

# Learning from incomplete data in Bayesian networks with qualitative influences

Andrés R. Masegosa<sup>a,\*</sup>, Ad J. Feelders<sup>b</sup>, Linda C. van der Gaag<sup>b</sup>

<sup>a</sup> Department of Computer and Information Science, The Norwegian University of Science and Technology, Norway

<sup>b</sup> Institute of Information and Computing Sciences, Utrecht University, The Netherlands

## ARTICLE INFO

### Article history:

Received 27 July 2015

Received in revised form 4 November 2015

Accepted 6 November 2015

Available online 11 November 2015

### Keywords:

Bayesian networks  
Qualitative influences  
Parameter learning  
Missing data  
EM algorithm  
Isotonic regression

## ABSTRACT

Domain experts can often quite reliably specify the sign of influences between variables in a Bayesian network. If we exploit this prior knowledge in estimating the probabilities of the network, it is more likely to be accepted by its users and may in fact be better calibrated with reality. We present two algorithms that exploit prior knowledge of qualitative influences in learning the parameters of a Bayesian network from incomplete data. The isotonic regression EM, or irEM, algorithm adds an isotonic regression step to standard EM in each iteration, to obtain parameter estimates that satisfy the given qualitative influences. In an attempt to reduce the computational burden involved, we further define the qirEM algorithm that enforces the constraints imposed by the qualitative influences only once, after convergence of standard EM. We evaluate the performance of both algorithms through experiments. Our results demonstrate that exploitation of the qualitative influences improves the parameter estimates over standard EM, and more so if the proportion of missing data is relatively large. The results also show that the qirEM algorithm performs just as well as its computationally more expensive counterpart irEM.

© 2015 Elsevier Inc. All rights reserved.

## 1. Introduction

While domain experts often have difficulties in coming up with numerical probability assessments, experience shows that they feel more comfortable with providing qualitative knowledge about the probabilistic influences between variables in a Bayesian network [10,19]. The qualitative knowledge provided by the experts, moreover, tends to be more robust than their numerical assessments. For example, experts can quite reliably state knowledge of the type: *smoking increases the probability of lung cancer*, but have much more problems with specifying the exact probabilities.

Previous work [2,27,14] has shown that the exploitation of prior knowledge about qualitative influences can improve parameter learning in Bayesian networks. This improvement can be observed in particular when training data is scarce, because in that case the order constraints on the parameters (resulting from the specified qualitative influences) may be violated in the training sample due to sampling variability. If the specified influences are correct and the training sample is large, the constraints tend to be satisfied anyway, and are therefore less useful. In practical learning problems we are often confronted with missing values in the training data, and it stands to reason that knowledge of qualitative influences can be

\* Corresponding author.

E-mail addresses: andresmasegosa@ntnu.no (A.R. Masegosa), A.J.Feelders@uu.nl (A.J. Feelders), L.C.vanderGaag@uu.nl (L.C. van der Gaag).

<sup>1</sup> This research was conducted when the author was affiliated at the Department of Computer Science and Artificial Intelligence, University of Granada, Spain.

particularly helpful in such cases. Therefore, in this paper we focus on parameter learning from incomplete data, assuming there are order restrictions on the parameters resulting from qualitative influences specified by a domain expert.

Earlier work on parameter learning with order constraints from complete data [14] proposed to use the *isotonic regression* [24] for this purpose. In case the network variables are binary, the isotonic regression produces the constrained maximum likelihood estimates, and in the more general case of ordinal variables its performance was shown to be indistinguishable from that of constrained maximum likelihood estimation [12]. In this paper we restrict our attention to networks with binary variables. We propose to augment the maximization step of the EM algorithm with the isotonic regression in order to obtain parameter estimates that satisfy the order constraints from incomplete data. This algorithm, called *irEM*, requires the application of the isotonic regression in each iteration of EM. In an attempt to speed up the learning process, we also propose *qirEM*, which applies the isotonic regression only once, namely after the standard EM algorithm has converged. Experiments show that *qirEM* produces parameter estimates of the same quality as the computationally more expensive *irEM* algorithm.

This work is related to [8,22] which address the problem of parameter learning from incomplete data with a more general class of constraints that contains the order constraints resulting from qualitative influences as a special case. To handle this more general class of constraints, they require the application of convex programming in the M-step of EM, thereby adding another iterative numerical optimization step. By limiting ourselves to qualitative influences we can compute the exact constrained maximum in polynomial time in the M step. Moreover, *qirEM* passes over the constrained optimization step altogether. It is applied only once, after the convergence of normal EM.

This paper is organized as follows. In the next section we introduce the basic concepts and notation required for the remainder of the paper. In section 4 we present the *irEM* algorithm and its faster alternative *qirEM*. This section also provides an analysis of the properties of these algorithms. Section 5 presents an experimental evaluation of the algorithms on three well-known Bayesian networks from the literature and five network structures that were learned from data. Finally, section 6 concludes.

## 2. Preliminaries

We introduce our notational conventions and briefly review the EM algorithm and the isotonic regression for parameter estimation under incomplete data and monotonicity constraints, respectively.

### 2.1. Notations

We consider a set  $\mathbf{X}$  of binary random variables. Each variable  $X \in \mathbf{X}$  takes the values 0 and 1; we will use  $x$  to denote  $X = 1$  and  $\bar{x}$  to denote  $X = 0$ . A joint value assignment to a (sub-)set of variables  $\mathbf{Y} \subseteq \mathbf{X}$  is written as a vector  $\mathbf{y}$ . The set of all joint assignments to  $\mathbf{Y}$  is denoted by  $\text{Val}(\mathbf{Y})$ .

A Bayesian network  $B$  is a probabilistic graphical model defining a joint probability distribution  $\text{Pr}$  over  $\mathbf{X}$ . This probabilistic model is composed of an acyclic digraph  $G$ , with nodes for the random variables and directed arcs to capture the independency structure over them, and an associated parameter vector  $\theta$ . We use  $\Pi_X \subset \mathbf{X}$  to denote the set of parents of the variable  $X$  in  $G$ . A joint value assignment to this set  $\Pi_X$  will be termed a parent configuration for  $X$ . We associate with the set  $\text{Val}(\Pi_X)$  of all parent configurations of  $X$  a partial order on its elements, which will be denoted as  $\preceq_X$ . The parameter vector  $\theta$  now includes for each variable  $X \in \mathbf{X}$ , the elements  $\theta_{x\pi} = \text{Pr}(x | \pi)$  for all parent configurations  $\pi$ ; we use  $\theta_X$  to denote the set of all elements specified for the variable  $X$ . The parameter vector  $\theta$  fully describes the joint distribution  $\text{Pr}$  given the independency structure from  $G$ . In the remainder of the paper, we consider the digraph  $G$  of a network  $B$  to be fixed, and address the estimation of its parameter vector  $\theta$ .

A qualitative influence between two variables  $X$  and  $Y$  connected by an arc  $X \rightarrow Y$  describes how observing a value for the one variable affects the probability distribution of the other variable. A positive qualitative influence of  $X$  on  $Y$  expresses that observing  $x$  increases the probability of observing  $y$  (i.e.  $Y = 1$ ), assuming that the values of the other parents of  $Y$  remain the same, that is,

$$\text{Pr}(y | x, \mathbf{s}) \geq \text{Pr}(y | \bar{x}, \mathbf{s}) \quad (1)$$

for any combination of values  $\mathbf{s}$  for the set of parents of  $Y$  other than  $X$ ; a negative influence between  $X$  and  $Y$  expresses that

$$\text{Pr}(y | x, \mathbf{s}) \leq \text{Pr}(y | \bar{x}, \mathbf{s}) \quad (2)$$

for any such combination  $\mathbf{s}$ .

We further consider a multiset  $\mathcal{D} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$ , of data samples that were independently drawn from the distribution  $\text{Pr}$  over  $\mathbf{X}$ . Each sample  $\mathbf{x}^{(i)}$  from  $\mathcal{D}$  is a partially observed vector which may include one or more missing values. We use  $\mathbf{O}^{(i)}$  and  $\mathbf{o}^{(i)}$  to denote the set of observed variables of the sample  $\mathbf{x}^{(i)}$  and its associated joint value assignment, respectively. We use  $\mathbf{U}^{(i)}$  for the set of unobserved variables of  $\mathbf{x}^{(i)}$ ; note that the set of unobserved variables may differ among samples. We further consider the possible completions of a data sample  $\mathbf{x}^{(i)}$  and write  $\mathcal{U}^{(i)}$  to denote the set of all possible joint value assignments  $\mathbf{u}^{(i)}$  to the sample's unobserved variables. The pair  $(\mathcal{U}, \mathcal{D})$  with  $\mathcal{U} = \times_{i=1, \dots, m} \mathcal{U}^{(i)}$  thus defines the possible completions of the dataset  $\mathcal{D}$ ; we will write  $\mathbf{u} \in \mathcal{U}$  to denote such a completion.

We use  $\ell(\theta; \mathcal{D})$  to denote the log-likelihood of the data  $\mathcal{D}$  given the Bayesian network  $B$  with the parameter vector  $\theta$ . Under the assumption that values are missing at random, the log-likelihood function  $\ell(\theta; \mathcal{D})$  in  $\theta$  equals:

$$\begin{aligned}\ell(\theta; \mathcal{D}) &= \ln \sum_{\mathbf{u} \in \mathcal{U}} \Pr(\mathbf{u}, \mathcal{D} \mid \theta) \\ &= \sum_{i=1}^m \ln \sum_{\mathbf{u}^{(i)} \in \mathcal{U}^{(i)}} \Pr(\mathbf{u}^{(i)}, \mathbf{o}^{(i)} \mid \theta)\end{aligned}$$

that is, the log-likelihood of the data given  $\theta$  is computed by marginalizing out the unobserved variables in all data samples [20].

The task of learning the parameter vector  $\theta$  for a Bayesian network  $B$  from the available data  $\mathcal{D}$  now amounts to maximizing the log-likelihood of the data given this vector. The parameter vector  $\hat{\theta}$  that best describes the data thus equals

$$\hat{\theta} = \arg \max_{\theta} \ell(\theta; \mathcal{D}) \quad (3)$$

Unfortunately, the log-likelihood function  $\ell(\theta; \mathcal{D})$  is not convex in  $\theta$ : while for a single completion  $\mathbf{u}$  of the data the distribution  $\Pr(\mathbf{u}, \mathcal{D} \mid \theta)$  is unimodal, the sum of the distributions over an (exponential) number of completions is not. Since the log-likelihood function in addition does not decompose as a product of independent terms, a closed-form solution for the maximum likelihood parameter vector  $\hat{\theta}$  is not known. Iterative approximation methods are usually employed to obtain local maximum solutions, as the one detailed in the next section.

