
Bayesian structure discovery in Bayesian networks with less space

Pekka Parviainen

Helsinki Institute for Information Technology HIIT and Department of Computer Science,
University of Helsinki, Finland

{pekka.parviainen, mikko.koivisto}@cs.helsinki.fi

Mikko Koivisto

Abstract

Current exact algorithms for score-based structure discovery in Bayesian networks on n nodes run in time and space within a polynomial factor of 2^n . For practical use, the space requirement is the bottleneck, which motivates trading space against time. Here, previous results on finding an optimal network structure in less space are extended in two directions. First, we consider the problem of computing the posterior probability of a given arc set. Second, we operate with the general partial order framework and its specialization to bucket orders, introduced recently for related permutation problems. The main technical contribution is the development of a fast algorithm for a novel zeta transform variant, which may be of independent interest.

1 INTRODUCTION

Score-based structure discovery in Bayesian networks has attracted lots of interest in the past couple of decades. There are two major approaches. One is to find an optimal Bayesian network structure, that is, a directed acyclic graph (DAG) that maximizes the sum of local scores (Cooper and Herskovits, 1992; Heckerman et al., 1995). The other, an inherently Bayesian approach, is to compute the posterior probabilities of subgraphs of interest, like individual arcs (Friedman and Koller, 2003; Koivisto and Sood, 2004). Being hard computational problems, heuristic methods represent the state of the art for large problem instances. Recent algorithmic developments, however, have made it possible to solve *exactly* moderate-sized problems.

Appearing in Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS) 2010, Chia Laguna Resort, Sardinia, Italy. Volume 9 of JMLR: W&CP 9. Copyright 2010 by the authors.

The fastest known algorithms find an optimal DAG by dynamic programming in time and space within a polynomial factor of 2^n , where n is the number of nodes (Ott and Miyano, 2003; Koivisto and Sood, 2004; Silander and Myllymäki, 2006). In practical implementation the space requirement is the bottleneck. For example, the Silander–Myllymäki algorithm finds an optimal DAG on 29 nodes in about 50 CPU hours, however, taking 89 GB of space (Silander and Myllymäki, 2006); while parallelization (to up to n CPUs) renders the time requirement feasible even for larger n , the space requirement becomes soon prohibitive.

Likewise, the posterior probability of an arbitrary fixed arc set can be computed by analogous dynamic programming techniques in time and space within a polynomial factor of 2^n (Koivisto and Sood, 2004; Koivisto, 2006). Interestingly, these results rely on the assumption that the prior distribution on DAGs obeys a special structure, which deviates, for example, from the simplest uniform distribution; if adhering to the uniform distribution, the fastest known algorithm is substantially slower, taking time $O(3^n)$ and space $O(2^n)$ (Tian and He, 2009). The space requirement is again the bottleneck in the former algorithms, but not in the latter.

The extensive space requirement of current algorithms has motivated some recent studies on trading space against time. Parviainen and Koivisto (2009) introduced space–time tradeoffs for finding an optimal DAG. Their algorithm is based on an observation that an optimal DAG compatible with a fixed partial order can be found in time and space that is typically much less than 2^n . To guarantee that an over-all optimal DAG is found, several partial orders that together “cover” all the $n!$ linear orders are considered one by one. Because the problems for different partial orders can be solved independently, the algorithm parallelizes easily and efficiently to thousands of processors. Parviainen and Koivisto (2009) implemented and analyzed a particular instantiation of this template, called the pairwise scheme.

In this paper, we extend the partial order based approach to compute the marginal posterior probabilities of arc sets or individual arcs in less space. After introducing some basic concepts and notation in Section 2, we present the main algorithm in Section 3; the contribution is rather conceptual: we adapt the generic partial order framework that was recently presented for related permutation problems by Koivisto and Parviainen (2010), of which the work of Parviainen and Koivisto (2009) is another, more specific instantiation. Existing techniques, however, seem insufficient for efficient implementation of the main algorithm. Indeed, it involves a critical subtask that calls for a fast computation of a novel zeta transform variant. As related zeta and Möbius transforms have applications well beyond the present context of Bayesian networks, e.g., in the Dempster–Shafer theory of evidence (Kennes, 1992) and in algorithm theory (Björklund et al., 2007, 2008; Nederlof, 2009), we present our fast algorithm in plain terms in Section 4. Then, in Section 5, we adapt the forward–backward algorithm of Koivisto (2006) to compute the posterior probabilities of all the $n(n-1)$ possible arcs simultaneously in about the same time and space as for a single arc (or any other fixed arc set).

2 PRELIMINARIES

We begin with a Bayesian treatment of Bayesian networks, describing a basic dynamic programming algorithm for computing the posterior probabilities of structural features, and introducing some basic concepts and notation related to partial orders.

