2nd International Symposium on Imprecise Probabilities and Their Applications, Ithaca, New York, 2001

Robust discovery of tree-dependency structures

Marco Zaffalon
IDSIA—Istituto Dalle Molle di Studi sull'Intelligenza Artificiale
Galleria 2, CH-6928 Manno, Switzerland
zaffalon@idsia.ch

Abstract

The problem of inferring dependency structures from random samples is a very fundamental topic in artificial intelligence and statistics. This paper reviews an early result from Chow and Liu on the approximation of unknown multinomial distributions by tree-dependency distributions, at the light of imprecise probabilities. Imprecision, arising here from Walley's imprecise Dirichlet model, generally makes many tree structures be plausible given the data. This paper focuses on the inference of the substructure common to all the possible trees. Such common pattern is a set of reliable dependencies. The problem of identifying the common pattern is abstracted and solved here in the general context of graph algorithms. On this basis, an algorithm is developed that infers reliable dependencies in time $O(k^3)$, from a set of k binary random variables, that converge to a tree as the sample grows. The algorithm works by computing bounds on the solutions of global optimization problems. There are a number of reasons why trees are a very important special case of dependence graphs. This work appears as a significant step in the direction of discovering dependency structures under the realistic assumption of imprecise knowledge.

1 Introduction

This paper deals with the following problem. We are given a random sample of N observations, which are jointly categorized according to a set of k binary variables X, Y, Z, etc. The dependency between two variables is measured by the information-theoretic symmetric index called *mutual information* [13]. If the chances $\boldsymbol{\theta}$ of all instances defined by the co-occurrence of $X = x, Y = y, Z = z$, etc., were known, it would be possible to approximate the distribution by another, for which all the dependences are bivariate and can graphically be represented as an undirected tree T, that is the optimal approximating tree-dependency distribution. This result is due to Chow and Liu [4], who use Kullback-Leiber's divergence [14] as a measure of the closeness of two distributions.

Since only a sample is available, the joint distribution $\boldsymbol{\theta}$ is unknown and an inferential approach is necessary. Prior uncertainty about the vector $\boldsymbol{\theta}$ is described by the *imprecise Dirichlet model* (IDM) [20], which results in posterior uncertainty about $\boldsymbol{\theta}$, the mutual information and the tree T. It follows that a set S of plausible trees is generally consistent with the evidence. The paper aims at making inferences about T, more precisely to identify which set of edges belongs to T. This can be realized by identifying the set of edges common to all the trees in S, which is called the *common pattern*.

The present work describes Chow and Liu's approach in more detail in Section 2, where the basic issues to be addressed for the extension are also discussed. The rest of the paper is organized in two conceptually disjoint parts. Sections 3 and 4 abstract the problem of determining the common pattern and solve it in the general context of graph algorithms. This part is independent on the specific application of recovering reliable dependences and aims at determining the common edges of a set of graphs. The following part, in Sections 5 and 6, develops tools to compute approximated lower and upper expectations of mutual information under the IDM, by formulating the computation as problems of global optimization. When these values are used together with the preceding graph algorithm, the overall procedure infers, in time $O(k^3)$, a set of reliable dependences that belong to the common pattern, as explained in Section 2. As the sample grows, these dependences converge to a tree.

There are many reasons to focus on the special case of tree-dependency structures, apart from the optimality of Chow and Liu's method. Trees (in their directed version, see Section 2) have been shown to provide a very good approximation to the more general structures of polytrees [6], whose discovery is NPhard, whereas Chow and Liu's algorithm works in time $O(k^2)$. On the application side, trees seem to be on the edge of models with a very good balance of representational power and ease of inference, with many successful applications. For example, trees constitute the underlying structure of the tree-augmented naive classifiers, which are the state-of-the-art classification models in the machine learning literature [7], and of mixture-of-trees, another promising classification model [16]. Also, tree-recovery algorithms are the basis of algorithms for the discovery of more complex structures [3].

However, it is important to infer tree dependences that are robust. Focusing on the common pattern is a way to do that. Robustness is needed because dependency structures are very fundamental to synthesize a domain and to improve the effectiveness of the related models, which can take advantage of the sparseness of the dependence relations. To my knowledge, literature only reports two other attempts to infer robust structures of dependence. Bernard [1] describes a method to build, from a multivariate binary database, a directed graph, the arcs of which are interpreted as implications. The method develops an inductive step based on the IDM and is based on Bayesian implicative analysis. It is not immediately clear whether the graphical structure can be directly interpreted as a set of probabilistic dependences or not. The second approach is due to Kleiter [10]. The author uses confidence intervals on the mutual information to robustly measure the degree of dependence between random variables. In this case, unlike in Bernard's proposal, imprecision is neglected. Most of other methods are descriptive in nature and hence are not robust.

2 Technical introduction

In this paper a tree is an undirected connected graph with k nodes and $k-1$ edges. Trees are acyclic graphs, i.e. they do not admit closed paths (see [18], Proposition 2). When a tree is regarded as a dependency structure, nodes are interpreted as random variables and edges as symmetric dependences between the connected variables. Note that trees can also be used as models of asymmetric dependences; it is sufficient to arbitrarily orient the arrows of the undirected tree, in a way that each node has a single predecessor at most. In fact, all the directed trees that share the same undirected structure are equivalent models of dependence [19].

Chow and Liu's original algorithm works by computing, from the sample, the descriptive values of mutual information (see Section 5.1) between pairs of random variables. These values are used as weights for the edges in a fully connected graph. The sought graph is then the tree for which the sum of the weights of its edges is maximum. (There can be several maximum trees, if there are equal weights on different edges, among which the choice is arbitrary.) In the literature of mathematical programming, the general version of the last problem is called the *maximum spanning tree* ([18], p. 271). Its construction takes $O(k^2)$ time. This is also the computational complexity of Chow and Liu's procedure.

By adopting the IDM as inferential tool, we obtain lower and upper expectations of mutual information. This fact makes a big difference as far as the original procedure of Chow and Liu is concerned: the maximum spanning tree problem assumes that the edge weights satisfy a relationship of total order. Now we can only rely on a partial order, i.e. not all the pairs of edges can be compared. Two basic questions follow from such considerations: (i) what should the generalization of the maximum spanning tree problem be, when only a partial order on the edge weights is available? (ii) By which method should the weights be computed and compared by the IDM?

