Efficient implementation of compositional models for data mining

Václav KRATOCHVÍL[1], Radim JIROUŠEK[1], and Tzong-Ru LEE[2]

[1] *Czech Academy of Sciences, Inst. Information Theory and Automation*
Pod Vodárenskou věží 4, CZ-182 08 Praha 8,
Czech Republic
velorex@utia.cas.cz, radim@utia.cas.cz

[2] *Department of Marketing, National Chung Hsing University*
145 Xingda Rd., South Dist., Taichung City 402,
Taiwan (R.O.C.)
trlee@dragon.nchu.edu.tw

Abstract

A compositional model encodes probabilistic relationships among variables of interest. In connection with various statistical techniques, it represents a practical tool for data modeling and data mining. Structure of the model represents (un)conditional independencies among all variables. Relationships of dependent variables are described by low-dimensional probability distributions. Having a compositional model, a data miner can easily apply an intervention on variables of interest, fix values of other variables (conditioning), or to narrow the context of a problem (marginalization). The model learning process can be controlled to avoid overfitting of data.

In this paper, we present a new semi-supervised web application that will enable researchers to design probabilistic (compositional) models (both causal and stochastic). Thanks to the web architecture of the system, the researchers will always have a possibility to influence the data-based model construction process from any place of the world. It is also expected that the application of this methodology to practical problems will open new problems that will be an inspiration for further theoretical research.
Keywords: Data mining, Mutual information, Compositional model, Conditional independence, Probability theory.

1 Introduction

This paper is a first introduction to a new computer system trying to implement the theory of compositional models for data-mining. We hope that it will attract researchers to apply the theory and encourage them in studying it because there are still blanks to be filled.

The basic idea of compositional models is very simple: it is beyond human capabilities to represent/express/understand global knowledge of an application area - one always has to work with pieces of local knowledge only. Such local knowledge can be, within probability theory, easily represented by a low-dimensional distribution. It should be stressed out that, based on the laws of mathematical statistics, it is evident that the dimensionality of the estimated distributions is strictly limited by the application of data-based models. Whatever size of data is at our disposal, we can hardly expect to obtain reliable estimates of probabilities of a 20-dimensional distribution (even for binary variables). Typically, one can assume that dimensionality of the considered distributions is between 2 and 8.

1.1 Compositional models

When pieces of local knowledge are represented by low-dimensional distributions, the global knowledge should be represented by a multidimensional probability distribution. The technique of compositional models describes directly how the multidimensional distribution is computed/composed from a system of low-dimensional distributions. Usually, one starts constructing such a model from a (usually great) number of low-dimensional distributions. Such a model resembles a jig-saw

puzzle that has a large number of parts, each bearing a local piece of a picture. The goal is to figure out how to assemble them in a way that the global picture makes sense and reflects all of the individual small parts. The only difference is that we look for a linear ordering of distributions in our case.

1.2 Data mining

Generally, data mining is understood to be the process of discovering patterns in large data sets involving various methods from machine learning, statistics, and database systems. The goal of a data mining process is not a model itself but its interpretation in the form of a distilled knowledge.

One possible usage of compositional models in data mining stands in the process of model construction. For example, constructing compositional models from two data files collected in different cultural environments enables the user to compare the structures of the two models, revealing qualitative differences between the studied societies, and the comparison of the respective probability tables enables the researchers to describe the quantitative differences. The already mined data can serve also in the opposite direction. The supervised approach to model construction enables the researchers to influence the resulting models in the way that these models are easily comprehensible and interpretative. The user can have some knowledge about data, based on which the model is constructed.

1.3 Notation

In this paper, we consider only finite-valued variables, which are denoted by upper-case Latin characters. Groups of variables are denoted by bold-face characters: i.e., for example, $\mathbf{M} = \{X, Y, Z, W\}$. The set of values of variable X is denoted by $\mathbb{X}_X$. Similarly, we use $\mathbb{X}_{\mathbf{M}}$. Generally, we use the same notation as in the second paper by same authors in this proceedings [1].

Let us highlight that by a *state* of a group of variables we understand any combination of values of the respective variables.