2.1 BAYESIAN NETWORKS AND STRUCTURAL FEATURES

The essence of a Bayesian network is captured by its structural component: a directed acyclic graph (DAG). Consider a DAG with node set $N = \{1, 2, \dots, n\}$ and arc set $A \subseteq N \times N$. We identify the DAG with its arc set and write A_v for the *parents* of node v , that is, $A_v = \{u : uv \in A\}$. Each node v is associated with m random variables D_{v1}, \dots, D_{vm} , which constitute the v th row of an $n \times m$ matrix D . Commonly made assumptions of independence (or exchangeability) imply that the rows of D are conditionally independent given the “parent rows”. In terms of probability (density),

$$p(D|A) = \prod_{v \in N} p(D_v | D_{A_v}, A_v).$$

We assume that D , called the data, is fully observed, and the task is to “learn” A . To this end, we de-

fine a prior $p(A)$ and consider the posterior probability of some structural feature $f(A)$, written simply as $p(f|D)$. For convenience, we will assume that f is an indicator function with the range $\{0, 1\}$.

To facilitate both representation and, more importantly, computation, we assume order-modularity (Friedman and Koller, 2003; Koivisto and Sood, 2004), which stems from two requirements for the prior $p(A)$. First, the prior $p(A)$ is obtained as the marginal of a joint prior $p(L, A)$, where L stands for a postulated, “true” linear order on the nodes. We write L_v for the set of nodes that precede v in the order L and say that A and L are compatible with each other if $A_v \subseteq L_v$ for each v , that is, L is an extension of A . Second, we assume that $p(L, A)$ vanishes if A and L are not compatible with each other, and otherwise it factorizes into a product of terms $\rho_v(L_v)q_v(A_v)$, one for each node v , where the ρ_v and q_v are nonnegative functions.

The feature f is also assumed to be modular, that is, $f(A)$ factorizes into a product of terms $f_v(A_v)$, one for each node v . For example, the indicator of an arc uv is represented by letting $f_v(A_v) = 1$ if $u \in A_v$ and $f_v(A_v) = 0$ otherwise, and $f_w(A_w) = 1$ for all $w \neq v$ and all A_w . Throughout the paper, we assume for simplicity that each f_v is an indicator function; we may sometimes write shortly f_v for the the event $f_v(A_v) = 1$.

2.2 COMPUTATION OF POSTERIOR PROBABILITIES

The following approach to compute the posterior probability of a structural feature f is due to Koivisto and Sood (2004). For each node v and parent set A_v define the local score as

$$\beta_v(A_v) = q_v(A_v)p(D_v | D_{A_v}, A_v)f_v(A_v).$$

In these terms, the joint probability that L_v are the predecessors of v in the linear order L , that the data are D_v , and that the local feature is f_v , is obtained by marginalizing out the A_v , as

$$\alpha_v(L_v) = \rho_v(L_v) \sum_{A_v \subseteq L_v} \beta_v(A_v).$$

The sum on the right is known as the *zeta transform* of β_v , evaluated at L_v . Now, because of modularity, the joint probability of the data and the feature is obtained by marginalizing out the linear order L :

$$p(D, f) = \sum_L \prod_{v \in N} \alpha_v(L_v). \quad (1)$$

We can compute $p(D, f)$ recursively. Define the forward sum as a function F over all subsets of N by

$$F(\emptyset) = 1,$$

$$F(S) = \sum_{v \in S} \alpha_v(S \setminus \{v\}) F(S \setminus \{v\}), \quad \emptyset \subset S \subseteq N.$$

Then it holds that $F(N) = p(D, f)$.

Finally, the posterior probability can be computed as the ratio $p(f|D) = p(D, f)/p(D)$, where $p(D)$ is computed like $p(D, f)$ but with f replaced by the trivial indicator function that evaluates everywhere to 1.

2.3 PARTIAL ORDERS AND IDEALS

The partial order approach of Parviainen and Koivisto (2009) and Koivisto and Parviainen (2010), which we will extend to computing the posterior probabilities of structural features, operates on the following central concepts and notation.

A *partial order* P on *baseset* M is a subset of $M \times M$ such that for all $x, y, z \in M$ it holds that $xx \in P$ (reflexive), $xy \in P$ and $yx \in P$ implies $y = x$ (anti-symmetry), and $xy \in P$ and $yz \in P$ implies $xz \in P$ (transitivity); P is a *linear order* (or, total order) if, in addition, $xy \in P$ or $yx \in P$ (comparability). Another partial order Q on M is an *extension* of P if $P \subseteq Q$. Note that a partial order fully specifies its baseset. If $xy \in P$ we say that x *precedes* y (in P).