Note that question (i) is formulated independently on the specific application and so it is addressed in the general case in Sections 3 and 4. Sections 5 and 6 address question (ii). The result of the latter part produces a partial order on the edges that can directly be used to feed the algorithm provided in the former sections, in a way that the overall procedure infers reliable dependences.

3 Identifying the common pattern

Let $G =< V, E >$ be a connected undirected graph with $k = |V|$ nodes, where $E \subseteq V \times V$ is the set of edges; and $(v, v) \notin E$ for each $v \in V$. I assume that each edge is associated with a set of real numbers called *weights*. For example, a set of weights might be specified as an interval of the real line. (The interpretation of trees as dependency structures is not needed in this and in the following section; here the problem is abstracted from the specific application, on which the attention will be focused later.)

Under these conditions, we obtain a partial ordered on the edges. In the following we say that an edge e is greater than another (or *dominates*), e', if it is not possible that a weight of e' is greater than or equal to a weight of e. I also assume that all the total orders consistent with the given partial order are admissible: i.e. for each total order extending the partial order, there are weights, one for each edge, in the related sets whose order relationship is the given total order.

For each extension of the partial order to a total order, there is a unique maximum spanning tree T of G (this follows from Kruskal's algorithm that only needs the total order on the edges to build the maximum spanning tree [11]). T is a connected graph, spanning all the nodes of G, with a set of $k-1$ edges that is maximum according to the total order. More precisely, a tree is maximum when the sum of its edge weights is maximum for any choice of weights in the related sets that is consistent with the total order.

Let S be the set of maximum spanning trees T obtained when all the possible extensions of the partial order to a total order are considered. Define the *common pattern* of G as the graph whose edges are the elements of E that are common to all the trees in S. The common pattern can be identified by the following result.

Theorem 1 *An edge e of G is in the common pattern iff in each cycle that contains e there is an edge e' dominated by e.*

Proof.

($\Leftarrow$) By contradiction, assume that there is a tree T in S that does not contain e. By adding e to T we create a cycle ([18], Proposition 2). By hypothesis, in such a cycle there must exist an edge e' dominated by e. Removing e', we obtain a new tree that improves upon T, which then cannot be optimal.

($\Rightarrow$) By contradiction, assume that there is a cycle C in G where e does not dominate any edge. So we can consider a total order, extending the partial order on the edges of G, for which e is less than any other edge in C. Call T the related tree. By removing e from T we create two subtrees, say T' and T''. One of these can possibly be a degenerate tree composed by a single node. Now consider that there must be an edge e_C of C that connects T' and T''. In fact, if there was not, C could not be a cycle passing through e. The graph composed by T', T'' and e_C has $k-1$ edges, spans all the nodes of G, and therefore it is a tree ([18], Proposition 2). It improves upon T, because e is less than e_C, so that T cannot be optimal. ■

Let us stress that the maximum spanning tree problem with set-based (or imprecise) weights, as presented here, does not seem to have been addressed in the literature of graph algorithms, also if it is possible to find a variant of the spanning tree problem obtained by considering fuzzy weights [18, 8, 2].

4 Pattern detection algorithms

Theorem 1 directly leads to a procedure that determines whether a given edge e is in the common pattern or not. It suffices to consider the graph G' obtained by removing the edges dominated by e from G. The edge e is in the common pattern iff there is no cycle in G' that contains e. Unfortunately, testing the last condition demands $O(k^2)$ time and, by repeating it for all the edges $e \in E$, the overall procedure requires $O(k^4)$ steps. We can reduce the computational complexity to $O(k^3)$ by requiring that only a subset of edges in the common pattern be determined, as follows.

Consider algorithm 2, outlined in a pseudo programming language in the points 1–5 below. It takes as input a fully connected graph $G =< V, E >$. In the algorithm, a tree with a number of nodes in $\{2, \ldots, k-1\}$ is called *subtree.*

Algorithm 2

```
1. LET $CP = \emptyset$.
2. FOR all $v \in V$
   (a) IF there is a node $v' \in V$ such that $(v, v') \notin CP$ and it dominates $(v, v'')$ for each $v'' \in V$, $v'' \neq v'$ THEN
       i. add $(v, v')$ to $CP$.
   (b) END IF.
3. END FOR.
4. IF there is a subtree in $CP$ THEN
   (a) make it the current subtree.
   (b) Consider the set of edges $E' \subseteq E$ with one endpoint in the nodes of the current subtree and the other outside.
   (c) IF there is an edge $e' \in E'$ that dominates all the other edges in $E'$ THEN
       i. add $e'$ to $CP$ and to the current subtree.
       ii. GO TO 4b.
   (d) ELSE
       i. IF there is another subtree in $CP$ not considered yet THEN
          A. GO TO 4a.
       ii. ELSE output $CP$.
       iii. END IF.
   (e) END IF.
5. END IF.
```

The following proposition shows that algorithm 2 is sound.

Proposition 3 *CP is a subset of the edges of the common pattern of G.*

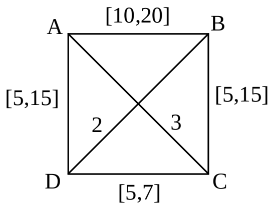

Figure 1: A graph with vertices {A,B,C,D} whose edge weights can be intervals. The partial order on the edges induced by the intervals permits to decide that the common pattern is {(A,B)}.

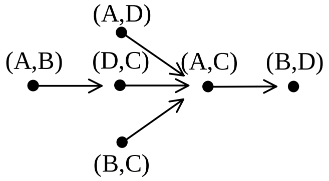

Figure 2: The partial order on the edges of the graph in the preceding figure. Here an arrow from e to e' means that e dominates e'.

Proof.

Consider the first possible insertion in step 2(a)i. The cycles that encompass (v, v') must pass through the set of edges $\{(v, v'') : v'' \in V, v'' \neq v'\}$. Since (v, v') dominates all the edges in the preceding set, for each cycle passing through (v, v') there is an edge in the cycle that is dominated by (v, v'), so that (v, v') is in the common pattern, by Theorem 1.