## 2.2. The EM algorithm

The EM algorithm is the most widely used iterative method for maximizing a non-convex log-likelihood function [9]. The algorithm starts with an initial guess  $\theta^0$  for the parameter vector  $\theta$  under study; this initial vector may be chosen randomly or be obtained by some heuristic method. In each iteration, EM then generates, from the vector  $\theta^t$  from the previous iteration, a new parameter vector  $\theta^{t+1}$  by the following two steps, until convergence:

**E Step:** Compute the expected log-likelihood function in  $\theta$ :

$$\mathbb{E}_{\Pr(\mathcal{U} \mid \mathcal{D}, \theta^t)} [\ln \Pr(\mathcal{U}, \mathcal{D} \mid \theta)] = \sum_{\mathbf{u} \in \mathcal{U}} \Pr(\mathbf{u} \mid \mathcal{D}, \theta^t) \cdot \ln \Pr(\mathbf{u}, \mathcal{D} \mid \theta)$$

**M Step:** Compute the new parameter vector  $\theta^{t+1}$  from  $\theta^t$ :

$$\theta^{t+1} = \arg \max_{\theta} \mathbb{E}_{\Pr(\mathcal{U} \mid \mathcal{D}, \theta^t)} [\ln \Pr(\mathcal{U}, \mathcal{D} \mid \theta)]$$

EM always increases, or leaves unchanged, the log-likelihood in each iteration due to a well-known theorem which states that if  $\mathbb{E}_{\Pr(\mathcal{U} \mid \mathcal{D}, \theta^t)} [\ln \Pr(\mathcal{U}, \mathcal{D} \mid \theta)] \geq \mathbb{E}_{\Pr(\mathcal{U} \mid \mathcal{D}, \theta^t)} [\ln \Pr(\mathcal{U}, \mathcal{D} \mid \theta^t)]$ , then  $\ell(\theta; \mathcal{D}) \geq \ell(\theta^t; \mathcal{D})$  [9]. Because the log-likelihood is bounded (under mild assumptions), this algorithm is guaranteed to converge. In general, it can be also guaranteed that it converges to a stationary point of the log-likelihood function (although there are exceptions) [28]. A stationary point can be a local maximum but also a local minimum, or a saddle point. However, the convergence to a non-local maximum is quite unusual in practice, so through the rest of the paper we assume that EM converges to local maxima of the log-likelihood.

Thus far, we stated the EM algorithm in rather general terms. In view of parameter estimation for a Bayesian network however, the algorithm can also be framed as applying the following simple updating rule for each element  $\theta_{x\pi}$  of the parameter vector being constructed:

$$\theta_{x\pi}^{t+1} = \frac{\hat{n}_{x\pi}^t}{\hat{n}_{\pi}^t} = \frac{\sum_i \Pr(x, \pi \mid \mathbf{o}^{(i)}, \theta^t)}{\sum_i \Pr(\pi \mid \mathbf{o}^{(i)}, \theta^t)} \quad (4)$$

where  $\Pr(x, \pi \mid \mathbf{o}^{(i)}, \theta^t)$  is the posterior probability of the joint value assignment  $x, \pi$  given the data sample  $\mathbf{o}^{(i)}$ , as it is induced by the Bayesian network supplemented with the parameter vector  $\theta^t$  from the previous iteration; the probability  $\Pr(\pi \mid \mathbf{o}^{(i)}, \theta^t)$  has an analogous meaning. We note that computing the probability  $\Pr(x, \pi \mid \mathbf{o}^{(i)}, \theta^t)$  requires inference on the network. A single propagation for each data sample throughout the associated junction tree suffices however for computing all the probabilities that are required for establishing the updated parameter vector  $\theta^{t+1}$  [20]. The numbers  $\hat{n}_{x\pi}^t$  and  $\hat{n}_{\pi}^t$  in the updating rule are called the expected sufficient statistics and can be looked upon as counts for the parameter at hand. We apply Laplace correction to forestall zero counts.

While the above description represents the most common view of the EM algorithm, we will in this paper also take an alternative view and look upon the EM algorithm as a lower-bound maximization method. In this view, the algorithm is seen as an iterative method for maximizing the following function  $F$  [23]:

$$\begin{aligned}F(\hat{\Pr}, \theta) &= \mathbb{E}_{\hat{\Pr}} [\ln \Pr(\mathcal{U}, \mathcal{D} \mid \theta)] + H_{\hat{\Pr}}(\mathcal{U}) \\ &= \ell(\theta; \mathcal{D}) - D(\hat{\Pr} \parallel \Pr_{\theta})\end{aligned}$$

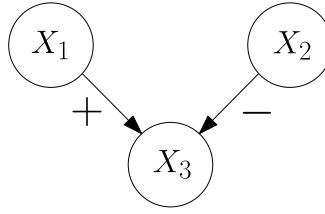


Fig. 1. Network fragment with qualitative influences.

where  $\hat{\Pr}$  is a probability distribution over the possible completions of the data, and  $H_{\hat{\Pr}}$  is the entropy function of  $\hat{\Pr}$ ;  $\Pr_{\theta}$  is a short-hand notation for the probability distribution  $\Pr(\mathcal{U} \mid \mathcal{D}, \theta)$  established from the Bayesian network supplemented with the parameter vector  $\theta$ .  $D(\cdot \parallel \cdot)$  denotes the Kullback–Leibler (KL) divergence between two distributions. We note that since the KL divergence is never negative, we have that the function  $F$  in  $\hat{\Pr}$  and  $\theta$  gives a lower bound on the log-likelihood function  $\ell(\theta; \mathcal{D})$  in  $\theta$ . For a fixed parameter vector  $\theta$ , there is a unique probability distribution  $\hat{\Pr}$  that maximizes  $F$ , and this distribution equals  $\hat{\Pr} = \Pr(\mathcal{U} \mid \mathcal{D}, \theta)$  [23]. In the following formulation of the EM algorithm, the E step serves to approximate this distribution:

**E Step:** Compute  $\hat{\Pr}^{t+1} = \arg \max_{\hat{\Pr}} F(\hat{\Pr}, \theta^t)$

**M Step:** Compute  $\theta^{t+1} = \arg \max_{\theta} F(\hat{\Pr}^{t+1}, \theta)$

This maximization formulation of the algorithm is equivalent to the previous formulation; performing the above two steps thus generates the same sequence of parameter vectors  $\theta^t$  as performing the two steps of the more commonly used expectation–maximization formulation [23]. An important result from the above formulation now is that the algorithm is guaranteed to converge to a local optimum  $(\hat{\Pr}^*, \theta^*)$  of the function  $F$  (with the same caveats as expressed above for the EM algorithm), where  $\theta^*$  is also a local optimum of the log-likelihood function  $\ell(\theta; \mathcal{D})$ .

### 3. Order constrained estimation with complete data

In this section we show how, in case there are no missing values, the order constrained estimates of the parameters of the BN can be computed using the isotonic regression. We start with a short general description of the isotonic regression.

Let  $(Z, \preceq)$  be a partially ordered set. Any real-valued function  $f$  on  $Z$  is *isotonic* with respect to  $\preceq$  if:

$$\forall z, z' \in Z : z \preceq z' \Rightarrow f(z) \leq f(z').$$

Suppose that each element  $z$  of  $Z$  is associated with a real number  $g(z)$ , and a positive weight  $w(z)$ . The function  $f^*$  that minimizes the weighted sum of squares

$$\sum_{z \in Z} w(z) [g(z) - f(z)]^2, \quad (5)$$

within the class of isotonic functions  $f$  on  $Z$  is called the isotonic regression of  $g$  with weights  $w$ .

To establish the connection with order constrained parameter estimation in Bayesian networks, let  $\preceq_X$  denote the partial order on the parent configurations of  $X$  induced by the qualitative influences of the BN. Furthermore, let  $\theta_X = \{\theta_{x\pi} : \pi \in \text{Val}(\Pi_X)\}$  denote the parameter set corresponding to the conditional probability table (henceforth: CPT) of the variable  $X$ . We say that  $\theta_X$  is compatible with the partial order  $\preceq_X$ , denoted by  $\theta_X \sim \preceq_X$ , if  $\theta_X$  is isotonic with respect to  $\preceq_X$ , that is:

$$\forall \pi, \pi' \in \text{Val}(\Pi_X) : \pi \preceq_X \pi' \Rightarrow \theta_{x\pi} \leq \theta_{x\pi'}.$$

Consider for example the network fragment depicted in Fig. 1. We have  $\text{Val}(\Pi_{X_3}) = \{(x_1, x_2), (x_1, \bar{x}_2), (\bar{x}_1, x_2), (\bar{x}_1, \bar{x}_2)\}$ . Since  $X_1$  has a positive influence on  $X_3$ , and  $X_2$  a negative influence, the order constraints on the parameters of the CPT of  $X_3$  are:

1.  $\theta_{x_3}(\bar{x}_1, x_2) \leq \theta_{x_3}(\bar{x}_1, \bar{x}_2)$ ,
2.  $\theta_{x_3}(\bar{x}_1, \bar{x}_2) \leq \theta_{x_3}(x_1, \bar{x}_2)$ ,
3.  $\theta_{x_3}(\bar{x}_1, x_2) \leq \theta_{x_3}(x_1, x_2)$ , and
4.  $\theta_{x_3}(x_1, x_2) \leq \theta_{x_3}(x_1, \bar{x}_2)$ .

The corresponding partial order  $\preceq_{X_3}$  on the parent configurations of  $X_3$  is  $(\bar{x}_1, x_2) \preceq_{X_3} (\bar{x}_1, \bar{x}_2)$ ,  $(\bar{x}_1, \bar{x}_2) \preceq_{X_3} (x_1, \bar{x}_2)$ ,  $(\bar{x}_1, x_2) \preceq_{X_3} (x_1, x_2)$ , and  $(x_1, x_2) \preceq_{X_3} (x_1, \bar{x}_2)$ .

