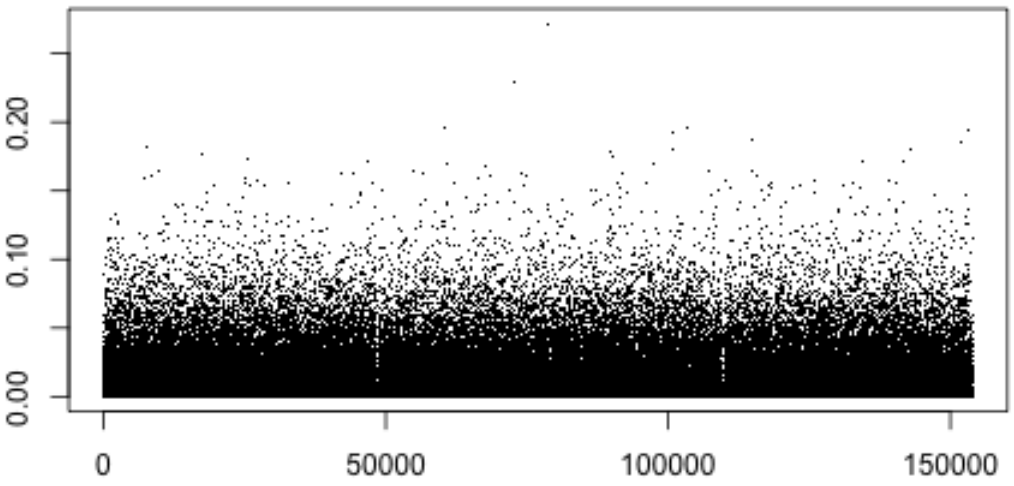

Figure B.9: MCMC trace plot for the change point rate λ_C.

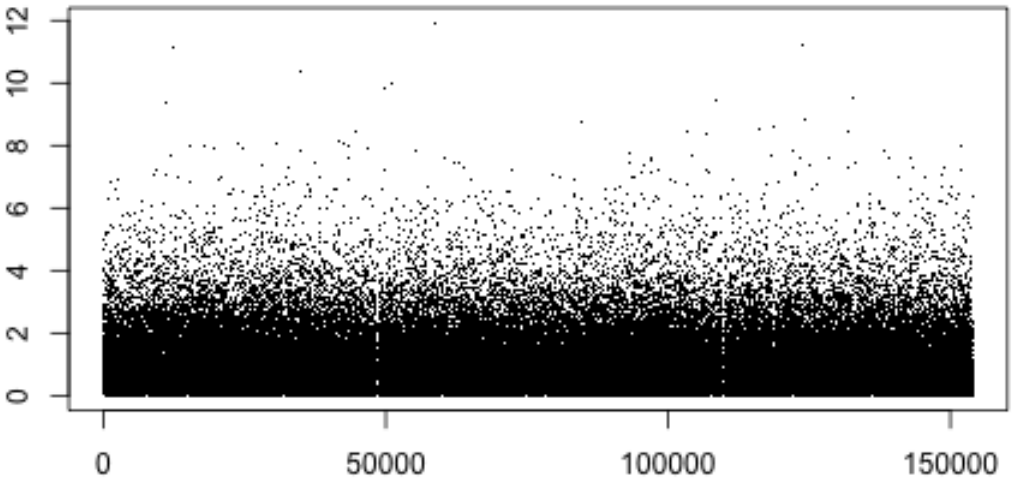

Figure B.10: MCMC trace plot for the mutation rate λ_S.