The algorithm can insert an edge in CP also in step 4(c)i. Recall that each subtree is a connected acyclic graph. It is clear that any cycle that contains e' must pass through an edge e'' that has one endpoint in the nodes of the subtree and the other outside. But e' dominates e'' by step 4c. This holds for all the cycles, so e' is in the common pattern by Theorem 1. ■

As outlined at the start of the section, algorithm 2 is not complete, i.e. it does not generally identify all the edges in the common pattern. For example, consider the graph in Figure 1. The graph is made by four vertices: A, B, C and D. The sets of weights are expressed as intervals, e.g., the set of the edge (A,B) is given by the interval $[10, 20] \subset \mathbb{R}$; the set of (A,C) is $\{3\}$. The imprecise specification of the weights originates the partial order, by defining that the interval $[\alpha, \beta]$ dominates $[\alpha', \beta']$ iff $\alpha > \beta'$, given in Fig. 2.

By Theorem 1, we obtain that only (A,B) is in the common pattern; but algorithm 2 cannot determine this fact.

Thus, algorithm 2 is the result of a trade-off between computational complexity (that is $O(k^3)$, as shown in the next section) and the capability to fully detect the common pattern. This choice does not seem critical to the specific extent of discovering tree-dependency structures. In fact, if the edges are totally ordered, algorithm 2 produces the maximum spanning tree. Since the knowledge of the mutual information increases with the sample size, eventually determining a total order on the edges (if edges with *approximately* equal weights are ordered arbitrarily), CP is anyway guaranteed to converge to the sought tree. Of course, the algorithm must be tested in practice to understand, in common situations, how many edges are prevented from being recovered by the above limitation.

4.1 Computational complexity

The assumption behind the following analysis is that the comparison of two edges can be done in constant time. In this case, given a set E' of edges, there is a procedure that determines in time $O(|E'|)$ if there is an edge $e' \in E'$ that dominates all the others. The first step of the procedure selects an edge that is candidate to be dominant. This is made by doing pairwise comparisons of edges and by always discarding the non-dominant edge (or edges) in the comparison. After at most $|E'| - 1$ comparisons, it is possible to know whether there is a candidate or not. If there is, the second step of the procedure compares such candidate e' with all the other edges, so deciding if e' dominates all the others. This requires $|E'| - 1$ comparisons. The two steps of the procedure then take $O(|E'|)$ time.

Let us now focus on algorithm 2. The loop 2–3 is repeated $k = |V|$ times. Each time the test 2a decides whether there is a dominant edge out of $k - 1$ edges (each node is connected to all the others). By the previous result, such task takes $O(k)$ time. Then the loop requires $O(k^2)$ time.

Now consider the two nested loops made by the instructions 4a, 4b, 4(c)ii, and 4(d)iA. Each time the instruction of jump 4(c)ii is executed, a new edge has been added to CP. Each time 4(d)iA is executed, a new subtree is considered. Since CP can have $k - 1$ edges at most and k is also an upper bound on the number of different subtrees, the two loops can jointly require $2k - 1$ iterations at most. Each such iteration executes the test 4c. By using k^2 as an upper bound on $|E'|$, we need $O(k^2)$ time to detect whether the dominant edge exists. So that the overall time required by the loops is $O(k^3)$. This is also the computational complexity of the entire procedure.

5 The expected value of mutual information under the IDM

Algorithm 2 is based on the comparison of edge weights. In the specific application, the edge weights measure the mutual information between pairs of random variables. The following subsections define the mutual information and show that under the IDM the partial knowledge arising from a finite sample can only provide us with a set of possible expectations of the mutual information. Section 6 will then exploit the values in such a set to define a method to partially compare the edges of the graph.

5.1 Definition of mutual information

Let X and Y be discrete random variables taking values in the finite sets $\mathcal{X}$ and $\mathcal{Y}$, respectively. The generic elements of these sets will be denoted by x and y. Assume that the joint distribution of (X, Y) is multinomial with unknown chances θ_{xy}. Define the *mutual information* [5] between X and Y as the real number

$$MI(X,Y) = \sum_{x,y} \left(\theta_{xy} \ln \frac{\theta_{xy}}{\theta_{x.}\theta_{.y}} \right), \tag{1}$$

where $\theta_{x.} = \sum_y \theta_{xy}$ and $\theta_{.y} = \sum_x \theta_{xy}$. Expression (1) can be applied also when some chances are zero, if we take $\theta_{xy} \ln \frac{\theta_{xy}}{\theta_{x.}\theta_{.y}} = 0$ when $\theta_{xy} = 0$, as it is natural since $\lim_{\theta_{xy} \to 0} \theta_{xy} \ln \frac{\theta_{xy}}{\theta_{x.}\theta_{.y}} = 0$.

The mutual information is a non-negative information-theoretic measure that is frequently used, especially in the artificial intelligence field, as a degree of dependence between random variables. It is zero iff the variables are independent.

5.2 Inferences about mutual information

The IDM models uncertainty about the unknown chances of a multinomial distribution by a set of prior densities [20]. These are combined with the likelihood function to produce the following posterior density function:

$$\pi(\boldsymbol{\theta}|\mathbf{n}_{XY}) \propto \prod_{x,y} \theta_{xy}^{n_{xy}+st_{xy}-1}. \tag{2}$$

Here $\boldsymbol{\theta}$ denotes the vector of the unknown chances and $\mathbf{n}_{XY}$ the vector of the data counts. The generic element of $\mathbf{n}_{XY}$ is the observed frequency of (x, y), i.e. n_{xy}. The hyperparameter s is a positive real number, representing a degree of caution of the inferences, that Walley suggests to choose in the interval $[1, 2]$. The hyperparameters t_{xy} take on all the possible values in the open unit simplex defined by the constraints:

$$\sum_{x,y} t_{xy} = 1 \tag{3}$$

$$t_{xy} > 0 \quad \forall x, y. \tag{4}$$

The vector of the t-hyperparameters is denoted by $\mathbf{t}_{XY}$.

Inferences by the IDM on the mutual information of two variables can be made by considering the expected value of the mutual information with respect to $\pi(\boldsymbol{\theta}|\mathbf{n}_{XY})$. Unfortunately, to the best of my knowledge, it is not known either a closed form of the expected value of the mutual information under a Dirichlet density ([17], p. 25), or a bounded-error approximation. Kleiter addresses this problem by providing a chi-square approximation to the distribution of the mutual information, based on empirical considerations [10]. The expected value that he proposes is similar to the one I give below. The expected value of the mutual information is approximated by replacing the multinomial chances in (1) by their expected values under a Dirichlet posterior:

$$\mu(X, Y|\mathbf{n}_{XY}, \mathbf{t}_{XY}) = \sum_{x,y} \left(\widehat{\theta}_{xy} \ln \frac{\widehat{\theta}_{xy}}{\widehat{\theta}_{x.}\widehat{\theta}_{.y}} \right), \tag{5}$$

where $\widehat{\theta}_{xy} = E[\theta_{xy}|\mathbf{n}_{XY}, \mathbf{t}_{XY}] = \frac{n_{xy}+st_{xy}}{N+s}$, N being the number of units in the random sample, and $\widehat{\theta}_{x.} = E[\theta_{x.}|\mathbf{n}_{XY}, \mathbf{t}_{XY}] = \sum_y \widehat{\theta}_{xy}$, $\widehat{\theta}_{.y} = E[\theta_{.y}|\mathbf{n}_{XY}, \mathbf{t}_{XY}] = \sum_x \widehat{\theta}_{xy}$. Note that each instance of the t-hyperparameters produces a possible value of the approximate expected value of the mutual information.