We say that the full parameter vector  $\theta$  is compatible with the qualitative influences of the BN, denoted by  $\theta \sim \preceq$ , if the parameters of each CPT are compatible with the qualitative influences of the BN, that is,  $\forall X \in \mathbf{X} : \theta_X \sim \preceq_X$ .

**Table 1**

The unconstrained estimates  $\hat{\theta}_{X_3}$  for the example corresponding to Fig. 1.

	$\bar{x}_2$	$x_2$
$\bar{x}_1$	$1/6 = 0.167$	$1/8 = 0.125$
$x_1$	$1/3 = 0.333$	$5/9 = 0.556$

**Table 2**

The constrained estimates  $\theta_{X_3}^*$  for the example corresponding to Fig. 1.

	$\bar{x}_2$	$x_2$
$\bar{x}_1$	$1/6 = 0.167$	$1/8 = 0.125$
$x_1$	$6/12 = 0.500$	$6/12 = 0.500$

Let  $\mathcal{D}$  be a complete data set for  $\mathbf{X} = (X_1, \dots, X_n)$ , let  $G$  be a network structure over these variables, and suppose that the parameters  $\theta_{X_i}$  are disjoint from  $\theta_{X_j}$  for all  $j \neq i$ . Let  $\theta_{X_i}^*$  be the parameter values that maximize  $\ell_i(\theta_{X_i} : \mathcal{D})$  within the order compatible space  $\theta_{X_i} \sim \preceq_{X_i}$ , where

$$\ell_i(\theta_{X_i} : \mathcal{D}) = \sum_{j=1}^m \ln \Pr(X_i^{(j)} \mid \Pi_{X_i}^{(j)} : \theta_{X_i})$$

is the local log-likelihood function for  $X_i$ . Then

$$\theta^* = (\theta_{X_1}^*, \theta_{X_2}^*, \dots, \theta_{X_n}^*)$$

maximizes  $\ell(\theta : \mathcal{D})$  within the order compatible space  $\theta \sim \preceq$ .

This global decomposition property follows immediately from the global decomposition property of Bayesian networks (see [20], Proposition 17.1, page 725), together with the observation that each order constraint only involves parameters from a single CPT.

**Theorem 1.** *The complete-data log-likelihood function is maximized over the order-compatible parameter space  $\theta \sim \preceq$ , by taking for each  $X \in \mathbf{X}$  the isotonic regression of*

$$\hat{\theta}_X = \{\hat{\theta}_{X\pi} = \frac{n_{X\pi}}{n_\pi} : \pi \in \text{Val}(\Pi_X)\},$$

with weights  $\{n_\pi : \pi \in \text{Val}(\Pi_X)\}$  and order  $(\text{Val}(\Pi_X), \preceq_X)$ .

**Proof.** Because of the global decomposition property we can solve the maximization problem for each CPT separately and then combine the solutions. Furthermore, it follows from Theorem 1.5.1 and Example 1.5.1 of Robertson et al. [24], that the constrained maximum for each CPT is obtained by the isotonic regression as stated.  $\square$

The example in Table 1 contains the unconstrained estimates  $\hat{\theta}_{X_3} = \frac{n_{X_3\pi}}{n_\pi}$  for the network fragment in Fig. 1 and for

$$\pi \in \{(\bar{x}_1, \bar{x}_2), (x_1, \bar{x}_2), (\bar{x}_1, x_2), (x_1, x_2)\}.$$

According to the specified qualitative influences, the estimates should be increasing (non-decreasing) in the columns, and decreasing (non-increasing) in the rows. We observe however that in the second row of Table 1 the unconstrained estimates are increasing. In this case, this violation is removed by taking the weighted average of the two violating cells and assigning this average to both cells, that is

$$\theta_{X_3(x_1\bar{x}_2)}^* = \theta_{X_3(x_1x_2)}^* = \frac{3 \times 1/3 + 9 \times 5/9}{3 + 9} = \frac{6}{12}.$$

The resulting constrained maximum likelihood estimates are given in Table 2. In general, the isotonic regression resolves violations of the order restrictions by averaging the unconstrained estimates over suitably-chosen subsets (also called *blocks* in this context) of the parent configurations. Hence, it *partitions* the set  $\text{Val}(\Pi_X)$  into a number of blocks on which the isotonic regression is constant.

#### 4. The isotonic regression EM algorithm

The problem of learning the parameters of a qualitative BN from a given data set  $\mathcal{D}$  can be stated as the following maximization problem:

$$\theta^* = \arg \max_{\theta \sim \preceq} \ell(\theta : \mathcal{D}) \quad (6)$$

To solve this maximization problem, we propose to use a combination of the EM algorithm and the isotonic regression. The proposed algorithms are described in the next sections.

##### 4.1. The irEM algorithm

The first algorithm is called the isotonic regression EM algorithm (irEM). The E-step is the same as the standard E-step, but the M-step needs to be restated as follows:

**M Step:** Maximize the *expected log-likelihood function* over those parameter values that are compatible with the partial order  $\preceq$ :

$$\theta^{t+1} = \arg \max_{\theta \sim \preceq} \mathbb{E}_{\Pr(\mathcal{U}|\mathcal{D}, \theta^t)} [\ln \Pr(\mathcal{U}, \mathcal{D}|\theta)] \quad (7)$$

The result stated in [Theorem 2](#) forms the basis of our algorithm.

**Theorem 2.** *The expected complete-data log-likelihood function*

$$\mathbb{E}_{\Pr(\mathcal{U}|\mathcal{D}, \theta^t)} [\ln \Pr(\mathcal{U}, \mathcal{D}|\theta)]$$

is maximized over the order-compatible parameter space  $\theta \sim \preceq$ , by taking for each  $X \in \mathbf{X}$  the isotonic regression of

$$\theta_X^{t+1} = \{\theta_{x\pi}^{t+1} = \frac{\hat{n}_{x\pi}^t}{\hat{n}_\pi^t} : \pi \in \text{Val}(\Pi_X)\},$$

with weights  $\{\hat{n}_\pi^t : \pi \in \text{Val}(\Pi_X)\}$  and order  $(\text{Val}(\Pi_X), \preceq_X)$ .

**Proof.** The proof is obtained from the proof of [Theorem 1](#) by substituting expected counts  $\hat{n}_{x\pi}^t$  and  $\hat{n}_\pi^t$  for observed counts  $n_{x\pi}$  and  $n_\pi$  respectively.  $\square$

The pseudo-code of the irEM algorithm is given in [Algorithm 1](#). First, the normal EM estimates are computed in line 6. In line 8, the order-constrained estimates are computed by performing the appropriate isotonic regression.

---

##### Algorithm 1 irEM ( $G, \mathcal{D}, \preceq$ ).

---

```

1:  $\hat{\theta}^0$  = available case estimates of parameters
2:  $t = 0$ 
3: repeat
4:   for all  $X \in \mathbf{X}$  do
5:     for all  $\pi \in \text{Val}(\Pi_X)$  do
6:
```

$$\hat{\theta}_{x\pi}^{t+1} = \frac{\hat{n}_{x\pi}^t}{\hat{n}_\pi^t} = \frac{\sum_m \Pr(x, \pi | \mathbf{o}^{(m)}, \hat{\theta}^t)}{\sum_m \Pr(\pi | \mathbf{o}^{(m)}, \hat{\theta}^t)}$$

```

7:   end for
8:
```

$$\theta_X^{*t+1} = \arg \min_{f(\cdot) \sim \preceq_X} \sum_{\pi \in \text{Val}(\Pi_X)} \hat{n}_\pi^t \left( \hat{\theta}_{x\pi}^{t+1} - f(\pi) \right)^2$$

```

9:   end for
10:   $t = t + 1$ 
11: until converged
12: return  $\theta^{*t}$ 
```

---

The computational complexity of the irEM method is the same as the EM algorithm for the E-step, but in the M-step we need to additionally compute the isotonic regression to estimate the parameters of each CPT. The complexity of this isotonic regression problem is discussed in section 4.2.

Convergence of irEM is stated by the following theorem,

**Theorem 3.** *For qualitative binary Bayesian networks, irEM converges to a stationary point of the log-likelihood function  $\ell(\theta)$  in the order-compatible parameter space  $\theta \sim \preceq$ .*

**Proof.** The proof is based on the EM convergence's proof given in [28], which guarantees convergence when the complete-data model can be described by a curved exponential family with a *compact* parameter space.<sup>2</sup> First, irEM does not decrease the log-likelihood function at any iteration because in each maximization step we have that

$$\mathbb{E}_{\Pr(\mathcal{U}|\mathcal{D},\theta^t)}[\ln \Pr(\mathcal{U}, \mathcal{D}|\theta^{t+1})] \geq \mathbb{E}_{\Pr(\mathcal{U}|\mathcal{D},\theta^t)}[\ln \Pr(\mathcal{U}, \mathcal{D}|\theta^t)]$$

because  $\theta^t$  is always compatible with  $\preceq$  and  $\theta^{t+1}$  is a maximum inside the order-compatible parameter space, see Equation (7). Second, the log-likelihood function is upper bounded. Third, the order-compatible parameter space  $\theta \sim \preceq$  of a binary BN is compact. This property is deduced from the compactness of the parameter space of a binary BN. The order-compatible parameter space is also compact because it is a subset of this space defined by a set of inclusive linear inequalities (e.g. see equations (1) and (2)). Fourth, qualitative Bayesian networks are curved exponential families because they have the same functional form as binary Bayesian networks, which are known to be curved exponential families [16].  $\square$

#### 4.2. Complexity of computing the isotonic regression

We consider both exact solutions and approximations. To facilitate the discussion, consider the representation of  $\preceq_X$  as a directed acyclic graph with vertex set  $V = \text{Val}(\Pi_X)$  and edges  $E$  such that  $\pi_i \preceq_X \pi_j$  if and only if there is a path from  $v_i$  to  $v_j$ . We call this graph the order graph. Note that there is a one-to-one correspondence between edges in the order graph, and order constraints on the parameters of the CPT of  $X$ . If all parents have a positive (or negative) influence on  $X$ , then the order graph  $(V, E)$  is a  $|\Pi_X|$ -dimensional grid structure. A  $|\Pi_X|$ -dimensional grid has  $\Theta(|\Pi_X||\text{Val}(\Pi_X)|)$  grid edges. Stout [25] shows that in this case an exact solution can be obtained in  $O(|\text{Val}(\Pi_X)|^3 \log |\text{Val}(\Pi_X)|)$  time.