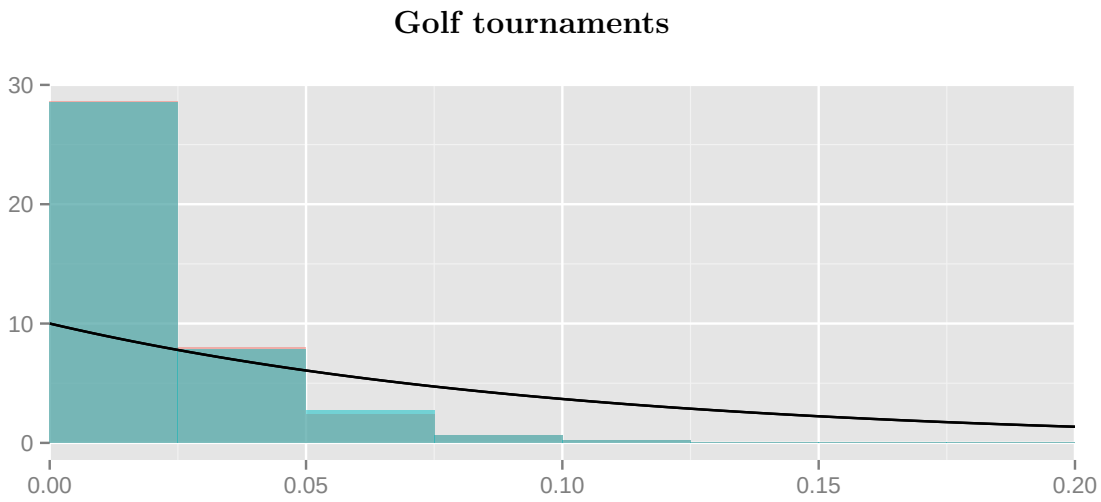

Figure B.11: Empirical posterior densities based on the complete MCMC chain (red) and the last 20% iterations (blue) for the change point rate λ_C with a curve of its prior density $\lambda_C \sim \mathcal{E}(10)$ (black).

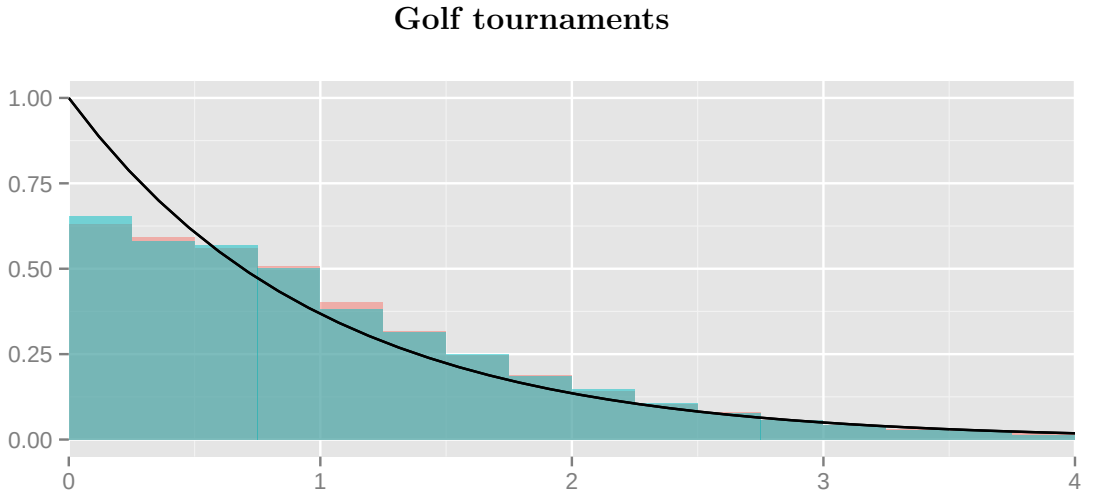

Figure B.12: Empirical posterior densities based on the complete MCMC chain (red) and the last 20% iterations (blue) for the change point rate λ_S with a curve of its prior density $\lambda_S \sim \mathcal{E}(1)$ (black).