6 Comparing edges

The edges in the graph are compared on the basis of expression (5). Section 6.1 addresses the case of the comparison of two edges that share a node. Section 6.2 deals with the case of two edges without a common node. Both cases reduce to coping with non-linear global optimization problems. These do not appear easy to be tackled exactly, so I propose to compute lower and upper bounds. The case of binary random variables is fully developed by providing a constant-time procedure to compute such bounds.

6.1 One node in common

Let us focus on three generic random variables: X, Y, Z. We want to compare the edge (X, Y) with the edge (Y, Z). Under complete knowledge of the chances of the multinomial distribution, the pair of variables (X, Y) should be preferred to (Y, Z) if $MI(X, Y) >$

$MI(Y,Z)$. Since we have only a partial knowledge of the mutual information, and with reference to the proposed approximation, we say that (X,Y) dominates (Z,Y) if the inequality $\mu(X,Y|\mathbf{n}_{XY},\mathbf{t}_{XY}) > \mu(Z,Y|\mathbf{n}_{ZY},\mathbf{t}_{ZY})$ holds for all the possible values of the vectors $\mathbf{t}_{XY}$ and $\mathbf{t}_{ZY}$. (Observe that the criterion only produces a partial order, since it may be the case that (X,Y) and (Y,Z) are mutually undominated.) This is equivalent to solving the following optimization problem:

$$\inf\left[\mu(X,Y|\mathbf{n}_{XY},\mathbf{t}_{XY}) - \mu(Z,Y|\mathbf{n}_{ZY},\mathbf{t}_{ZY})\right] \tag{6}$$

$$\sum_{x,y} t_{xy} = 1 \tag{7}$$

$$\sum_{z,y} t_{zy} = 1 \tag{8}$$

$$\sum_{x} t_{xy} = \sum_{z} t_{zy} \quad \forall y \tag{9}$$

$$t_{xy}, t_{zy} > 0 \qquad\qquad \forall x,y,z. \tag{10}$$

If the optimal value is positive, the dominance holds. Note the constraints (9), which maintain the consistency between the vectors $\mathbf{t}_{XY}$ and $\mathbf{t}_{ZY}$ that are logically dependent because of Y.

Let us rewrite $\mu(X,Y|\mathbf{n}_{XY},\mathbf{t}_{XY})$ by using (5):

$$\begin{aligned}
&\mu(X,Y|\mathbf{n}_{XY},\mathbf{t}_{XY})\\
=\ &\sum_{x,y}\left(\frac{n_{xy}+st_{xy}}{N+s}\ln\frac{\frac{n_{xy}+st_{xy}}{N+s}}{\frac{n_{x.}+st_{x.}}{N+s}\,\frac{n_{.y}+st_{.y}}{N+s}}\right)\\
=\ &\frac{1}{N+s}\sum_{x,y}\left[(n_{xy}+st_{xy})\ln\frac{n_{xy}+st_{xy}}{n_{x.}+st_{x.}}\right]+\\
&-\frac{1}{N+s}\sum_{y}\left[(n_{.y}+st_{.y})\ln\frac{n_{.y}+st_{.y}}{N+s}\right],
\end{aligned}$$

with the obvious meanings for $n_{x.}$, $t_{x.}$, $n_{.y}$ and $t_{.y}$. By rewriting $\mu(Z,Y|\mathbf{n}_{ZY},\mathbf{t}_{ZY})$ analogously, and by defining

$$f(\mathbf{t}_{XY}) = \sum_{x,y}\left[(n_{xy}+st_{xy})\ln\frac{n_{xy}+st_{xy}}{n_{x.}+st_{x.}}\right], \tag{11}$$

$$f(\mathbf{t}_{ZY}) = \sum_{z,y}\left[(n_{zy}+st_{zy})\ln\frac{n_{zy}+st_{zy}}{n_{z.}+st_{z.}}\right], \tag{12}$$

we obtain the new form of the objective function (i.e., the function to optimize):

$$\begin{aligned}
&\mu(X,Y|\mathbf{n}_{XY},\mathbf{t}_{XY}) - \mu(Z,Y|\mathbf{n}_{ZY},\mathbf{t}_{ZY})\\
=\ &\frac{1}{N+s}f(\mathbf{t}_{XY}) - \frac{1}{N+s}f(\mathbf{t}_{ZY}).
\end{aligned}$$

In the subsequent development, the constraints (9) will be relaxed in order to simplify the problem that otherwise does not seem easy to solve. The weaker version of the problem can be decomposed in two problems that, respectively, compute the infimum of $\frac{1}{N+s}f(\mathbf{t}_{XY})$ and the supremum of $\frac{1}{N+s}f(\mathbf{t}_{ZY})$. The overall optimum in the larger domain is the difference of these two values. Of course, this will generally be a lower bound on the optimal value of the original problem. The bound is originated by the assumption, implied by the relaxation of the constraints (9), that knowing the mutual information on one edge does not constrain the possible values of mutual information on the other, which is generally false.

6.1.1 Optimization of $f(\cdot)$

The previous arguments allow us to focus on the following problem (the maximization of $f(\mathbf{t}_{ZY})$ is analogous):

$$\min f(\mathbf{t}_{XY}) \tag{13}$$

$$\sum_{x,y} t_{xy} = 1 \tag{14}$$

$$t_{xy} \geq 0 \qquad\qquad \forall x,y. \tag{15}$$

Note that the variables are allowed to be zero. This is possible by extending $f(\cdot)$ to the closed domain, by an analogous observation to that in Section 5.1: the generic term of the sum in (11), i.e. $(n_{xy}+st_{xy})\ln\frac{n_{xy}+st_{xy}}{n_{x.}+st_{x.}}$, is defined to be zero when $n_{xy}+st_{xy}=0$.