If we are content with an approximation we can do better. To find an approximation where the value for each parameter of the CPT is at most  $\delta$  removed from the optimal regression value, we only require  $O(|\text{Val}(\Pi_X)|^2 \log |\text{Val}(\Pi_X)| \log \frac{1}{\delta})$  time [25].

Dykstra and Robertson [11] describe an iterative algorithm for grid-structured partial orders. This algorithm is very simple to implement, because it only requires the repeated application of the Pool Adjacent Violators (PAV) algorithm for linear orders. The runtime of the PAV algorithm is linear in the number of elements in the order [1]. The computational complexity of such numerical optimization methods depends on the number of iterations needed to reach the optimum and the computational cost of performing each iteration. In turn, the required number of iterations depends of the degree of accuracy  $\epsilon$  with which we want to obtain the solution and of the convergence rate of the particular optimization algorithm [4]. Let  $I_A(\epsilon)$  denote the number of iterations needed by an optimization algorithm  $A$  to converge to an optimum with a degree of accuracy  $\epsilon$ . As is shown in [12], each iteration of the algorithm of Dykstra and Robertson takes  $O(|\Pi_X||\text{Val}(\Pi_X)|)$  operations, so its total complexity is  $O(I_{\text{DR}}(\epsilon) \cdot |\Pi_X||\text{Val}(\Pi_X)|)$ . We do not have a result for the rate of convergence of this algorithm however.

If there are parents without an a priori positive or negative influence on  $X$ , then the isotonic regression problem decomposes: we get a separate problem for each possible value assignment to the set of such parents. In terms of the order graph: each connected component of the order graph can be solved separately. Furthermore, if an unconstrained estimate  $\theta_{x\pi}$  does not violate order constraints with any other unconstrained estimate, then  $\theta_{x\pi}^* = \hat{\theta}_{x\pi}$  and the parameter  $\theta_{x\pi}$  and the corresponding order constraints can be removed from the isotonic regression altogether without influencing the solution obtained. Hence, if the unconstrained estimates are 'almost monotone', then we only have to solve a smaller subproblem.

As we already mentioned in the introduction, other previously proposed works [8,22] use general numerical optimization techniques, instead of isotonic regression, to solve a similar problem to the M-step of the irEM algorithm (see Equation (7)). As we saw above, this problem is a classic constrained concave maximization problem because the expected log-likelihood is a concave function. Two sets of inequality constraints apply to this problem: one set defines that parameters  $\theta_{x\pi}$  should be in the interval  $[0, 1]$ ; and another set defines the inequalities associated to constraints in  $\preceq_X$ . The first set of constraints can be obviated by a proper reparametrization of the problem as done in [22].

In [8], the constrained maximization step is solved by using interior-point (IP) methods, a family of state-of-the-art non-linear convex optimization techniques that can deal with inequality constraints by using the so-called log barrier functions [4]. However this technique requires, for each iteration, the solution of a system of linear equations with  $|\text{Val}(\Pi_X)|$  variables. The computational complexity of this operation is  $O(|\text{Val}(\Pi_X)|^3)$  if we use the Gauss–Jordan elimination method.<sup>3</sup> In this case, the complexity of the approach is  $O(I_{\text{IP}}(\epsilon) \cdot |\text{Val}(\Pi_X)|^3)$ . Additionally, the use of log barrier functions involves

<sup>2</sup> I.e. it is a closed and bounded subset of the Euclidean space.

<sup>3</sup> There exist other more sophisticated algorithms whose complexity is not cubic (e.g. the Strassen algorithm is  $O(|\text{Val}(\Pi_X)|^{2.807})$ ), but their constant factors are much higher.



a larger number of iterations to approximate the optimum because it requires the solution of a sequence of optimizations problems [4].

In [22], the constrained M-step is solved using a simple gradient ascent (GA) method where the optimization function includes extra penalty functions for each inequality associated to the constraints in  $\leq_X$  (i.e. there are  $|\Pi_X| \cdot |\text{Val}(\Pi_X)|$  extra penalty functions) using Karush–Kuhn–Tucker (KKT) multipliers. In this algorithm, the cost of computing the gradient at each iteration is  $O(|\text{Val}(\Pi_X)|^2)$  (i.e. consider that each parameter  $\theta_{x\pi}$  is always involved in  $|\text{Val}(\Pi_X)|$  inequalities). Then, the computational complexity of this optimization algorithm is  $O(I_{GA}(\epsilon)|\text{Val}(\Pi_X)|^2)$ . In this case, although the computational cost of one iteration of GA is lower than in the case of IP methods, this algorithm usually requires a much larger number of iterations to converge to the optimum.

Summarizing, by using specialized algorithms for the isotonic regression instead of general purpose constrained optimization algorithms, we can exploit the additional structure of the order constrained estimation problem at hand. In particular, we have an algorithm that computes the exact solution in polynomial time. Moreover, this exact algorithm does not suffer from issues which are usually associated with numerical optimization algorithms such as assessing the stop convergence point, numerical stability problems, etc.

#### 4.3. Speeding up irEM: the qirEM algorithm

Since the irEM algorithm maximizes the log-likelihood function  $\ell(\theta; \mathcal{D})$  over the order-compatible parameter space and thus needs to perform isotonic regression in each and every iteration, its runtime complexity can become prohibitively large for real-world applications. To alleviate the computational burden involved, we propose an alternative approximate method for maximizing the log-likelihood function over the constrained parameter space. Informally spoken, the alternative algorithm runs the standard EM algorithm until convergence to obtain an unconstrained maximum of the log-likelihood function, and then applies isotonic regression only once to build an order-compatible parameter vector. We call this efficiency-improved algorithm the quick irEM, or qirEM, algorithm for order-compatible parameter estimation.

We elaborate on the qirEM algorithm and show that it derives from the lower-bound maximization point of view of EM described in section 2.2. To this end, we re-consider the constrained maximization problem of having to find

$$\theta^* = \arg \max_{\theta \sim \leq} \ell(\theta; \mathcal{D})$$

and re-frame it as the following minimization problem

$$\theta^* = \arg \min_{\theta \sim \leq} \ell(\hat{\theta}; \mathcal{D}) - \ell(\theta; \mathcal{D})$$

where  $\hat{\theta}$  is the unconstrained maximum log-likelihood parameter vector that describes the data best. We note that the log-likelihood of  $\hat{\theta}$  given the data constitutes an upper bound on the log-likelihood of any order-compatible parameter vector; the above difference thus is always non-negative. We now note that minimizing the difference with respect to a fixed upper bound is an alternative approach to maximizing the log-likelihood function over the order-compatible parameter space. Focusing on the difference  $\ell(\hat{\theta}; \mathcal{D}) - \ell(\theta; \mathcal{D})$  that we now have to minimize, we know from the definition of the function  $F$  from section 2.2 that

$$F(\text{Pr}_{\hat{\theta}}, \hat{\theta}) - F(\text{Pr}_{\hat{\theta}}, \theta) = \ell(\hat{\theta}; \mathcal{D}) - \ell(\theta; \mathcal{D}) + D(\text{Pr}_{\hat{\theta}} \parallel \text{Pr}_{\theta}) \quad (8)$$

for any parameter vector  $\theta$  from the constrained parameter space. Because the Kullback–Leibler divergence  $D(\text{Pr}_{\hat{\theta}} \parallel \text{Pr}_{\theta})$  is always positive, we thus have that

$$\Delta F_{\hat{\theta}}(\theta) \geq \ell(\hat{\theta}; \mathcal{D}) - \ell(\theta; \mathcal{D}) \geq 0 \quad (9)$$

where  $\Delta F_{\hat{\theta}}(\theta)$  is a short-hand notation for the difference of the two  $F$  terms in equation (8). We note that the function  $\Delta F_{\hat{\theta}}(\theta)$  in  $\theta$  is a non-tight upper-bound function on the difference  $\ell(\hat{\theta}; \mathcal{D}) - \ell(\theta; \mathcal{D})$ ; the term  $D(\text{Pr}_{\hat{\theta}} \parallel \text{Pr}_{\theta})$  constitutes the gap with this bound.

The qirEM algorithm is now based on the idea of minimization of the upper-bound function  $\Delta F_{\hat{\theta}}(\theta)$  over the order-compatible parameter space, that is, the algorithm establishes the parameter vector

$$\theta_F^* = \arg \min_{\theta \sim \leq} \Delta F_{\hat{\theta}}(\theta)$$

The main advantage of building upon this minimization formulation of the problem of constrained parameter estimation lies in the observation that the function  $\Delta F_{\hat{\theta}}(\theta)$  is a convex function, from which the minimum can be established in closed form. More specifically, the minimum of the function equals

$$\begin{aligned} \theta_F^* &= \arg \min_{\theta \sim \leq} F(\text{Pr}_{\hat{\theta}}, \hat{\theta}) - F(\text{Pr}_{\hat{\theta}}, \theta) \\ &= \arg \max_{\theta \sim \leq} F(\text{Pr}_{\hat{\theta}}, \theta) \\ &= \arg \max_{\theta \sim \leq} \mathbb{E}_{\text{Pr}(\mathcal{U}|\mathcal{D}, \hat{\theta})} [\ln \text{Pr}(\mathcal{U}, \mathcal{D}|\theta)] \end{aligned} \quad (10)$$



and can be established by performing isotonic regression once (see [Theorem 2](#)). We note that the step from a minimization formulation to a maximization formulation in the derivation above is allowed since  $F(\text{Pr}_{\hat{\theta}}, \hat{\theta}) \geq F(\text{Pr}_{\hat{\theta}}, \theta)$  for all order-compatible parameter vectors  $\theta \sim \preceq$ , as shown in Equation (9). The pseudo-code of qirEM is given in [Algorithm 2](#).

As outlined above, the qirEM algorithm builds upon the assumption that we are able to compute the unconstrained maximum log-likelihood parameter vector  $\hat{\theta}$ . By using the standard EM algorithm for its computation however, we are not guaranteed to actually find the global maximum of the log-likelihood function. We note that our observations for the qirEM algorithm remain to hold whenever the EM algorithm returns a parameter vector  $\hat{\theta}'$  for which  $\ell(\hat{\theta}': \mathcal{D}) \geq \max_{\theta \sim \preceq} \ell(\theta: \mathcal{D})$ . Although this precondition is not necessarily met in general, we found that it was always satisfied throughout our empirical study of the qirEM algorithm.