A family of partial orders \mathcal{P} is an *exact cover* of, or *exactly covers*, the linear orders on N if every linear order on N is an extension of exactly one member of \mathcal{P} .

An *ideal* (or down-set) of a partial order P is a set of elements that “begin” a linear extension of P . Formally, an ideal I of a partial order P is a subset of elements such that if $y \in I$ and $xy \in P$, then $x \in I$. Another ideal S of P is called a *subideal* of I if $S \subseteq I$. We denote by $\mathcal{I}(P)$ the set of all ideals of P .

3 POSTERIOR PROBABILITIES OF STRUCTURAL FEATURES

We consider the computation of the joint probability of the data D and a structural feature f using Equation 1. To reduce the space requirement of the basic dynamic programming algorithm, we split the summation over the linear orders on N into a double summation: the inner summation runs through the linear extensions of a fixed partial order, while the outer summation runs through an appropriate set of partial orders. To this end, let \mathcal{P} be a family of partial orders that exactly covers the linear orders on N . Then

$$p(D, f) = \sum_{P \in \mathcal{P}} \sum_{P \subseteq L} \prod_{v \in N} \alpha_v(L_v),$$

where L runs through the linear extensions of P .

The inner sum can be computed in a forward manner. Let $F^P(\emptyset) = 1$ and for every nonempty ideal S of P , let

$$F^P(S) = \sum_{v \in S: S \setminus \{v\} \in \mathcal{I}(P)} \alpha_v(S \setminus \{v\}) F^P(S \setminus \{v\}).$$

Then one can show by simple induction that $F^P(N)$ equals the inner sum above:

$$F^P(N) = \sum_{P \subseteq L} \prod_{v \in N} \alpha_v(L_v).$$

We analyze the time and space requirement of evaluating $p(D, f)$ using the recurrence for F^P . For a moment we assume that the values $\alpha_v(L_v)$ can be accessed with no cost in time or space. Then the time requirement is $O(n|\mathcal{I}(P)|)$ for computing $F^P(N)$ for a fixed P , and hence $O(n \sum_{P \in \mathcal{P}} |\mathcal{I}(P)|)$ for computing $p(D, f)$. The space requirement is $O(\max_{P \in \mathcal{P}} |\mathcal{I}(P)|)$.

Consider then the computation of the values $\alpha_v(L_v)$. If these were precomputed and stored for every possible v and L_v , the space requirement would be of order $n2^n$ and no reduction in space requirement is obtained. Fortunately, given that the partial order P is fixed, we only need the values $\alpha_v(L_v)$ for all v and ideals L_v of P . We will show in the next section (Theorem 7) that for a fixed v these values can be computed in time $O(n|\mathcal{I}(P)|)$ and space $O(|\mathcal{I}(P)|)$, under a mild assumption: namely that large parent sets can be ignored, that is, the prior on A vanish when A_v is larger than k for some v , with $\sum_{i=0}^k \binom{n}{i} \leq |\mathcal{I}(P)|$. This condition is not very restrictive as with the partial orders of interest it allows k to grow linearly with n (Parviainen and Koivisto, 2009; Koivisto and Parviainen, 2010).

Combining the bounds in the previous two paragraphs yields the following.

Theorem 1 *Let \mathcal{P} be a family of partial orders that exactly covers the linear orders on N . Then the posterior probability of a structural feature can be computed in time $O(n^2 \sum_{P \in \mathcal{P}} |\mathcal{I}(P)|)$ and space $O(n \max_{P \in \mathcal{P}} |\mathcal{I}(P)|)$.*

This theorem states the time and space requirement for computing the posterior probability of any fixed arc set. Later, in Theorem 8, we show that the posterior probabilities for all the $n(n-1)$ arcs can be computed simultaneously without increasing time or space requirement (more than by a small constant factor).

It remains to choose a good family of partial orders. Intuitively, a partial order family is good if it contains only a few members, each of which is relatively “thin”. Formally, we call a partial order family *optimal* if the

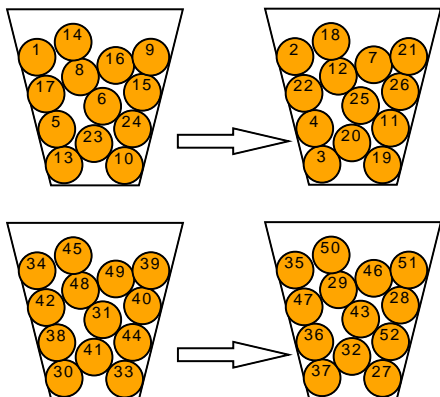


Figure 1: Two 13×13 Bucket Orders. One on the set $\{1, 2, \dots, 26\}$ and the other on the set $\{27, 28, \dots, 52\}$. Any two elements within a bucket are incomparable, whereas every element in the left bucket precedes every element in the right bucket. The two bucket orders in parallel form a partial order on the set $\{1, 2, \dots, 52\}$.

product of the associated time and space requirements (cf. Theorem 1) is as small as possible (Koivisto and Parviainen, 2010).