Appendix C

Names reference

Table C.1: Bishops 1120-30

Element	Bishop
1	William, Giffard, bishop of Winchester, 1100-1129
2	Roger, Bishop of Salisbury
3	Richard, Bishop of Bayeux
4	John, Bishop of Lisieux"
5	William, de Warelwast, bishop of Exeter
6	Bernard, Bishop of St David's
7	Ouen, Bishop of Evreux
8	Urban, bishop of Llandaff
9	Alexander, Bishop of Lincoln
10	Richard, de Belmeis I, bishop of London, 1108-1127

Table C.2: Bishops 1130-40

Element	Bishop
1	Roger, Bishop of Salisbury
2	Philip, Bishop of Bayeux
3	John, Bishop of Lisieux
4	Bernard, Bishop of St David's
5	Ouen, Bishop of Evreux
6	Geoffrey, Rufus, Bishop of Durham
7	Nigel, Bishop of Ely
8	Henry, de Blois, Bishop of Winchester, 1129-1171
9	Robert, de Bethune, Bishop of Hereford

Table C.3: Golf

Element	Player
1	Arnold Palmer
2	Ben Hogan
3	Jack Nicklaus
4	Sam Snead
5	Tiger Woods
6	Nick Faldo
7	Gary Player
8	Tom Watson
9	Lee Trevino
10	Seve Ballesteros

Bibliography

Noga Alon, Bela Bollobas, Graham Brightwell, and Svante Janson. Linear Extensions of a Random Partial Order. *The Annals of Applied Probability*, 4(1):108–123, 1994.

Paul D Amer, Christophe Chassot, Thomas J Connolly, Student Member, Michel Diaz, Senior Member, and Phillip Conrad. Partial-Order Transport Service for Multimedia and Other Applications. 2(5), 1994.

Christophe Andrieu and Gareth O Roberts. The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, 37(2):697–725, apr 2009. ISSN 0090-5364. doi: 10.1214/07-AOS574.

Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010. ISSN 1467-9868. doi: 10.1111/j.1467-9868.2009.00736.x.

Rose D. Baker and Ian G. Mchale. Deterministic evolution of strength in multiple comparisons models: Who is the greatest golfer? *Scandinavian Journal of Statistics*, 42(1993):180–196, 2014. ISSN 03036898. doi: 10.1111/sjos.12101.

Paul Balister and Balazs Patkos. Random partial orders defined by angular domains. *Order*, pages 1–18, 2010.

Niko Beerenwinkel, Nicholas Eriksson, and Bernd Sturmfels. Conjunctive Bayesian networks. *Bernoulli*, 13(4):893–909, nov 2007. ISSN 1350-7265. doi: 10.3150/07-BEJ6133.

N Berestycki and Rick Durrett. A phase transition in the random transposition random walk. *Discrete Mathematics and Theoretical Computer Science*, (1995): 17–26, 2003.

N Berestycki and Rick Durrett. Limiting behavior for the distance of a random walk. pages 1–25, 2007.

Bela Bollobas and Graham Brightwell. The structure of random graph orders. *SIAM J. Discrete Math.*, 10(2):318–335, 1997.

R.A. Bradley and M.E. Terry. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4):324–345, dec 1952. ISSN 00063444. doi: 10.2307/2334029.

Graham Brightwell. *Models of random partial orders, Surveys in Combinatorics, London Mathematical Society Lecture Note Series (No. 187)*. Cambridge University Press, 1993.

Graham Brightwell and Peter Winkler. Counting linear extensions. *Order*, 8(3): 225–242, 1991. ISSN 0167-8094. doi: 10.1007/BF00383444.

Francois Caron and Arnaud Doucet. Efficient Bayesian Inference for Generalized Bradley-Terry Models. nov 2010.

Francois Caron and Yee W Teh. Bayesian nonparametric models for ranked data. *Advances in Neural Information Processing Systems 25*, (3):1520–1528, 2012. ISSN 10495258.