---

**Algorithm 2** qirEM ( $G, \mathcal{D}, \preceq$ ).

---

```

1:  $\hat{\theta}^{(0)}$  = available case estimates of parameters
2:  $t = 0$ 
3: repeat
4:   for all  $X \in \mathbf{X}$  do
5:     for all  $\pi \in \text{Val}(\Pi_X)$  do
6:
```

$$\hat{\theta}_{x\pi}^{(t+1)} = \frac{\hat{n}_{x\pi}^{(t)}}{\hat{n}_{\pi}^{(t)}} = \frac{\sum_m P(x, \pi | \mathbf{o}^{(m)}, \hat{\theta}^{(t)})}{\sum_m P(\pi | \mathbf{o}^{(m)}, \hat{\theta}^{(t)})}$$

```

7:   end for
8: end for
9:    $t = t + 1$ 
10: until converged
11: for all  $X \in \mathbf{X}$  do
12:
```

$$\theta_X^* = \arg \min_{f(\cdot) \sim \preceq_X} \sum_{\pi \in \text{Val}(\Pi_X)} \hat{n}_{\pi}^{(t-1)} \left( \hat{\theta}_{x\pi}^{(t)} - f(\pi) \right)^2$$

```

13: end for
14: return  $\theta^*$ 

```

---

Since the algorithm minimizes the non-tight upper-bound function  $\Delta F_{\hat{\theta}}(\theta)$ , it is not guaranteed to thereby establish a global maximum of the log-likelihood function  $\ell(\theta: \mathcal{D})$ : there might exist an alternative constrained solution  $\hat{\theta}_F^*$  which does not minimize  $\Delta F_{\hat{\theta}}(\theta)$  but with higher log-likelihood if  $D(\text{Pr}_{\hat{\theta}} \| \text{Pr}_{\hat{\theta}_F^*}) > D(\text{Pr}_{\hat{\theta}} \| \text{Pr}_{\hat{\theta}_F^*})$ .

Another issue is that, in general, we cannot give any guarantee about the local maximality of  $\hat{\theta}_F^*$ , in contrast to the solutions given by irEM. However we have found two particular situations in which  $\hat{\theta}_F^*$  is a local constrained maximum of the log-likelihood function.

The first of them is when the solution given by EM lies inside the order-compatible parameter space.

**Theorem 4.** *If the solution given by EM is inside the order-compatible space, i.e.  $\hat{\theta} \sim \preceq$ , then  $\theta_F^*$  is equal to  $\hat{\theta}$  and therefore locally optimal.*

**Proof.** By the local optimality of  $\hat{\theta}$ , we have that  $\hat{\theta} = \arg \max_{\mathbb{E}_{\text{Pr}(\mathcal{U}|\mathcal{D}, \hat{\theta})}} [\ln \text{Pr}(\mathcal{U}, \mathcal{D}|\theta)]$ , which is the same maximization problem as in Equation (10) provided that  $\hat{\theta} \sim \preceq$ . Then,  $\hat{\theta}_F^* = \hat{\theta}$  and the solution given by qirEM in this case is locally optimal.  $\square$

The second situation is more subtle. It arises from the fact that when minimizing the upper bound  $\Delta F_{\hat{\theta}}(\theta)$  we may be also minimizing  $D(\text{Pr}_{\hat{\theta}} \| \text{Pr}_{\theta})$ , the upper bound gap, because both are quite related problems. We detail together both minimization problems:

$$\arg \min_{\theta \sim \preceq} \Delta F_{\hat{\theta}}(\theta) = \arg \max_{\theta \sim \preceq} \sum_{\mathcal{U}} \text{Pr}(\mathcal{U}|\mathcal{D}, \hat{\theta}) \ln \text{Pr}(\mathcal{U}, \mathcal{D}|\theta) \quad (11)$$

$$\arg \min_{\theta \sim \preceq} D(P_{\hat{\theta}} \| P_{\theta}) = \arg \max_{\theta \sim \preceq} \sum_{\mathcal{U}} \text{Pr}(\mathcal{U}|\mathcal{D}, \hat{\theta}) \ln \text{Pr}(\mathcal{U}|\mathcal{D}, \theta) \quad (12)$$

As can be seen, minimizing  $\Delta F_{\hat{\theta}}(\theta)$  is equivalent to performing a maximization over an expected log-likelihood function, while minimizing  $D(\text{Pr}_{\hat{\theta}} \| \text{Pr}_{\theta})$  is equivalent to performing a maximization over a *conditional* expected log-likelihood function. This is directly related to the concept of *generative training* (maximizing a log-likelihood function) versus *discriminative training* (maximizing a conditional log-likelihood function) [21], but now the class variable(s)  $\mathcal{U}$  would be the expected completions of the missing values according to  $\text{Pr}(\mathcal{U}|\mathcal{D}, \hat{\theta})$  (i.e. the class values would not be either 0 or 1 rather than both

values at the same time with an associated probability each). In consequence, we expect that by performing *generative training* (i.e. minimizing  $\Delta F_{\hat{\theta}}(\theta)$ ) we also perform a good *discriminative training* (i.e. to find parameters close to the minimum of  $D(\text{Pr}_{\hat{\theta}} \parallel \text{Pr}_{\theta})$ ). Although both are different objective functions, it is known that if the model generating the data is close to the model used for learning, then generative training can obtain solutions which perform very well for discriminative purposes [21]. **Theorem 5** tells us that if  $\hat{\theta}_F^*$  minimizes both  $\Delta F_{\hat{\theta}}(\theta)$  and  $D(\text{Pr}_{\hat{\theta}} \parallel \text{Pr}_{\theta})$ , then  $\hat{\theta}_F^*$  could be a local optimum point of the log-likelihood function in the order-compatible space: we can only guarantee that  $\hat{\theta}_F^*$  is a stationary point subject to a further constrained parameter space.

**Theorem 5.** *If the following condition holds:*

$$\theta_F^* = \arg \min_{\theta \sim \leq} \Delta F_{\hat{\theta}}(\theta) = \arg \min_{\theta \sim \leq} D(\text{Pr}_{\hat{\theta}} \parallel \text{Pr}_{\theta})$$

*then at  $\theta_F^*$  we have a stationary point of the log-likelihood function  $\ell(\theta : \mathcal{D})$  subject to the boundary of the parameter space constrained by the inequalities which bind or are active in the above constrained minimum  $\theta_F^*$ .*

**Proof.** Because  $\Delta F$  and  $D$  are continuously differentiable and the constraints imposed by  $\leq$  are linear, we have that the first order Karush–Kuhn–Tucker (KKT) conditions are satisfied at  $\theta_F^*$ , that is:

$$\frac{\partial \Delta F_{\hat{\theta}}(\theta_F^*)}{\partial \theta} = \sum_{k=1}^I \lambda_k \frac{\partial c_k(\theta_F^*)}{\partial \theta} + \sum_{l=1}^E \gamma_l \frac{\partial h_l(\theta_F^*)}{\partial \theta} \quad (13)$$

$$\frac{\partial D(\text{Pr}_{\hat{\theta}} \parallel \text{Pr}_{\theta_F^*})}{\partial \theta} = \sum_{k=1}^I \tau_k \frac{\partial c_k(\theta_F^*)}{\partial \theta} + \sum_{l=1}^E \eta_l \frac{\partial h_l(\theta_F^*)}{\partial \theta} \quad (14)$$

where  $c_k(\cdot)$  ( $k = 1, \dots, I$ ) are the inequality constraints functions (i.e. those that define the order-compatible parameter space),  $h_l(\cdot)$  ( $l = 1, \dots, E$ ) are the equality constraints functions (i.e. those associated to the normalization of the parameters  $\theta$ ) and  $\lambda_k, \tau_k, \gamma_l, \eta_l$  are the so-called KKT multipliers. We also have that  $c_k(\theta_F^*) \geq 0, h_l(\theta_F^*) = 0$  (i.e. primal feasibility conditions);  $\lambda_k \geq 0, \tau_k \geq 0$  (i.e. dual feasibility conditions); and  $\lambda_k c_k(\theta_F^*) = 0, \tau_k c_k(\theta_F^*) = 0$  (i.e. slackness conditions) for  $k = 1, \dots, I$  and  $l = 1, \dots, E$ . Let  $\bar{I} = \{1, \dots, I\}$  and let  $I' \subseteq \bar{I}$  denote the subset of inequalities which binds or are active:  $I' = \{k \in \bar{I} : \lambda_k \neq 0, \tau_k \neq 0, c_k(\theta_F^*) = 0\}$ . Note that the minimum  $\theta_F^*$  is located on the boundary of the region defined by the inequalities in  $I'$ . For those inequalities not binding, we have that  $\forall k \in \bar{I} \setminus I' : \lambda_k = \tau_k = 0$ .

If we take derivatives in Equation (8) and use Equations (13) and (14) and rearrange terms, we have the stationary KKT condition for the  $-\ell(\theta : \mathcal{D})$  function at  $\theta_F^*$ :

$$-\frac{\partial \ell(\theta_F^* : \mathcal{D})}{\partial \theta} = \sum_k (\lambda_k - \tau_k) \frac{\partial c_k(\theta_F^*)}{\partial \theta} + \sum_l (\gamma_l - \eta_l) \frac{\partial h_l(\theta_F^*)}{\partial \theta}$$

The primal feasibility condition follows immediately. The slackness condition also holds because if  $k \in I'$  then  $c_k(\theta_F^*) = 0$  and if  $k \in \bar{I} \setminus I'$  then  $(\lambda_k - \tau_k) = 0$ . The dual feasibility condition holds when we turn those inequalities which binds or are active into equality constraints because we then have that  $(\tau_k - \lambda_k) = 0 \ \forall k \in \bar{I} \setminus I'$ . This change does not affect the above conditions because  $\forall k \in I' : c_k(\theta_F^*) = 0$  (i.e. the primal feasibility condition also holds for the new equality constraints). Then,  $\theta_F^*$  is a stationary point of  $\ell(\theta : \mathcal{D})$  in the constrained parameter space defined by inequality constraints functions  $\{c_k(\cdot)\}_{k \in \bar{I} \setminus I'}$  and the equality constraints functions  $\{h_l(\cdot)\} \cup \{c_k(\cdot)\}_{k \in I'}$ .  $\square$

Note from the above proof that when  $\forall k \in I' : (\tau_k - \lambda_k) \geq 0$ , then  $\theta_F^*$  will be a stationary point subject to our order-compatible parameter space. This analysis is also complemented at the end of the experimental section, where we empirically evaluate how often  $\theta_F^*$  is a local optimum and what happens if this is not the case.