While finding an optimal family is an open problem in general, we have found (Koivisto and Parviainen, 2010) some rather complete answers concerning a special class of partial order families, called *parallel bucket orders*: the best time–space product is achieved by taking p disjoint groups of 26 nodes and considering all possible ways to arrange the nodes in each group into a bucket order with 13 nodes in the first bucket and 13 in the second; this is called the 13×13 scheme; see Figure 1 for an illustration of 2 parallel 13×13 bucket orders. In particular, the 13×13 scheme is uniformly better than the pairwise 1×1 scheme studied by Parviainen and Koivisto (2009). In Figure 2 we adopt from Koivisto and Parviainen (2010) a graphical comparison of different space–time tradeoff schemes, including also a divide & conquer (D&C) scheme and a naive $m \times (n - m)$ scheme sketched in Parviainen and Koivisto (2009).

For an example of the tradeoff in practice, if $n = 26$ the 13×13 scheme yields about a 4000-fold decrease in the space requirement at about a 2500-fold increase in the time requirement, compared to the basic dynamic programming algorithm.

4 SPARSE ZETA TRANSFORM

Let g be a mapping from the subsets of an n -element set N to the reals. The *zeta transform* of g is another

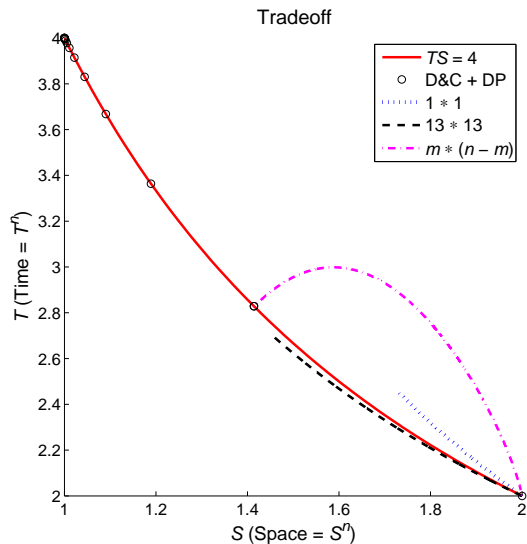


Figure 2: Comparison of Time and Space Requirements of Different Schemes (Koivisto and Parviainen, 2010).

such function \hat{g} defined by

$$\hat{g}(Y) = \sum_{X \subseteq Y} g(X), \quad Y \subseteq N.$$

Given g , the *fast zeta transform* computes \hat{g} in time $O(n2^n)$ and space $O(2^n)$. The algorithm may be considered folklore (Kennes, 1992); a variant of it appears already in Yates (1937).

Algorithm 1 (the fast zeta transform)

Input: $g(X)$ for $X \subseteq N$.

Output: $\hat{g}(Y)$ for $Y \subseteq N$.

1. For $Y \subseteq N$: Let $h_0(Y) = g(Y)$.
2. For $j = 1, \dots, n$:
 - (a) For $Y \subseteq N$:
 - If $j \notin Y$, let $h_j(Y) = h_{j-1}(Y)$;
 - else let $h_j(Y) = h_{j-1}(Y) + h_{j-1}(Y \setminus \{j\})$.
3. Return $h_n(Y)$ for all $Y \subseteq N$.

In our interests is a restricted version of the zeta transform. Namely, we wish to evaluate the transform only at sets Y that are ideals of a given partial order P on N . Note that this generalization subsumes the ordinary zeta transform when P is the trivial partial order $\{uu : u \in N\}$. Naturally, we wish to find a fast algorithm, that is, one that computes all the $\hat{g}(Y)$ for $Y \in \mathcal{I}(P)$ in time (and space) proportional to the

number of ideals of P , up to a factor polynomial in n . However, this goal cannot be achieved in general, since the input function g may be nonzero also at sets that are not ideals of P . Because of this, we develop our algorithm in two phases so as to split the sum in the zeta transform into two nested sums: the outer sum will be only over ideals X of P , while the inner sum will gather the needed terms for each X .