Geometrically, the constraints of the problem define a polytope. This is composed of the inner part, where all the variables are positive, and the border, where at least one variable is zero. The border is made by a set of faces: each face can be identified by the particular set of variables that are equal to zero on it. In the following, the task of searching for the global minimum in the polytope will be split into the subtasks of searching the global minima of $f(\cdot)$ over the inner part of the polytope and its faces. Each time $f(\cdot)$ will be considered as function of the non-zero variables only.

I address such global optimization problems by applying Karush-Kuhn-Tucker's first-order necessary conditions [12]. These allow us to restrict the set of points that contain the minimum, by constraining the gradient of $f(\cdot)$. Let us consider the partial derivative of $f(\cdot)$ with respect to a generic variable t_{xy}, which is positive over the considered portion of the domain. Define the set $\Phi = \{(x',y') \in \mathcal{X}\times\mathcal{Y}\,|\,n_{x'y'}+st_{x'y'} > 0\}$. The function $f(\cdot)$ is rewritten as the sum of three terms:

$$(n_{xy}+st_{xy})\ln\frac{n_{xy}+st_{xy}}{n_{x.}+st_{x.}} \tag*{(f_1)}$$

$$+\sum_{(x,y')\in\Phi,\,y'\neq y}\left[(n_{xy'}+st_{xy'})\ln\frac{n_{xy'}+st_{xy'}}{n_{x.}+st_{x.}}\right] \tag*{(f_2)}$$

$$+\sum_{(x',y')\in\Phi,x'\neq x,}\left[(n_{x'y}+st_{x'y})\ln\frac{n_{x'y}+st_{x'y}}{n_{x'.}+st_{x'.}}\right](f_3)$$

recalling that for a generic $x'\in\mathcal{X}$, $t_{x'.}=\sum_{y'\in\mathcal{Y}}t_{x'y'}$. We have immediately that $\partial f_3(\mathbf{t}_{XY})/\partial t_{xy}=0$. We have also

$$\frac{\partial f_1(\mathbf{t}_{XY})}{\partial t_{xy}}=s\left(1-\frac{n_{xy}+st_{xy}}{n_{x.}+st_{x.}}+\ln\frac{n_{xy}+st_{xy}}{n_{x.}+st_{x.}}\right)$$

and

$$\begin{aligned}\frac{\partial f_2(\mathbf{t}_{XY})}{\partial t_{xy}} &= -\frac{s}{n_{x.}+st_{x.}}\sum_{y'\neq y}(n_{xy'}+st_{xy'})\\ &= s\frac{n_{x.}+st_{x.}-n_{xy}-st_{xy}}{n_{x.}+st_{x.}}\\ &= s\left(-1+\frac{n_{xy}+st_{xy}}{n_{x.}+st_{x.}}\right).\end{aligned}$$

Finally, we obtain

$$\frac{\partial f(\mathbf{t}_{XY})}{\partial t_{xy}}=s\ln\frac{n_{xy}+st_{xy}}{n_{x.}+st_{x.}}=s\ln\frac{\widehat{\theta}_{xy}}{\widehat{\theta}_{x.}}. \tag{16}$$

The first-order necessary condition for a problem with linear constraints of equality states that for a point $\mathbf{t}_{XY}$ to be a local optimum there must exist a vector λ of real numbers such that $\nabla f(\mathbf{t}_{XY})=A^T\lambda$, where A^T is the transposed constraints matrix ([15], pp. 314–316). In the current problem, the matrix is made by the unique equality constraint (14), so $A=[1,1,\ldots,1]$ and $\lambda\in\mathbb{R}$. The necessary condition then becomes:

$$\frac{\partial f(\mathbf{t}_{XY})}{\partial t_{xy}}=\lambda\quad\forall(x,y):t_{xy}>0.$$

That is, in order for $\mathbf{t}_{XY}$ to be a local optimum, all the partial derivatives of the function must have the same value in such point. By (16), this is:

$$\widehat{\theta}_{xy}=\alpha\widehat{\theta}_{x.}\quad\forall(x,y):t_{xy}>0, \tag{17}$$

where $\alpha\in\mathbb{R}$ is a constant. The value of α can be determined as follows. First, note that $\alpha\neq0$ otherwise it would be $t_{xy}=0$. Second, consider that $\sum_y\widehat{\theta}_{xy}=\widehat{\theta}_{x.}$ for all $x\in\mathcal{X}$. This holds iff $\sum_{y:t_{xy}>0}\widehat{\theta}_{xy}+\sum_{y:t_{xy}=0}\widehat{\theta}_{xy}=\widehat{\theta}_{x.}$, which, by (17), holds iff $\alpha\widehat{\theta}_{x.}\nu_x+\gamma_x=\widehat{\theta}_{x.}$. Here γ_x is equal to $\sum_{y:t_{xy}=0}\widehat{\theta}_{xy}$ and ν_x is a constant denoting the number of elements y of $\mathcal{Y}$ for which $t_{xy}>0$. γ_x is a constant for the problem, too, because it does not depend on any positive variable t_{xy}. We obtain

$$\alpha=\frac{\widehat{\theta}_{x.}-\gamma_x}{\nu_x\widehat{\theta}_{x.}}\quad\forall x:\exists y:t_{xy}>0 \tag{18}$$

6.1.2 The case of binary random variables

This section applies (18) and (17) to derive a procedure that solves problem (13) with constraints (14) and (15) when the random variables are binary, i.e.: $\mathcal{X}=\{x_1,x_2\}$ and $\mathcal{Y}=\{y_1,y_2\}$. In this case there are 4 t-variables in the problem: $t_{x_1y_1}$, $t_{x_1y_2}$, $t_{x_2y_1}$ and $t_{x_2y_2}$; and there are $2^4-1=15$ different ways to assign zeroes to them—excluding the assignment of 4 zeroes that is not allowed by (14), but including the assignment of no zero. Therefore the domain of the optimization problem is a polytope that can be represented as the partition made by the 14 faces and the inner part.

Below the necessary condition of the first order is specialized on the elements of the partition. Each element is identified by the set of variables that are positive on it (indicated in parentheses).