Finally, we want to point out that the order-compatible parameter space is smaller but not necessarily simpler than the unconstrained parameter space. Although it does not contain those local optima which fall in the non-compatible region, it adds complexity to the parameter space by introducing new local optima: those which are on the boundaries of the order-compatible region [7]. It means that irEM might get trapped into constrained local optima which are ignored by EM because it moves in an unconstrained space. We assume that this might be the cause that qirEM could obtain solutions with a higher log-likelihood than irEM, as we show in the experimental evaluation section.

## 5. Experimental evaluation

### 5.1. Experimental set-up

In this section, we compare the empirical performance of irEM to that of plain EM, in order to establish whether exploiting prior knowledge about signs (positive or negative) of qualitative influences helps, and how this may depend on the proportion of missing data. Moreover, we compare the empirical performance of irEM to that of qirEM.

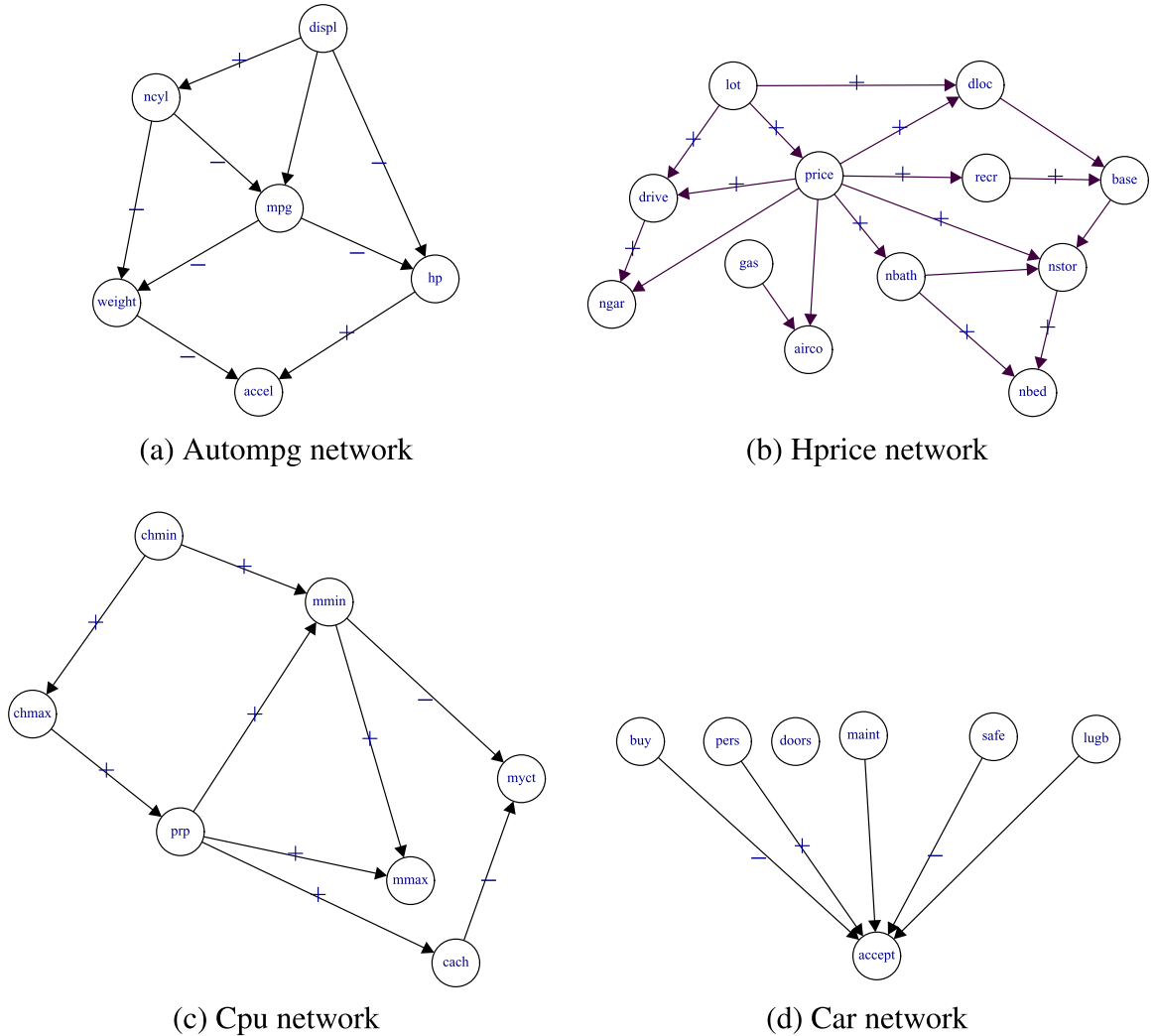


Fig. 2. Examples of Bayesian networks used in the experiments. Edges without a sign indicate an undefined qualitative influence.

Data is generated from completely specified Bayesian Networks with binary variables. The signs of qualitative influences are determined by inspection of the CPT's. These signs are provided to the irEM and qirEM algorithms, that use them to constrain their parameter estimates accordingly.

On one hand, we consider three publicly available BNs which were completely built using expert knowledge and whose majority of edges have a defined qualitative influence: Boerlage92 [5], Car-starts [17] and Oesophagus [13]. On the other hand, we also consider BNs built using automatic learning methods on five publicly available data sets. These networks are included because they are of a different nature than those built using expert knowledge. Three of these data sets (automp, car and wisconsin) were also used by Altendorf et al. [2] and are available from the UCI machine learning repository [15]. The other two data sets, the cpu data set and the hprice (Windsor housing) data set, are available from the UCI repository and from the Journal of Applied Econometrics data archive [3], respectively. All variables were made binary using equal frequency bins, and the complete data set was used to learn the network structure and parameters in each case. The structure was learned by performing a greedy hill climbing search with 100 random restarts using a BDeu score [18] with an equivalent sample size parameter equal to 2 (i.e.  $s = 2$ , which implies a Laplace Beta prior over the parameters) using the *deal* package in R [6]. The parameters were learned under the same settings. In Fig. 2, we display the structure and also the signs of the edges of four of the eight BNs used in the experiments. As can be seen, most edges in these networks have either a positive or negative qualitative influence.

In Table 3, we detail for each network built with expert knowledge and for each network learnt from data the number of nodes, the number of edges with a positive, negative or undefined influence, the number of parameters, and the maximum number of parents.

**Table 3**  
Bayesian networks used in the experiments.

Name	Nodes	N. edges			N. param.	Max. parents
		“+”	“−”	“?”		
autompg	6	2	6	1	19	2
boerlage92	23	31	5	0	86	4
car	7	1	2	2	38	5
car-starts	18	14	1	1	60	3
cpu	7	7	2	0	19	2
hprice	12	12	0	6	40	3
oesophagus	40	41	0	3	95	3
wisconsin	10	13	0	8	51	3

**Table 4**  
% of edges estimated with the EM algorithm whose signs do not match the prior knowledge.

$M$	$\rho$	autompg	boerlag.	car	car-starts	cpu	hprice	oesoph.	wiscon.	Av
50	0.2	22.8	42.2	100.0	50.1	7.8	15.5	11.5	18.5	33.5
50	0.3	23.5	43.9	100.0	49.9	7.8	15.7	11.7	17.5	33.7
50	0.5	28.0	45.0	99.3	47.9	8.4	20.2	12.0	18.5	34.9
100	0.2	8.5	37.4	100.0	55.8	6.0	11.3	10.4	14.9	30.5
100	0.3	8.8	38.4	100.0	55.9	5.3	11.7	10.7	15.7	30.8
100	0.5	13.3	41.2	100.0	55.4	6.2	12.2	10.9	15.4	31.8
500	0.2	0.0	30.6	100.0	66.1	6.0	4.7	8.1	9.4	28.1
500	0.3	0.3	30.3	100.0	66.4	6.2	5.3	8.1	8.6	28.1
500	0.5	1.5	31.4	100.0	66.4	5.6	5.8	8.7	9.8	31.5

The algorithms are evaluated over data sets artificially generated from these BNs by means of logic sampling. As commented above, we use this approach because it guarantees that the data used for learning the parameters of the BNs come from a distribution which satisfies the given qualitative influences. Data sets with different sample sizes were considered:  $M = 50$ ,  $M = 100$ , and  $M = 500$ . We then simulate missing observations as follows: for each data set we randomly select a subset of the variables, and then simulate missing observations only for these selected variables with a given probability  $\rho$ . Three different probabilities are considered:  $\rho = 0.2$ ,  $\rho = 0.3$  and  $\rho = 0.5$ . The subset of variables with missing observations was allowed to be different at each replication of the experiment and always contained half of the variable set.

Following this procedure, 72 evaluation settings are used by considering the eight BNs, the three sample sizes, and the three  $\rho$  values. One evaluation setting is then composed by 50 data sets generated from the same BN, with the same sample size and with the same missing observation probability.

## 5.2. EM vs. irEM

In a first analysis, we want to highlight a previously commented problem of using plain EM in a domain where experts know that some links have well-defined qualitative influences: the models learned contain many edges whose signs are in contradiction with the expert beliefs. In Table 4, we detail the percentage of the edges in the models learned with EM whose signs do not match the sign of the generating network. Note that if plain EM gets a sign wrong in just a single context  $\mathbf{s}$  (see equations (1) and (2)), then the sign becomes a “?” instead of positive or negative, and is considered to be wrong.

As can be seen, this proportion largely depends on the particular BN. In general, it tends to decrease with higher sample sizes and to increase with higher missing observation rates.<sup>4</sup> In many cases the proportion of wrongly inferred edge signs is large. This especially happens with the car network, where in almost all cases every edge has the wrong sign. If we look at Fig. 2(d), we can see why: the conditional probability table (CPT) of the variable “accept” involves 5 variables and, hence, a high number of contexts  $\mathbf{s}$  for which an order constraint has to be satisfied in order to obtain the correct sign.