We begin the derivation of with some key definitions. For any ideal Y of P denote

$$\check{Y} = \{u \in Y : uv \notin P \text{ for all } v \in Y \setminus \{u\}\}.$$

In words, \check{Y} consists of those elements of Y that are maximal in Y . Furthermore, define the *tail* of Y as the collection of subsets that are “in between” Y and \check{Y} :

$$\mathcal{T}_Y = \{Z \subseteq Y : \check{Y} \subseteq Z\}.$$

We will soon see that the power set of an ideal Y of P can be partitioned into tails \mathcal{T}_X where X ranges over certain subsets of Y . To this end, the following observation (Parviainen and Koivisto, 2009) is useful. It gives a partitioning of the power set of an arbitrary set.

Lemma 2 *Let X and Y be sets with $X \subseteq Y$. Let*

$$\begin{aligned} \mathcal{A} &= \{Z \subseteq Y : X \subseteq Z\}, \\ \mathcal{B} &= \{Z \subseteq Y : x \notin Z \text{ for some } x \in X\}. \end{aligned}$$

Then (i) $2^Y = \mathcal{A} \cup \mathcal{B}$ and (ii) $\mathcal{B} = \bigcup_{x \in X} 2^{Y \setminus \{x\}}$.

Proof (i) “ \supseteq ” is obvious. So, consider “ \subseteq ”: Let $Z \subseteq Y$. If $X \subseteq Z$, then $Z \in \mathcal{A}$. Otherwise, there exists $x \in X$ such that $x \notin Z$, hence $Z \in \mathcal{B}$.

(ii) By definition, $Z \in \mathcal{B}$ if and only if $Z \subseteq Y \setminus \{x\}$ for some $x \in X$. ■

We next apply this lemma iteratively to show that the power set of an ideal Y of P can be represented as the union of the tails of the subideals of Y .

Lemma 3 *Let Y be an ideal of P . Then $2^Y = \bigcup_{X \subseteq Y, X \in \mathcal{I}(P)} \mathcal{T}_X$.*

Proof By induction on the size of the ideal. (i) if $Y = \emptyset$, then $2^\emptyset = \{\emptyset\} = \mathcal{T}_Y$, and thus the claim holds. (ii) Suppose the claim holds for all ideals of size $k - 1$. Assume Y is an ideal of size k . Then, by Lemma 2, we have that $2^Y = \mathcal{T}_Y \cup \bigcup_{x \in \check{Y}} 2^{Y \setminus \{x\}}$, and by the induction assumption we have that $2^{Y \setminus \{x\}} = \bigcup_{Z \subseteq Y \setminus \{x\}, Z \in \mathcal{I}(P)} \mathcal{T}_Z$. Now, because every subideal of Y is either Y itself or a subideal of $Y \setminus \{x\}$ for some $x \in \check{Y}$, the claim holds for Y . ■

The above results construct a set cover for any power set of an ideal; these results generalize of our earlier results for parallel $1 * 1$ bucket orders (Parviainen and Koivisto, 2009). For the present problem where we are interested in summation rather than minimization, the covering property is not sufficient, but we need to show that the cover is actually a partition into disjoint parts. To this end, we next show that for any partial order P , the tails of two distinct ideals are disjoint.

Lemma 4 *Let Y and Y' be two distinct sets in $\mathcal{I}(P)$. Then the tails of Y and Y' are disjoint.*

Proof Suppose the contrary that $Z \in \mathcal{T}_Y \cap \mathcal{T}_{Y'}$. By symmetry we may assume that $Y \setminus Y'$ contains an element w . Thus $w \notin Z$, because $Z \subseteq Y'$. Because $\check{Y} \subseteq Z$, we have $w \notin \check{Y}$. By the definition of \check{Y} we conclude that for every $u \in Y \setminus \check{Y}$ there exists $v \in \check{Y}$ such that $uv \in P$. Therefore, in particular there exists $v \in \check{Y}$ such that $wv \in P$. Since $w \notin Y'$ and Y' is an ideal of P it follows by definition that $v \notin Y'$. On the other hand $v \in \check{Y}$ and $\check{Y} \subseteq Z \subseteq Y'$ implies that $v \in Y'$. Contradiction. ■

Armed with Lemmas 3 and 4 we can split the zeta transform into two nested summations, as follows. Let

$$g'(Y) = \sum_{X \in \mathcal{T}_Y} g(X), \quad Y \in \mathcal{I}(P).$$

Then, by Lemmas 3 and 4,

$$\hat{g}(Y) = \sum_{X \subseteq Y, X \in \mathcal{I}(P)} g'(X)$$

for all $Y \in \mathcal{I}(P)$. Accordingly, we will compute \hat{g} given g in two phases: first, given g we evaluate g' at all ideals of P ; second, given g' we evaluate \hat{g} at all ideals of P . The first phase is computationally straightforward as the tails \mathcal{T}_Y are disjoint, and thus the evaluation tasks are independent for each ideal Y . The second phase is less straightforward.