- Face 1 (the positive variables are $t_{x_1y_1}$, $t_{x_1y_2}$ and $t_{x_2y_1}$; in this case we have $\nu_{x_1}=2$, $\nu_{x_2}=1$ and $\gamma_{x_1}=0$, by definition). α is a constant for all $x\in\mathcal{X}$ for which there exists $y\in\mathcal{Y}$ such that $t_{xy}>0$. In the particular case, this implies the equality $\left(\widehat{\theta}_{x_1.}-\gamma_{x_1}\right)/\left(\nu_{x_1}\widehat{\theta}_{x_1.}\right)=\left(\widehat{\theta}_{x_2.}-\gamma_{x_2}\right)/\left(\nu_{x_2}\widehat{\theta}_{x_2.}\right)$, i.e. $\widehat{\theta}_{x_1.}/\left(2\widehat{\theta}_{x_1.}\right)=\left(\widehat{\theta}_{x_2.}-\gamma_{x_2}\right)/\widehat{\theta}_{x_2.}$, which holds iff $\widehat{\theta}_{x_1.}=1-2\gamma_{x_2}$. Note that since $\widehat{\theta}_{x_1.}$ and $\widehat{\theta}_{x_2.}$ are positive on the current face, the cases $\gamma_{x_2}=0$ and $\gamma_{x_2}=1/2$ are not allowed.
- Face 2 ($t_{x_1y_1},t_{x_1y_2},t_{x_2y_2}$; $\nu_{x_1}=2,\nu_{x_2}=1,\gamma_{x_1}=0$). Analogously to the preceding case, $\widehat{\theta}_{x_1.}=1-2\gamma_{x_2}$, $\gamma_{x_2}\neq0$ and $\gamma_{x_2}\neq1/2$.
- Face 3 ($t_{x_1y_1},t_{x_2y_1},t_{x_2y_2}$; $\nu_{x_1}=1,\nu_{x_2}=2,\gamma_{x_2}=0$). By similar arguments to the preceding cases, $\widehat{\theta}_{x_1.}=2\gamma_{x_1}$, $\gamma_{x_1}\neq0$ and $\gamma_{x_1}\neq1/2$.
- Face 4 ($t_{x_1y_2},t_{x_2y_1},t_{x_2y_2}$; $\nu_{x_1}=1,\nu_{x_2}=2,\gamma_{x_2}=0$). Analogously to the preceding case, $\widehat{\theta}_{x_1.}=2\gamma_{x_1}$, $\gamma_{x_1}\neq0$ and $\gamma_{x_1}\neq1/2$.
- Face 5 ($t_{x_1y_1},t_{x_1y_2}$; $\nu_{x_1}=2,\nu_{x_2}=0,\gamma_{x_1}=0$). We have $\sum_{x,y}\widehat{\theta}_{xy}=1$, which is also $\widehat{\theta}_{x_1.}+\gamma_{x_2}=1$, or $\widehat{\theta}_{x_1.}=1-\gamma_{x_2}$.
- Face 6 ($t_{x_1y_1},t_{x_2y_1}$; $\nu_{x_1}=1,\nu_{x_2}=1$). By (18) we have $(\widehat{\theta}_{x_1.}-\gamma_{x_1})/\widehat{\theta}_{x_1.}=\left(\widehat{\theta}_{x_2.}-\gamma_{x_2}\right)/\widehat{\theta}_{x_2.}$. If $\gamma_{x_1}=\gamma_{x_2}=0$, the equation is an identity; from (18) we have $\alpha=1$ and from (11) we have that f is constantly equal to zero on the present face. When both γ_{x_1} and γ_{x_2} are positive, the equation holds iff $\widehat{\theta}_{x_1.}=\gamma_{x_1}/(\gamma_{x_1}+\gamma_{x_2})$. (When

only one constant between γ_{x_1} and γ_{x_2} is zero, the equation has no solution.)

- Face 7 ($t_{x_1y_2}, t_{x_2y_1}$; $\nu_{x_1} = 1, \nu_{x_2} = 1$). This case has exactly the same solution as face 6.

- Face 8 ($t_{x_1y_1}, t_{x_2y_2}$; $\nu_{x_1} = 1, \nu_{x_2} = 1$). This case has exactly the same solution as face 6.

- Face 9 ($t_{x_1y_2}, t_{x_2y_2}$; $\nu_{x_1} = 1, \nu_{x_2} = 1$). This case has exactly the same solution as face 6.

- Face 10 ($t_{x_2y_1}, t_{x_2y_2}$; $\nu_{x_1} = 0, \nu_{x_2} = 2, \gamma_{x_2} = 0$). Analogously to face 5, $\widehat{\theta}_{x_2\cdot} = 1 - \gamma_{x_1}$.

- Face 11 ($t_{x_1y_1}$; $\nu_{x_1} = 1, \nu_{x_2} = 0$). From (14), it follows $t_{x_1y_1} = 1$.

- Face 12 ($t_{x_1y_2}$; $\nu_{x_1} = 1, \nu_{x_2} = 0$). Analogously to face 11, $t_{x_1y_2} = 1$.

- Face 13 ($t_{x_2y_1}$; $\nu_{x_1} = 0, \nu_{x_2} = 1$). Analogously to face 11, $t_{x_2y_1} = 1$.

- Face 14 ($t_{x_2y_2}$; $\nu_{x_1} = 0, \nu_{x_2} = 1$). Analogously to face 11, $t_{x_2y_2} = 1$.

- Inner part ($t_{x_1y_1}, t_{x_1y_2}, t_{x_2y_1}, t_{x_2y_2}$; $\nu_{x_1} = 2, \nu_{x_2} = 2, \gamma_{x_1} = 0, \gamma_{x_2} = 0$). In this case it is possible that more than one point satisfies the first-order necessary condition. By $\sum_{x,y} \widehat{\theta}_{xy} = 1$ and (17) we have $\sum_{x,y} \alpha\widehat{\theta}_{x\cdot} = 1$ and hence $\alpha = 1/2$. This, together with (17) and (11), implies that the value of $f(\cdot)$ is the same for all the points that satisfy (17), i.e.: $(N + s)\ln(1/2)$. In order to cope with the minimization on the inner part of the polytope, it is sufficient to test whether there is at least one point that satisfies (17). Proposition 4 realizes such a test.

Proposition 4 *The first-order necessary condition holds for at least one point in the inner part of the polytope iff* $|n_{\cdot y_1} - n_{\cdot y_2}| < s$.

Proof.