J Carpenter, P Clifford, and P Fearnhead. Improved particle filter for nonlinear problems. *Radar, Sonar and Navigation, IEE Proceedings* -, 146(1):2–7, feb 1999. ISSN 1350-2395. doi: 10.1049/ip-rsn:19990255.

Gemma Casas-Garriga. Summarizing Sequential Data with Closed Partial Orders. In *Proceedings of the 2005 SIAM International Conference on Data Mining*, pages 380–391, 2005.

Michael L Case and Robert K Brayton. Maintaining A Minimum Equivalent Graph In The Presence of Graph Connectivity Changes. Technical report, 2007.

D M Ceperley and M Dewing. The Penalty Method for Random Walks with Uncertain Energies. *Journal of Chemical Physics*, 110, 1999.

Nicolas Chopin. Fast simulation of truncated gaussian distributions. pages 1–22, 2012.

J. Andrés Christen and Colin Fox. Markov chain Monte Carlo Using an Approximation. *Journal of Computational and Graphical Statistics*, 14(4):795–810, dec 2005. ISSN 1061-8600. doi: 10.1198/106186005X76983.

Joshua Cooper. Linear Extension Counting is Fixed- Parameter Tractable in Essential Width. In *Special Session on Partially Ordered Sets, AMS Southeastern Sectional Fall 2013*, 2013.

Pierre Del Moral. *Feynman-Kac Formulae - Genealogical and Interacting Particle Systems*. Springer, New-York, mar 2004.

Pierre Del Moral and Arnaud Doucet. Particle Motions in Absorbing Medium with Hard and Soft Obstacles. *Stochastic Analysis and Applications*, 22(5):1175–1207, jan 2005. ISSN 0736-2994. doi: 10.1081/SAP-200026444.

Proceso L. Fernandez, Lenwood S. Heath, Naren Ramakrishnan, and John Paul C Vergara. Mining Posets from Linear Orders. Technical Report 2, Virginia Tech, 2009.

Holger Froehlich, Mark Fellmann, Holger Sueltmann, Annemarie Poustka, and Tim Beissbarth. Large scale statistical inference of signaling pathways from RNAi and microarray data. *BMC bioinformatics*, 8:386, jan 2007. ISSN 1471-2105. doi: 10.1186/1471-2105-8-386.

M. Galassi et al. *GNU Scientific Library Reference Manual (3rd Ed.)*. URL http://www.gnu.org/software/gsl/.

Aristides Gionis, Heikki Mannila, Kai Puolamäki, and Antti Ukkonen. Algorithms for Discovering Bucket Orders from Data. In *KDD '06 Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 561–566, 2006. ISBN 1595933395.

Mark E Glickman. A stochastic rank ordered logit model for rating multi-competitor games and sports. (781):1–33, 2015.

N.J Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlionear/non-Gaussian Bayesian state estimation. *IEE Proceedings-F*, 140:107–113, 1993.

Jayanti Gupta and Paul Damien. Conjugacy class prior distributions on metric-based ranking models. *Journal of the Royal Statistical Society: Series B (Statistical*

Methodology), 64(3):433–445, aug 2002. ISSN 13697412. doi: 10.1111/1467-9868.00343.

Hajivassiliou. Simulation of multivariate normal rectangle probabilities and their derivatives, 1996.

T. Hiraguchi. On the dimension of partially ordered sets. Technical report, Kanazawa University, 1951.

Yuma Inoue and Shin-ichi Minato. An Efficient Method of Indexing All Topological Orders for a Given DAG. 2014.

D. P. Johnson. Bishops and deans: London and the province of Canterbury in the twelfth century. *Historical Research*, 86(234):551–578, 2013. ISSN 09503471. doi: 10.1111/1468-2281.12021.

Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. URL http://www.scipy.org/.

Alexander Karzanov and Leonid Khachiyan. On the conductance of order Markov chains. *Order*, 8(1):7–15, 1991. ISSN 0167-8094. doi: 10.1007/BF00385809.