Because the second phase operates only over the ideals of P , there is a chance that it might be computed fast, that is, in time (and space) proportional to the number of ideals of P . Motivated by this, we call \hat{g} (restricted to the ideals of P) the *sparse zeta transform* of g' . The fast zeta transform appears not to apply directly: while it splits the transform into a sequence of n transforms in an arbitrary order, here we need to pick a particular order. To this end, let σ be a bijection from $\{1, 2, \dots, n\}$ onto N , such that if $\sigma(i)\sigma(j) \in P$, then $i < j$; we say that σ is compatible with P . A fast algorithm for the restricted zeta transform is as follows.

Algorithm 2 (the fast sparse zeta transform)

 Input: a partial order P and $g'(X)$ for $X \in \mathcal{I}(P)$.

 Output: the sparse zeta transform of g' at each $Y \in \mathcal{I}(P)$.

1. Find a bijection σ that is compatible with P .
2. For $Y \in \mathcal{I}(P)$: Let $h_0(Y) = g'(Y)$.
3. For $j = 1, \dots, n$:
 - (a) For each $Y \in \mathcal{I}(P)$:
 - If $\sigma(j) \notin \check{Y}$, let $h_j(Y) = h_{j-1}(Y)$;
 - else let $h_j(Y) = h_{j-1}(Y) + h_{j-1}(Y \setminus \{\sigma(j)\})$.
4. Return $h_n(Y)$ for $Y \in \mathcal{I}(P)$.

Lemma 5 *Algorithm 2 works correctly.*

Proof We may assume that σ is compatible with P . We show by induction on j that if Y is an ideal of P , then the computed function satisfies

$$h_j(Y) = \sum_{X \in (Y)_j} g'(X),$$

where we use the shorthand

$$(Y)_j = \{I \in \mathcal{I}(P) : Y \cap \{\sigma(j+1), \dots, \sigma(n)\} \subseteq I \subseteq Y\};$$

 This will suffice, since $(Y)_n$ consists of all the subideals of Y .

We proceed by induction on j and the size of the ideal Y . (i) Trivially $(Y)_0 = \{Y\}$. Therefore, $h_0(Y) = g'(Y) = \sum_{X \in (Y)_0} g'(X)$, as claimed. (ii) Assume then that $h_{j-1}(Y) = \sum_{X \in (Y)_{j-1}} g'(X)$.

First, consider the case $\sigma(j) \notin \check{Y}$. Now, if $\sigma(j) \notin Y$, then $(Y)_j = (Y)_{j-1}$ because $Y \cap \{\sigma(j+1), \dots, \sigma(n)\} = Y \cap \{\sigma(j), \dots, \sigma(n)\}$. Thus $h_j(Y) = h_{j-1}(Y)$, as correctly computed in step 3.

If, on the other hand, $\sigma(j) \in Y$, then $\sigma(j) \notin \check{Y}$ implies that $\sigma(j)$ is not maximal in Y . This means that there exists another maximal element $\sigma(i) \in \check{Y}$ with $\sigma(j)\sigma(i) \in P$. Because σ is compatible with P , we have $j < i$. Now suppose $(Y)_j$ contains an ideal I that is not contained in $(Y)_{j-1}$. Then it must be that $\sigma(j) \notin I$. However, $\sigma(i) \in I$ because $\sigma(i) \in Y$ and $i > j$. This is a contradiction, for $\sigma(j)\sigma(i) \in P$ and I is an ideal. Thus $(Y)_j = (Y)_{j-1}$, and so $h_j(Y) = h_{j-1}(Y)$, as correctly computed in step 3.

Second, consider the case $\sigma(j) \in \check{Y}$. Then $Y \setminus \{\sigma(j)\}$ is an ideal. To prove the correctness of step 3, it suffices, by the induction assumption, to show that $(Y)_{j-1}$ and $(Y \setminus \{\sigma(j)\})_{j-1}$ are disjoint and their union is

$(Y)_j$. To this end, it suffices to observe that $(Y)_{j-1}$ consists of all subideals $I \supseteq Y \cap \{\sigma(j+1), \dots, \sigma(j)\}$ of Y that do contain $\sigma(j)$, whereas $(Y \setminus \{\sigma(j)\})_{j-1}$ consists of all subideals $I \supseteq Y \cap \{\sigma(j+1), \dots, \sigma(j)\}$ of Y that do not contain $\sigma(j)$. ■

We analyze the time and space requirement of Algorithm 2 under the assumption that the values of the input function can be accessed in constant time. The space requirement of Algorithm 2 is clearly $O(|\mathcal{I}(P)|)$. Steps 1 and 2 are easily computed in time $O(n|\mathcal{I}(P)|)$. In step 3, at most $|\mathcal{I}(P)|$ additions and substitutions are performed in each of the n phases. In each phase, deciding whether $\sigma(j) \in \check{Y}$ takes time $O(n)$. Thus the total time requirement is $O(n|\mathcal{I}(P)|)$.