When dealing with the inner part of the polytope, condition (17) is equivalent to the system of linear equalities below,

$$\begin{cases} n_{x_1y_1} + st_{x_1y_1} = n_{x_1y_2} + st_{x_1y_2} \\ n_{x_2y_1} + st_{x_2y_1} = n_{x_2y_2} + st_{x_2y_2}. \end{cases}$$

Call $\delta = n_{x_1y_2} + n_{x_2y_2} - n_{x_2y_1} - n_{x_1y_1} = n_{\cdot y_2} - n_{\cdot y_1}$. From the system and from (14), we get $t_{x_1y_1} + t_{x_2y_1} = (s + \delta)/(2s)$ and $t_{x_1y_2} + t_{x_2y_2} = (s - \delta)/(2s)$. If δ did not belong to the interval $(-s, s)$, there could not be a point in the domain satisfying (17), because the constraints $\sum_{x,y} t_{xy} = 1$ and $t_{xy} > 0$ (for all x, y) could not be met. Conversely, when $\delta \in (-s, s)$, we have $t_{x_1y_1} + t_{x_2y_1} \in (0, 1)$, $t_{x_1y_2} + t_{x_2y_2} \in (0, 1)$ and $\sum_{x,y} t_{xy} = 1$. The thesis follows by choosing $t_{x_1y_1} = t_{x_2y_1} = (s + n_{\cdot y_2} - n_{\cdot y_1})/(4s)$ and $t_{x_1y_2} = t_{x_2y_2} = (s - n_{\cdot y_2} + n_{\cdot y_1})/(4s)$. ■

This proposition makes it trivial to verify the existence of a point in the inner part of the polytope that satisfies the necessary condition. Furthermore, note that the necessary condition is always satisfied by one point as far as the faces 11–14 are concerned. The test related to the remaining faces is carried out by verifying the equalities (17) given (18) and the $\widehat{\theta}_{x\cdot}$-values provided by the points above (recall that $\widehat{\theta}_{x_1\cdot} + \widehat{\theta}_{x_2\cdot} = 1$).

The overall procedure then reduces to verify the necessary condition 15 times and to compute the value of $f(\cdot)$ on the set of points that satisfy it. This needs to be done 15 times at most. The minimum value of $f(\cdot)$ among those computed is the sought global optimum.

6.2 No node in common

Let us develop the comparison of two edges that do not share a node. Consider four different random variables: X, Y, W, Z. We must compare the edge (X, Y) with the edge (W, Z). Similarly to Section 6.1, we say that (X, Y) dominates (W, Z) if the inequality $\mu(X, Y|\mathbf{n}_{XY}, \mathbf{t}_{XY}) > \mu(W, Z|\mathbf{n}_{W,Z}, \mathbf{t}_{W,Z})$ holds for all the values of the vectors $\mathbf{t}_{XY}$ and $\mathbf{t}_{W,Z}$. By letting $\underline{\mu}(X, Y|\mathbf{n}_{XY})$ denote the minimum of $\mu(X, Y|\mathbf{n}_{XY}, \mathbf{t}_{XY})$ over the unit simplex for $\mathbf{t}_{XY}$, and by $\overline{\mu}(W, Z|\mathbf{n}_{W,Z})$ the maximum of $\mu(W, Z|\mathbf{n}_{W,Z}, \mathbf{t}_{W,Z})$ over the unit simplex for $\mathbf{t}_{W,Z}$, the preceding inequality holds iff $\underline{\mu}(X, Y|\mathbf{n}_{XY}) > \overline{\mu}(W, Z|\mathbf{n}_{W,Z})$.

Let us focus on $\underline{\mu}(X, Y|\mathbf{n}_{XY})$ (the remaining case is analogous). This is the result of the following optimization problem:

$$\min \mu(X, Y|\mathbf{n}_{XY}, \mathbf{t}_{XY}) \tag{19}$$

$$\sum_{x,y} t_{xy} = 1 \tag{20}$$

$$t_{xy} \geq 0 \qquad \forall x, y. \tag{21}$$

As for the previously addressed problems, also this optimization can hardly be computed exactly. For this reason, the objective function is firstly rewritten as $\sum_{x,y}\left(\widehat{\theta}_{xy} \ln \frac{\widehat{\theta}_{xy}}{\widehat{\theta}_{x\cdot}}\right) - \sum_{y}\left(\widehat{\theta}_{\cdot y} \ln \widehat{\theta}_{\cdot y}\right)$. Then the two summations are optimized separately by minimizing the first and maximizing the second, so that their difference provides us with a lower bound on $\underline{\mu}(X, Y|\mathbf{n}_{XY})$. (Note that the procedure can be applied also by expanding the objective as

$\sum_{x,y}\left(\widehat{\theta}_{xy}\ln\frac{\widehat{\theta}_{xy}}{\widehat{\theta}_{.y}}\right)-\sum_y\left(\widehat{\theta}_{x.}\ln\widehat{\theta}_{x.}\right)$; and the approximation can be improved by taking the maximum of the two lower bounds.)

The minimization of the first term has already been tackled in Sections 6.1.1 and 6.1.2, so here I will focus on the maximization of $g(\mathbf{t}_Y)=\sum_y\left(\widehat{\theta}_{.y}\ln\widehat{\theta}_{.y}\right)$, i.e. on the problem

$$\max\sum_y\left(\frac{n_{.y}+st_{.y}}{N+s}\ln\frac{n_{.y}+st_{.y}}{N+s}\right) \tag{22}$$

$$\sum_y t_{.y}=1 \tag{23}$$

$$t_{.y}\geq 0 \qquad \forall y. \tag{24}$$

Following the same procedure used in past sections, we have that $\frac{\partial}{\partial t_{.y}}g(\mathbf{t}_Y)=\frac{s}{N+s}\left(1+\ln\widehat{\theta}_{.y}\right)$ by assuming $t_{.y}>0$. Karush-Kuhn-Tucker's first-order necessary condition states that the partial derivatives must have the same value in the stationary points (see Section 6.1.1). This implies

$$\widehat{\theta}_{.y}=\alpha \quad \forall t_{.y}>0, \tag{25}$$

where $\alpha\in\mathbb{R}^+$ is the common value of the derivatives. Define $\gamma=\sum_{y:t_{.y}=0}\widehat{\theta}_{.y}$ and let ν denote the number of variables $t_{.y}$ that are zero on a given subset of the unit simplex. It is easily obtained $\alpha=(1-\gamma)/\nu$.