As can be seen, EM might learn BNs whose signs differ from the qualitative influences expressed by the expert. This might cause the experts to reject the inferences provided by these graphical models because they will see that some parts of the model are in contradiction with their domain knowledge. Unlike plain EM, irEM and qirEM always produce models whose signs fully match those specified by the domain expert.

Next, let us compare how well irEM and plain EM approximate the true probability distribution. In Table 5(a) we show the results of the comparisons made using a Wilcoxon signed-rank test with  $\alpha = 0.01$  for paired samples when comparing the KL-divergence from the real BN of the models learned with irEM versus the models learned with plain EM in each of

<sup>4</sup> In some cases (e.g. cpu with  $M = 100$ ), this trend is broken. The reason is that the reported values are percentages of the total number of edges over 50 different data sets with different partially observed variable subsets. And the differences in terms of absolute number of edges between the different configurations where this trend does not hold are small. E.g., in cpu network, there are 0.54, 0.48 and 0.56 edge violations for  $M = 100$  and  $\rho = 0.2, 0.3, 0.5$ , respectively.

**Table 5**

Comparison of the KL-divergence/log-likelihood of irEM versus EM using a Wilcoxon signed-rank test with  $\alpha = 0.01$ . For each sample size and missing observation rate, we detail “L/E/H”, where L means the number of networks where irEM has statistically significant lower log-likelihood/KL-divergence than EM; E means the number of networks where both have equal log-likelihood/KL-divergence (i.e. there are not statistically significant differences); H means the number of networks where irEM has statistically significant higher log-likelihood/KL-divergence than EM. Last columns and last rows display aggregate information over columns and over rows, respectively.

$M/\rho$	0.2	0.3	0.5	
(a) KL-divergence comparison				
50	8/0/0	8/0/0	8/0/0	24/0/0
100	8/0/0	8/0/0	8/0/0	24/0/0
500	7/1/0	7/1/0	6/2/0	20/4/0
	23/1/0	23/1/0	22/2/0	68/4/0
(b) Log-likelihood comparison				
50	7/1/0	8/0/0	8/0/0	23/1/0
100	7/1/0	7/1/0	7/1/0	21/3/0
500	7/1/0	7/1/0	7/1/0	21/3/0
	21/3/0	22/2/0	22/2/0	65/7/0

**Table 6**

Average value of  $D(\text{Pr}_\theta \parallel \text{Pr}_{\hat{\theta}}) - D(\text{Pr}_\theta \parallel \text{Pr}_{\theta^*})$ , that is, the average difference between the KL-divergence of plain EM and the KL-divergence of irEM. As the proportion  $\rho$  of missing data increases the advantage of irEM becomes more pronounced.

$M/\rho$	0.2	0.3	0.5
50	0.0273	0.0301	0.0378
100	0.0139	0.0153	0.0201
500	0.0049	0.0054	0.0081

the evaluation settings.<sup>5</sup> The results are also aggregated over the eight BNs using the following notation: for each sample size and missing observation rate, we detail “L/E/H”, where L denotes the number of networks where irEM has a statistically significant lower KL-divergence than EM; E denotes the number of networks where both have equal KL-divergence (i.e. the difference is not statistically significant); H denotes the number of networks where irEM has a statistically significant higher KL-divergence than EM. The final column and final row of the table display aggregate information over columns and over rows, respectively.

When looking at the KL-divergence we find that irEM produces a better approximation of the true distribution than plain EM in 68 out of 72 of the experimental settings, and it is never worse. Hence, exploiting prior knowledge about the signs of qualitative influences really pays off. Looking at Table 5(b), we find that in most of the experimental settings the log-likelihood score of EM on the training sample is significantly higher than the log-likelihood score of irEM. Combining this observation with the findings presented in Table 5(a), this suggests that plain EM is over-fitting on the training sample. Only for the largest sample size we considered, we find that for some networks there is no statistically significant difference between irEM and plain EM. We may expect however, that the difference between irEM and plain EM becomes smaller as the sample size increases, since with large sample sizes the estimates of plain EM will tend to satisfy the order constraints emanating from the signs in the network anyway.

We additionally performed another extra Wilcoxon signed-rank test for each noise rate and sample size but using eight paired samples corresponding to the average log-likelihood (and KL-divergence) for each Bayesian network in order to compare the performance of irEM and EM with a lower number of statistical tests. Because the number of paired samples was low (i.e. eight in this case) we employed an alpha value equal to 0.05. The results obtained show the same conclusion: irEM gets a significantly lower KL-divergence and a significantly lower log-likelihood than EM for all the noise rates and sample sizes configurations.

In Table 6 we show the difference between the KL-divergence of plain EM and irEM for different proportions of missing data and samples sizes. The numbers illustrate that the advantage of irEM tends to increase for higher proportions of missing data. This supports our claim that the exploitation of qualitative influences is particularly helpful in case of incomplete data.

In the next analysis we want to evaluate which is the overall computational burden associated to irEM. For this purpose we detail in Table 7 the average ratio between the computational time of irEM and EM (i.e. ratio = Time(irEM)/Time(EM)) for each noise rate and sample size across the eight Bayesian networks. As can be seen, irEM is computationally heavier than EM, specially, for small sample sizes and small noise rates. However, this difference reduces for higher sample sizes

<sup>5</sup> Although the total number of performed tests in this experimental evaluation is high, 250 tests, we do not employ a false discovery rate (FDR) control method because the possible number of false positive errors is low, around 2.5, and would not alter the main conclusions of the experiments.

**Table 7**

Average computational time ratio between irEM and EM (i.e. ratio = Time(irEM)/Time(EM)), across the eight Bayesian networks and the 50 replicas, for the different sample size and noise rates.

$M/\rho$	0.2	0.3	0.5
50	16.75	12.54	8.95
100	6.89	4.94	3.69
500	1.93	1.63	1.31

**Table 8**

Comparison of the (a) KL-divergence and (b) log-likelihood of irEM and qirEM using a Wilcoxon signed-rank test with  $\alpha = 0.01$ . For each sample size and missing observation rate, we give "L/E/H", where L is the number of networks where irEM has statistically significant lower KL-divergence/log-likelihood than qirEM; E is the number of cases with no significant difference and H is the number of networks where irEM has statistically significant higher KL-divergence/log-likelihood than qirEM. The final column and final row display aggregate information over columns and rows, respectively.

$M/\rho$	0.2	0.3	0.5	
(a) Comparison of KL-divergence				
50	0/7/1	1/7/0	1/7/0	2/21/1
100	0/8/0	0/8/0	3/4/1	3/20/1
500	1/5/2	1/5/2	1/5/2	3/15/6
	1/20/3	2/20/2	5/16/3	8/56/8
(b) Comparison of Log-likelihood				
50	3/5/0	4/4/0	5/3/0	12/12/0
100	5/3/0	6/2/0	6/2/0	17/7/0
500	4/4/0	4/4/0	4/4/0	12/12/0
	12/12/0	14/10/0	15/9/0	41/31/0

and higher noise rates. This has a simple explanation. The computational cost of EM (irEM) directly depends of the cost of the E step and the cost of the (constrained) M step. The cost of the M steps depends on the number of samples, but it is simply a matter of counting and, for the normal M step, normalizing these counts at the end or, for the constrained M step, applying isotonic regression. The cost of the E step equals the cost of performing one inference in the model multiplied by the number of samples. And the cost of inference directly depends of the number of missing observations. So, when the number of observations and the noise rate increases the computational cost of the E step largely dominates the computational cost of the M step. In consequence, the computational cost of irEM and EM becomes more similar.

### 5.3. irEM vs. qirEM

In a second analysis, we evaluate how qirEM compares to irEM. In Table 8(a) and Table 8(b), we give the results of the comparisons between irEM and qirEM. The information is displayed similarly to the previous comparison between irEM and plain EM.

When comparing the KL-divergence of irEM and qirEM, we find that in most cases (56 out of 72) the difference is not statistically significant. There are cases where irEM performs better than qirEM, particularly in the autmpg and car networks. In other cases qirEM performs better than irEM, in particular in the car-starts and oesophagus networks. All in all, we may conclude that the two algorithms approximate the true probability distribution equally well.

As can be seen from Table 8(b), the log-likelihood of qirEM is significantly higher in many of the evaluation settings and never significantly worse. This may be considered surprising, since in general qirEM is not guaranteed to find a locally optimal solution. As discussed at the end of section 4.3, this finding may be due to the fact that irEM needs to traverse a parameter landscape which is, in many situations, more complex than the unconstrained one because, although smaller, the constrained parameter space contains new local optima on the boundaries of the order-compatible region. This might cause irEM to get trapped into worse constrained local optima which are ignored by plain EM because it moves in an unconstrained space. In any case, the differences we found between the log-likelihood of irEM and qirEM are very small, in comparison to the ones we found between irEM and plain EM.

Similarly to the previous comparison between irEM and EM, we additionally performed another extra Wilcoxon signed-rank test with  $\alpha = 0.05$  for each noise rate and sample size but using eight paired samples corresponding to the average log-likelihood (and KL-divergence) for each Bayesian network in order to compare irEM and qirEM with a lower number of statistical tests. And the results obtained show similar conclusions. irEM gets a non-significantly different KL-divergence with respect to qirEM for all the noise rates and sample sizes configurations. For the log-likelihood we get that irEM has non-different log-likelihood than qirEM except for noise rate 0.2 and 500 samples and noise rate 0.3 and 500 samples where irEM gets a lower log-likelihood than qirEM.

**Table 9**

Average number of iterations until convergence of irEM/qirEM.

$M/\rho$	0.2	0.3	0.5
50	6.99/7.37	8.75/9.28	13.86/15.24
100	6.52/6.93	8.29/9.03	13.93/14.64
500	6.32/6.75	8.40/9.20	13.39/15.83

**Table 10**

Average computational time ratio between irEM and qirEM (i.e. ratio = Time(irEM)/Time(qirEM)), across the eight Bayesian networks and the 50 replicas, for the different sample size and noise rates.

$M/\rho$	0.2	0.3	0.5
50	3.89	3.71	3.54
100	2.67	2.48	2.24
500	1.46	1.33	1.19

**Table 11**

Evaluation of the informed-irEM (see text for details).