Probability distributions are denoted by characters of Greek alphabet (κ, ν, π). To highlight that the given probability distribution is defined over variables a set of variables $\mathbf{K}$ we write $\kappa(\mathbf{K})$ A *marginal distribution* of $\kappa(\mathbf{K})$ defined for variables $\mathbf{L}$ is denoted as $\kappa^{\downarrow\mathbf{L}}$.

2 Compositional models

The key element of the theory of compositional models is the operator of composition. To be able to introduce these models, let us briefly recall its definition and a couple of its most important properties (to read more about basic properties of the operator of composition, we refer the reader to [2] and [3]).

For arbitrary two distributions $\kappa(\mathbf{K})$ and $\lambda(\mathbf{L})$, for which [1] $\kappa^{\downarrow\mathbf{K}\cap\mathbf{L}} \ll \lambda^{\downarrow\mathbf{K}\cap\mathbf{L}}$ is their composition given by the following formula[2]

$$(\kappa \triangleright \lambda)(x) = \frac{\kappa(x^{\downarrow\mathbf{K}})\lambda(x^{\downarrow\mathbf{L}})}{\lambda^{\downarrow\mathbf{K}\cap\mathbf{L}}(x^{\downarrow\mathbf{K}\cap\mathbf{L}})}. \tag{1}$$

Otherwise, the composition remains undefined.

The operator of composition is used to construct multidimensional compositional models. Composing two distributions, we can define a distribution of a higher dimensionality than any of the original ones.

By a compositional model of a multidimensional probability distribution we understand a sequence of low-dimensional distributions that assembled together using the operator of composition somehow models the original multidimensional distribution that would be difficult to handle otherwise.

Denoting the low-dimensional distributions $\kappa_1(\mathbf{K}_1)$, $\kappa_2(\mathbf{K}_2), \ldots, \kappa_n(\mathbf{K}_n)$, we get the compositional multidimensional model by the application of the operator of composition $\triangleright$ to this sequence

[1] $\kappa(\mathbf{M}) \ll \lambda(\mathbf{M})$ denoted that the distribution κ is absolutely continues with respect to distribution λ, which in our finite settings means that whenever κ is positive also λ*mustbepositive*.

[2] Define $\frac{0\cdot 0}{0} = 0$.

from left to right: $\kappa_1 \triangleright \kappa_2 \triangleright \ldots \triangleright \kappa_n$. (Note that the operator is neither commutative nor associative.) This sequence of probabilistic distributions, if all compositions are defined, is called the *generating sequence* of the compositional model. The sequence of sets of variables the generating sequence is defined for – $\mathbf{K}_1, \mathbf{K}_2, \ldots, \mathbf{K}_n$ – is called a compositional model *structure*. To visualize the structure we use a tool called a *persegram* - a table of markers where rows correspond to variables and columns to sets from the structure in the given ordering. A position in the table is marked if the variable is among variables of the respective variables sets. Markers for the first occurrence of each variable (i.e., the leftmost markers in rows) are box-markers, and for other occurrences there are bullets.

We say that a generating sequence is perfect if all elements are marginals of the resulting multidimensional distribution. It is worth noting that among all models, perfect models play an important role because they faithfully reflect the information contained in the individual distributions. This property is thus important from the point of view of potential applications: when the individual low-dimensional distributions κ_i represent pieces of local knowledge, then $\kappa_1 \triangleright \kappa_2 \triangleright \ldots \triangleright \kappa_n$ is a proper representative of global knowledge.

Note that compositional models can be also used for the representation of causal models [4]. In that case, interventions can be easily modeled by composing the model with a simple degenerated probability distribution.

3 Implementation

It is always beneficial for a theoretical work to experiment with a real problem. To evaluate various hypotheses and to support further theoretical research, it is necessary to have an experimental tool for calculations with compositional models. In this section, we would like to describe the basic ideas standing behind the implementations. As already written above, we have implemented a semi-supervised web application that enables a researcher to design the probabilistic compositional model (both causal and stochastic). We assume that the web architecture of the system will make it easily accessible and open for the wide range of audience.

The system is implemented in R environment [5] using Shiny web application framework [6] and data.table package [7].