Theorem 6 *Algorithm 2 evaluates the sparse zeta transform on a partial order P in time $O(n|\mathcal{I}(P)|)$ and space $O(|\mathcal{I}(P)|)$.*

Let us now return to the evaluation of the zeta transform \hat{g} of a given function g at the ideals of a partial order P . Because the second phase of the computation takes time $O(|\mathcal{I}(P)|)$, it is interesting to know when this holds also for the first phase. In the first phase g is evaluated at every subset of N if no assumptions are made. However, if we assume that g vanishes at all sets that contain more than k elements, we find that the first phase can be computed in time $O(\sum_{i=0}^k \binom{n}{i})$. Thus the total running time is determined by the larger of the two bounds.

Theorem 7 *Suppose $g(X)$ vanishes if $|X| > k$. Then the zeta transform $\hat{g}(Y)$ can be computed for all ideals Y of a partial order P in time $O(\sum_{i=0}^k \binom{n}{i} + n|\mathcal{I}(P)|)$ and space $O(|\mathcal{I}(P)|)$.*

We end this section by considering a “dual” of the above described restricted zeta transform. Let P be a partial order on N . Let g be a function from the ideals of P to the reals. We define the *sparse up-zeta transform* of g by

$$\check{g}(Y) = \sum_{X \supseteq Y, X \in \mathcal{I}(P)} g(X), \quad Y \in \mathcal{I}(P).$$

We use this formula to define $\check{g}(S)$ also at sets S that are not ideals of P . In fact, if S is not an ideal of P , there exists a unique ideal,

$$\hat{S} = \{u \in N : uv \in P \text{ for some } v \in S\},$$

such that $\check{g}(S) = \check{g}(\hat{S})$. To see the equality, observe that if $X \supseteq S$ is an ideal of P , then necessarily $X \supseteq \hat{S}$. That said, the computational problem reduces to the computation of the sparse up-zeta transform (over the ideals only).

The fast sparse up-zeta transform is analogous to the fast sparse (down-)zeta transform. Rather than modifying the details of the construction and the proof of correctness, we make use of a “complementary symmetry.” Indeed, by denoting $\bar{X} = N \setminus X$ for any $X \subseteq N$ and $\bar{P} = \{vu : uv \in P\}$ for any partial order P on N , we see that if Y is an ideal of P , then $\check{g}(Y)$ equals $\hat{h}(\bar{Y})$, where h is defined by $h(\bar{X}) = g(X)$; note that X is an ideal of P if and only if \bar{X} is an ideal of \bar{P} .

5 POSTERIOR PROBABILITIES OF ALL ARCS

We have shown how to compute the marginal posterior probability of a given arc set. Perhaps the most important case is when the arc set contains a single arc. Indeed, it is handy to summarize the posterior distribution of DAGs by reporting the posterior probabilities for all the possible $n(n - 1)$ arcs. The straightforward way to compute these probabilities would run the algorithm separately for each each. However, there is a faster forward–backward algorithm (Koivisto, 2006) that completely avoids the multiplicative factor of $n(n - 1)$ in the running time. Next we adapt this algorithm to the presented partial order framework.

Let P be a partial order on N . Recall that we defined a “forward function” F^P that satisfies the recurrence

$$F^P(S) = \sum_{v \in S: S \setminus \{v\} \in \mathcal{I}(P)} \alpha_v(S \setminus \{v\}) F^P(S \setminus \{v\}),$$

where S is an ideal of P . Analogously, we define a “backward function” B^P by

$$B^P(T) = \sum_{v \in T: T \setminus \{v\} \in \mathcal{I}(P)} \alpha_v(N \setminus T \setminus \{v\}) B^P(T \setminus \{v\}),$$

where T is an ideal of P . Then we combine the forward and backward functions into

$$\gamma_v^P(A_v) = \sum_S q_v(S) F^P(S) B^P(N \setminus S \setminus \{v\}),$$

where A_v is a subset of $N \setminus \{v\}$ and S runs over all ideals of P with $A_v \subseteq S \subseteq N \setminus \{v\}$.