Network	KL-divergence		% local optimum	Iterations inf-irEM
	inf-irEM	qirEM		
autompg	0.1026	0.1026	58	2
boerlage92	0.3554	0.3561	0	5.98
car	0.0817	0.0822	2	4.94
car-starts	0.1426	0.1431	16	5.68
cpu	0.0979	0.0980	68	1.7
hprice	0.2061	0.2056	26	3.48
oesophagus	0.5958	0.5954	2	3.62
wisconsin	0.2001	0.2005	16	3.58

To compare the computational burden of qirEM and irEM we firstly consider the number of iterations until convergence of both algorithms, displayed in Table 9. As can be seen, qirEM converges on average slightly slower than irEM. Only for high missing value rates ( $\rho = 0.5$ ) the differences in the number of iterations are larger than 1, which means that qirEM requires on average the application of a few more E-steps than irEM. However, as already commented, irEM needs the application of a constrained M-step at each iteration in contrast to qirEM which only applies this operation once. So, whether qirEM is in general quicker than irEM depends on the computational complexity of the E-step, which in turn depends on the structure of the BN and on the particular variables which have missing observations. In Table 10 we also display the average ratio between the computational time of irEM and qirEM (i.e. ratio = Time(irEM)/Time(qirEM)) for the evaluated Bayesian networks. As can be seen, irEM is overall computationally heavier than qirEM. But, as in the previous comparison between irEM and EM, these differences reduces for higher sample sizes and noise rates because, as previously commented, the computational cost of the E step dominates the computational cost of the constrained M step.

As we mentioned at the end of section 4.3, unlike irEM, qirEM is not guaranteed to converge to locally optimal solutions in the constrained parameter space. In the final experiment we try to measure how far the solutions given by qirEM are removed from a local optimum in the order-compatible space. For this purpose, we perform the following experiment: we run qirEM until convergence over data sets with 100 samples and  $\rho = 0.3$ ; and we use this solution  $\hat{\theta}_F^*$  as the initial parameter value of irEM,  $\theta^0 = \hat{\theta}_F^*$ , which is then run until convergence. If  $\hat{\theta}_F^*$  is a local optimum in the order-compatible space, then this *informed irEM* (inf-irEM) will converge in a single iteration, otherwise it will move further in the order-compatible space until converge to a nearby local optimum. Table 11 shows, disclosed for each BN, the KL-divergence of the solutions found by inf-irEM and by qirEM; the percentage over the 50 repetitions of the experiment where qirEM converges to a local optimum; and the number of iterations until convergence of inf-irEM. As can be seen, for some networks qirEM often converges to a stationary point. However, when comparing the KL-divergence of qirEM and inf-irEM, we find that there are no statistically significant differences for any of the networks under the Wilcoxon signed-rank test with  $\alpha = 0.01$ . That is to say, in terms of the generalization capacity measured via the KL-divergence, these results suggest that the parameter values produced by qirEM are not worse than those corresponding to a nearby local optimum of the log-likelihood function (i.e. those obtained by inf-irEM). Hence, although in some cases qirEM does not converge to a stationary point, it seems that in these cases is not worthwhile to continue with further maximization efforts.

## 6. Conclusions and future works

Building upon the observation that domain experts can quite readily provide qualitative information about the influences between the variables of a Bayesian network, we presented two algorithms, irEM and qirEM, for learning the parameters of a network from incomplete data that take such knowledge into account. Both methods take an EM approach extended



with isotonic regression but employ different optimization strategies. The two methods both result however, in Bayesian networks that exhibit the given qualitative influences and hence in networks that are more likely to be accepted by their domain users than networks that violate common knowledge about the directions of influence. By means of experiments, we demonstrated that both methods serve to improve the quality of the parameters learned from incomplete data, in the sense of correcting parameter estimates obtained by traditional methods that violate the constraints from the given qualitative influences. The results show more specifically that the improvement is the stronger for data sets with larger proportions of missing values; from the experiments we found moreover that the standard EM algorithm often results in violating estimates from such data sets.

We found that irEM and qirEM approximate the true probability distribution equally well. From a computational view-point, irEM has the advantage that it tends to converge somewhat faster, but on the other hand it has to perform more work than qirEM in each iteration. Which of these two is dominant depends very much on the properties of the data set, such as the proportion of missing data.

One possible direction of further research is to extend the algorithms presented here to non-binary data. In [12] Feelders analyzes a method for parameter estimation with order constraints emanating from known positive and negative qualitative influences for non-binary data. The estimates are computed by performing a number of isotonic regressions, one for each value (except for the highest value) of the child variable. This estimation procedure can probably be made to work for incomplete data in a similar fashion as it was done in this paper for binary data.

Another interesting line of future research would be to exploit the presence of qualitative influences when learning the structure of the Bayesian network from (partially observed) data. Many expert knowledge could be given in terms of positive/negative correlations between two variables which might not have a causal and direct connection between them, as we assume in this paper. How to exploit this knowledge when learning the structure of the network is a completely open problem. For example, one should consider the constraints induced by the provided qualitative influences when evaluating the goodness of a given graph structure. Moreover, one should also consider how the qualitative influences propagate through the network [26] to further exploit these constraints.

## Acknowledgements

This work was jointly supported by the research programme Consolider Ingenio 2010 under projects CSD2007-00018, by the Spanish Ministry of Economy and Competitiveness under project TIN2013-46638-C3-2-P and the European Regional Development Fund (FEDER).

## References

- [1] R.K. Ahuja, J.B. Orlin, A fast scaling algorithm for minimizing separable convex functions subject to chain constraints, *Oper. Res.* 49 (5) (2001) 784–789.
- [2] E.A. Altendorf, A.C. Restificar, T.G. Dietterich, Learning from sparse data by exploiting monotonicity constraints, in: F. Bacchus, T. Jaakkola (Eds.), *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence, UAI-05*, AUAI Press, 2005, pp. 18–25.
- [3] P.M. Anglin, R. Gençay, Semiparametric estimation of a hedonic price function, *J. Appl. Econom.* 11 (6) (1996) 633–648.
- [4] A. Ben-Tal, A. Nemirovski, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, vol. 2, SIAM, 2001.
- [5] B. Boerlage, Link strength in Bayesian networks, Technical report 94-17, Dept. of Computer Science, Univ. of British Columbia, 1992.
- [6] S.G. Boettcher, C. Dethlefsen, deal: a package for learning Bayesian networks, *J. Stat. Softw.* 8 (20) (2003) 1–40, 12.
- [7] S. Boyd, L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.
- [8] C.P. de Campos, Y. Tong, Q. Ji, Constrained maximum likelihood learning of Bayesian networks for facial action recognition, in: *ECCV* (3), 2008, pp. 168–181.
- [9] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, *J. R. Stat. Soc. B* 39 (1) (1977) 1–38.
- [10] M.J. Druzdzel, Probabilistic reasoning in decision support systems: from computation to common sense, Ph.D. thesis, Department of Engineering and Public Policy, Carnegie Mellon University, 1993.
- [11] R.L. Dykstra, T. Robertson, An algorithm for isotonic regression for two or more independent variables, *Ann. Stat.* 10 (3) (1982) 708–716.
- [12] A. Feelders, A new parameter learning method for Bayesian networks with qualitative influences, in: R. Parr, L.C. van der Gaag (Eds.), *Proceedings of Uncertainty in Artificial Intelligence, UAI07*, AUAI Press, 2007, pp. 117–124.
- [13] A. Feelders, L.C. van der Gaag, Learning Bayesian network parameters with prior knowledge about context-specific qualitative influences, in: *Proceedings of the Twenty-First Conference Annual Conference on Uncertainty in Artificial Intelligence, UAI-05*, AUAI Press, Arlington, Virginia, 2005, pp. 193–200.
- [14] A. Feelders, L.C. van der Gaag, Learning Bayesian network parameters under order constraints, *Int. J. Approx. Reason.* 42 (1–2) (2006) 37–53.
- [15] A. Frank, A. Asuncion, *UCI machine learning repository*, 2010.
- [16] D. Geiger, C. Meek, Graphical models and exponential families, in: *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers Inc., 1998, pp. 156–165.
- [17] D. Heckerman, J.S. Breese, K. Rommelse, Decision-theoretic troubleshooting, *Commun. ACM* 38 (3) (1995) 49–57.
- [18] D. Heckerman, D. Geiger, D.M. Chickering, Learning Bayesian networks: the combination of knowledge and statistical data, *Mach. Learn.* 20 (3) (1995) 197–243.
- [19] E.M. Helsen, L.C. van der Gaag, F. Groenendaal, Designing a procedure for the acquisition of probability constraints for Bayesian networks, in: E. Motta, N.R. Shadbolt, A. Stutt, N. Gibbins (Eds.), *Engineering Knowledge in the Age of the Semantic Web: 14th International Conference*, Springer, 2004, pp. 280–292.
- [20] D. Koller, N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*, MIT Press, 2009.
- [21] J. Lasserre, C.M. Bishop, Generative or discriminative? Getting the best of both worlds, *Bayesian Stat.* 8 (2007) 3–24.
- [22] W. Liao, Q. Ji, Learning Bayesian network parameters under incomplete data with domain knowledge, *Pattern Recognit.* 42 (11) (2009) 3046–3056.
- [23] R.M. Neal, G.E. Hinton, A view of the EM algorithm that justifies incremental, sparse, and other variants, in: M.I. Jordan (Ed.), *Learning in Graphical Models*, MIT Press, Cambridge, MA, USA, 1999, pp. 355–368.

- [24] T. Robertson, F. Wright, R.L. Dykstra, *Order Restricted Statistical Inference*, Wiley, 1988.
- [25] Q.F. Stout, Isotonic regression via partitioning, *Algorithmica* 66 (1) (2013) 93–112.
- [26] Linda C. van der Gaag, Hans L. Bodlaender, Ad Feelders, Monotonicity in Bayesian networks, in: *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, AUAI Press, 2004, pp. 569–576.
- [27] F. Wittig, A. Jameson, Exploiting qualitative knowledge in the learning of conditional probabilities of Bayesian networks, in: C. Boutilier, M. Goldszmidt (Eds.), *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, 2000, pp. 644–652.
- [28] C.F. Wu, On the convergence properties of the EM algorithm, *Ann. Stat.* 11 (1) (1983) 95–103.