In order to solve problem (22) with constraints (23) and (24), it therefore suffices to follow a method similar to that in Section 6.1.1. The unit simplex is considered as the partition made by the faces of the polytope and its inner part. $g(\cdot)$, as function of the positive variables only, is maximized over each element of the partition. The global maximum is the largest value among those obtained over the elements of the partition. Note that for each element there can be at most one point that satisfies (25).

7 Conclusions

This work proposes a robust procedure to infer tree-dependency structures from a multinomial sample. The basic tool used is Walley's imprecise Dirichlet model. This makes several structures be plausible, given the data, that converge to a single tree as the sample relative frequencies approach the underlying chances. This work focuses on the inference about the structure common to all the dependency trees.

The considered task is difficult. I have proposed several approximations to make it viable. The greatest part of these simply add an excess of caution to the inferences, due to the current limited ability to exactly solve some problems efficiently. This is the case of the bounds computed in Sections 6.1 and 6.2, and of algorithm 2 that cannot generally detect the entire common pattern. This is also the case of the assumption that all the total orders consistent with the given partial order are admissible, in Section 3, which is not necessarily verified by the partial order produced by the mutual information (there may be restrictions of this type due to the logical dependence of the values of mutual information on different edges). On another side, the proposed approximation to the expected value of the mutual information under the IDM, in Section 5.2, is due to a seemingly missing method, in the literature, to approximate the distribution of mutual information under a Dirichlet density for the unknown chances. It is important for future work to provide the error of the approximation: the proposed procedures can be easily modified to take it into account.

Note that these problems are not shared by the discovery algorithms that neglect imprecision. Choosing an ideally precise approach removes all of them and helps creating much simpler algorithms. But the problems are not actually avoided, they are simply transferred to the output of the algorithms that can be unreliable and whose usefulness is thus questionable.

There are many improvements of the present work that should be pursued, as relaxing the constraint of dealing with binary random variables or studying new ways to compute bounds on the values of the optimization problems. Reducing the computational complexity of the discovery algorithm would be very useful for data analysis applications. Extending the approach to incomplete samples is another important issue. Previous work on missing data might be exploited to this extent [21]. Moreover, experimental analyses are needed to verify in practice the capability of the proposed algorithm to infer tree structures, with special emphasis on the relationship between the sample size and the fraction of structure recovered.

Some other extensions might be worth trying. Robustness might be emphasized by computing credibility intervals for mutual information under the IDM [20] instead of expectations. This would involve extending Section 5 and the followings, in order to produce the partial order on the edges under the new conditions. It would not require modifying the first part of the paper, concerned with the general problem of the common pattern, given that it works for a general partial order. Some of the ideas in this paper might also apply to the problem of recovering dependence structures when dependency indexes different from mutual information were used, such as the statistical coefficient ϕ^2 ([9], pp. 556–561). The advantage of focusing on optimal structures would be lost, though, unless a proof similar to Chow and Liu's one were available.

In summary, the task of inferring reliable dependences is difficult, but it is possible, as shown in this paper, to act on the relationship between caution and complexity to develop practicable algorithms: by adding an excess of caution it is often possible to reduce the complexity of the problems. This can lead to efficient algorithms while avoiding overconfident methods.

Acknowledgements

Thanks to Fabio Stella for his helpful support in the literature search. Thanks also to two anonymous reviewers for many valuable comments.

References

[1] J.-M. Bernard. Implicative analysis for multivariate binary data using an imprecise Dirichlet model. *Journal of Statistical Planning and Inference*, 2001. To appear.

[2] S. Chanas and D. Kuchta. Discrete fuzzy optimization. In R. Słowiński, editor, *Fuzzy Sets in Decision Analysis, Operations Research and Statistics*, pages 249–280. Kluwer, Dordrecht, The Netherlands, 1998.

[3] J. Cheng, D. A. Bell, and W. Liu. An algorithm for Bayesian belief network construction from data. In P. Smyth and D. Madigan, editors, *AI&STAT'97*, pages 83–90, Florida, 1997.

[4] C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, IT-14(3):462–467, 1968.

[5] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, New York, 1991.

[6] S. Dasgupta. Learning polytrees. In *UAI-99*, pages 134–141, San Francisco, 1999. Morgan Kaufmann.

[7] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian networks classifiers. *Machine Learning*, 29(2/3):131–163, 1997.

[8] T. Itoh and H. Ishii. An approach to the fuzzy spanning tree problem by maximizing the possibility measure. In M. Fushimi and K. Tone, editors, *Proc. of the 3rd Conference of the Asian-Pacific Operations Research Societies*, Singapore, 1994. World Scientific.

[9] M. G. Kendall and A. Stuart. *The Advanced Theory of Statistics*. Griffin, London, 1967. 2nd edition.

[10] G. D. Kleiter. The posterior probability of Bayes nets with strong dependences. *Soft Computing*, 3:162–173, 1999.

[11] J. B. Kruskal Jr. On the shortest spanning subtree of a graph and the traveling salesman problem. In *Proc. Am. Math. Soc.*, volume 7, pages 48–50, 1956.

[12] H. W. Kuhn and A. W. Tucker. Nonlinear programming. In J. Neyman, editor, *Proc. of the 2nd Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492, Berkeley, 1951. Univ. of California Press.

[13] S. Kullback. *Information Theory and Statistics*. Dover, 1968.

[14] S. Kullback and R. A. Leiber. On information and sufficiency. *Ann. Math. Statistics*, 22:79–86, 1951.

[15] D. G. Luenberger. *Linear and Nonlinear Programming*. Addison Wesley, 1984. 2nd edition.

[16] M. Meila and M. I. Jordan. Learning with mixtures of trees. *Journal of Machine Learning Research*, 1:1–48, 2000.

[17] R. Orre. *Data Mining and Process Modeling using a Bayesian Confidence Propagation Neural Network*. PhD thesis, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm Univ., 1998.

[18] H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice Hall, New York, 1982.

[19] T. Verma and J. Pearl. Equivalence and synthesis of causal models. In P. P. Bonissone, M. Henrion, L. N. Kanal, and J. F. Lemmer, editors, *UAI'90*, pages 220–227, New York, 1990. Elsevier.

[20] P. Walley. Inferences from multinomial data: learning about a bag of marbles. *J. R. Statist. Soc. B*, 58(1):3–57, 1996.

[21] M. Zaffalon. Exact credal treatment of missing data. *Journal of Statistical Planning and Inference*. To appear.