D. J. Kleitman and B. L. Rothschild. Asymptotic enumeration of partial orders on a finite set. 205(1970):205–220, 1975.

D. E. Knuth. *The Art of Computer Programming, vol. 2: Seminumerical Algorithms*. Addison-Wesley, 1969.

D. E. Knuth and Jayme L. Szwarcfiter. A Structure Program to Generate All Topological Sorting Arrangements. Technical report, Computing Laboratory, University of Newcastle Upon Tyne, 1974.

Wing-Ning Li, Zhichun Xiao, and Gordon Beavers. On Computing the Number of Topological Orderings of a Directed Acyclic Graph. In *Congressus Numerantium*, pages 143–159. Utilitas Mathematica Pub., 2005.

Rorbert Duncan Luce. *Individual Choice Behavior: A Theoretical Analysis*. Wiley, 1959.

C. L. Mallows. Non-Null Ranking Models. I. *Biometrika*, 44:114–130, 1957.

Heikki Mannila and Christopher Meek. Global Partial Orders from Sequential Data. pages 161–168, 2000.

Rolf H. Möhring. Computationally tractable calsses of ordered sets. In Ivan Rival, editor, *Algorithms and Order*, pages 105–194. Springer Netherlands, 1989.

Geoff K. Nicholls and Alexis Muir Watt. Partial Order Models for Episcopal Social Status in 12th Century England. *Proceedings of the 26th International Workshop on Statistical Modelling*, 2011.

Geoff K Nicholls, Bernard W Silverman, David Johnson, and Nicholas E Karn. Bayesian inference for a partial order from random linear extensions. 2010.

Marcin Peczarski. Sorting 13 Elements Requires 34 Comparisons. In *Algorithms ESA 2002 Lecture Notes in Computer Science*, number 13, pages 785–794. Springer, 2002.

Michael Pitt, Ralph Silva, Paolo Giordani, and Robert Kohn. Auxiliary Particle filtering within adaptive Metropolis-Hastings Sampling. jun 2010.

R. L. Plackett. The Analysis of Permutations. *Journal of the Royal Statistical Society, Series C*, 24(2):193–202, 1975.

Gara Pruesse and Frank Ruskey. Generating Linear Extensions Fast, 1994.

Kai Puolamäki, Mikael Fortelius, and Heikki Mannila. Seriation in paleontological data using markov chain Monte Carlo methods. *PLoS computational biology*, 2(2): e6, feb 2006. ISSN 1553-7358. doi: 10.1371/journal.pcbi.0020006.

Thomas Sakoparnig and Niko Beerenwinkel. Efficient sampling for Bayesian inference of conjunctive Bayesian networks. *Bioinformatics (Oxford, England)*, 28(18):2318–24, sep 2012. ISSN 1367-4811. doi: 10.1093/bioinformatics/bts433.

Øystein Sørensen, Valeria Vitelli, Arnoldo Frigessi, and Elja Arjas. Bayesian inference from rank data. 2:1–59, may 2014.

L.L. Thurstone. A law of comparative judgement. *Psychological Review*, 34:273–286, 1927.

Antti Ukkonen, Mikael Fortelius, and Heikki Mannila. Finding partial orders from unordered 0-1 data. *Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining - KDD '05*, page 285, 2005. doi: 10.1145/1081870.1081904.

Y. L. Varol and D. Rotem. An algorithm to generate all topological sorting arrangements. *The Computer Journal*, 24, 1981.

Mark B. Wells. *Elements of Combinatorial Computing*. Pergamon Press, 1971.

Theron [1] Westervelt. Royal charter witness lists and the politics of the reign of Edward IV. *Historical Research*, 81(212):211–223, 2008. ISSN 0950-3471. doi: 10.1111/j.1468-2281.2006.00406.x.

Peter Winkler. Random orders. *Order*, 16(4):317–331, 1985. doi: 10.1007/BF00582738.