We now express the marginal posterior probability of an arc uv as a weighted average of the terms $\gamma_v^P(A_v)$. To this end, observe that

$$\sum_{P \subseteq L} \prod_{v \in N} \alpha_v(L_v) = \sum_{A_v \subseteq N \setminus \{v\}} \beta_v(A_v) \gamma_v^P(A_v).$$

Thus, if \mathcal{P} is a family of partial orders that exactly covers the linear orders on N , then, by (1), the joint

probability of the data D and a feature f can be written, as

$$p(D, f) = \sum_{P \in \mathcal{P}} \sum_{A_v \subseteq N \setminus \{v\}} \beta_v(A_v) \gamma_v^P(A_v)$$

Now we make use of the assumption that f is the indicator of the edge uv . Under this assumption, the choice of the arc uv affect only the term $\beta_v(A_v)$ and thus the terms $\gamma_v^P(A_v)$ can be precomputed assuming the trivial feature $f \equiv 1$.

It remains to show that the values $\gamma_v^P(A_v)$ can be computed fast. First, note that for a fixed partial order P , the values $\alpha_v(S \setminus \{v\})$ for all nodes v and ideals $S \setminus \{v\}$ of P can be computed using the fast sparse zeta transform in time $O(n^2 |\mathcal{I}(P)|)$. Second, note that the forward and backward functions can be computed by a straightforward recursion in time $O(n |\mathcal{I}(P)|)$. Third, the values $\gamma_v^P(S)$ for all nodes v and ideals $S \subseteq N \setminus \{v\}$ can be computed by the fast sparse up-zeta transform in time $O(n^2 |\mathcal{I}(P)|)$. Finally, given these values for the ideals, the values $\gamma_v^P(A_v)$ for all nodes v and parent sets A_v can be found in time $O(\sum_{i=0}^k \binom{n}{i})$ assuming $|A_k| \leq k$. Clearly all these steps can be computed in space $O(n |\mathcal{I}(P)|)$.

Theorem 8 *Let \mathcal{P} be a family of partial orders that exactly covers the linear orders on N . Then the posterior probabilities of all arcs of a Bayesian network on N can be computed in time $O(n^2 \sum_{P \in \mathcal{P}} |\mathcal{I}(P)|)$ and space $O(n \max_{P \in \mathcal{P}} |\mathcal{I}(P)|)$.*

In other words, the posterior probabilities of all the $n(n - 1)$ arcs can be computed in essentially the same time and space than the posterior probability of a single feature (see Theorem 1).

Acknowledgements

This research was supported in part by the Academy of Finland, Grant 125637 (M.K.).

References

- A. Björklund, T. Husfeldt, P. Kaski, and M. Koivisto. Fourier meets Möbius: fast subset convolution. In *Proceedings of the 39th ACM Symposium on Theory of Computing (STOC)*, pages 67–74. ACM Press, 2007.
- A. Björklund, T. Husfeldt, P. Kaski, and M. Koivisto. Trimmed Moebius inversion and graphs of bounded degree. In *Proceedings of the 25th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 85–96. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, 2008.

- G. F. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–347, 1992.
- N. Friedman and D. Koller. Being Bayesian about network structure: A Bayesian approach to structure discovery in Bayesian networks. *Machine Learning*, 50(1–2):95–125, 2003.
- D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, 1995.
- R. Kennes. Computational aspects of the Möbius transformation of graphs. *IEEE Transaction on Systems, Man, and Cybernetics*, 22(2):201–223, 1992.
- M. Koivisto. Advances in exact Bayesian structure discovery in Bayesian networks. In *Proceedings of the 22th Conference on Uncertainty in Artificial Intelligence (UAI 2006)*, 2006.
- M. Koivisto and P. Parviainen. A space–time trade-off for permutation problems. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA 2010)*, 2010.
- M. Koivisto and K. Sood. Exact Bayesian structure discovery in Bayesian networks. *Journal of Machine Learning Research*, 5:549–573, 2004.
- J. Nederlof. Fast polynomial-space algorithms using möbius inversion: Improving on steiner tree and related problems. In S. Albers, A. Marchetti-Spaccamela, Y. Matias, S. E. Nikolettseas, and W. Thomas, editors, *ICALP (1)*, volume 5555 of *Lecture Notes in Computer Science*, pages 713–725. Springer, 2009. ISBN 978-3-642-02926-4.
- S. Ott and S. Miyano. Finding optimal gene networks using biological constraints. *Genome Informatics*, 14:124–133, 2003.
- P. Parviainen and M. Koivisto. Exact structure discovery in Bayesian networks with less space. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI 2009)*, 2009.
- T. Silander and P. Myllymäki. A simple approach for finding the globally optimal Bayesian network structure. In *Proceedings of the 22th Conference on Uncertainty in Artificial Intelligence (UAI 2006)*, 2006.
- J. Tian and R. He. Computing posterior probabilities of structural features in Bayesian networks. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI 2009)*, 2009.
- F. Yates. The design and analysis of factorial experiments. *Harpden: Imperial Bureau of Soil Science Technical Communication*, 35, 1937.