3.1 Probability distribution

The key problem is the representation of a probability distribution in a computer memory. Because we restricted random variables to finitely valued discrete variables only, it seemed like a natural step to store probability distributions not as multidimensional arrays (hypercubes) where individual dimensions correspond to random variables, but as a listing of states in form of a table. Actually, we have been inspired by relational database theory [8]. We store a probability distribution as a database table - a *data.table* [7] in R.

Example 1 *Having a probability distribution over e.g. three random variables Y, Z, W - the corresponding table has four columns. First three columns correspond to random variables, the last column corresponds to an amount of probability dedicated to each row. Every row represents a unique state - a combination of values of the respective random variables. In case of a zero probability state, it does not have to be listed in the table.*

Y	Z	W	probability
0	0	1	0.3
0	1	1	0.4
0	1	0	0.1
1	0	1	0.2

Table 1: Representation of $\lambda(Y, Z, W)$ - table `lambda`

It turns out that this approach is convenient for basic computations needed when working with compositional models.

3.1.1 Marginalization

It appears that marginalization procedure is a simple example of aggregation in relational databases. Indeed, to calculate $\lambda^{\downarrow Y,Z}$ from $\lambda(X,Y,Z)$ defined in Example 1, it is enough to provide the following query:

```
SELECT Y, Z, SUM(probability) AS probability
        FROM lambda
        GROUP BY (Y,Z);
```

It results in the following table

Y	Z	probability
0	0	0.90
0	1	0.10

Table 2: $\lambda^{\downarrow Y,Z}$ - table `lambda_marginal`

3.1.2 Composition operator

In case of the operator of composition, the situation is quite similar. We split the fraction from (1) into a product of two terms:

$$(\kappa \triangleright \lambda)(x,y,z,w) = \frac{\kappa(y,z,w)\lambda(x,y,z)}{\lambda^{\downarrow Y,Z}(y,z)} = \big(\kappa(y,z,w)\big)\frac{\lambda(x,y,z)}{\lambda^{\downarrow Y,Z}(y,z)}$$

The term

$$\bar{\lambda} = \frac{\lambda(x,y,z)}{\lambda^{\downarrow Y,Z}(y,z)}$$

can be computed using the following SQL query:

```
COPY lambda INTO lambda_bar;

UPDATE lambda_bar
    SET probability = probability / SUM(probability)
    GROUP BY (Y,Z);
```

The, using $\bar{\lambda}$, one can proceed with a standard JOIN operator. To guarantee that the operator is defined, i.e. whether $\forall y \in \mathbb{X}_{\mathbf{K}\cap\mathbf{L}}\ \ \kappa^{\downarrow \mathbf{K}\cap\mathbf{L}}(y) > 0 \implies \lambda^{\downarrow \mathbf{K}\cap\mathbf{L}}(y) > 0$ it is enough to use LEFT OUTER JOIN operator and then check whether the resulting table does not contain NULL value in columns corresponding to variables $K \setminus L$. If it is the case then the composition remains undefined and the compositional process is stopped.

In case of probability distributions $\kappa(Y,Z,W)$ and $\lambda(X,Y,Z)$ the query has the following form:

```
SELECT X, Y, Z, W, kappa.probability * lambda_bar.probability AS probability
        FROM kappa LEFT OUTER JOIN lambda
        USING (Y, Z);
```

Example 2 *To illustrate the operator of composition, we will compose probability distribution κ from Table 3b with distribution λ from Example 1. The updated $\bar{\lambda}$ distribution can be found in Table 3a and the result corresponding to distribution $(\kappa \triangleright \lambda)$ is in Table 3c.*

3.1.3 Point-wise multiplication

For other more advanced operations with compositional models (like anticipating operator [2]), we also need the so-called point-wise multiplication of two probability distributions. The point-wise multiplication can be easily implemented using CROSS JOIN operator. In the case of κ and λ from previous Examples, the result is in Table 3c and the query has the following form:

```
SELECT X,Y,Z,W kappa.probability*lambda.probability AS probability
            FROM kappa CROSS JOIN lambda
            USING (Y,Z);
```

Y	Z	W	probability
0	0	1	1.0
0	1	1	0.8
0	1	0	0.2
1	0	1	1.0

(a) $\bar{\lambda}(Y, Z, W)$ - table `lambdabar`

X	Y	Z	probability
0	0	0	0.20
0	0	1	0.15
1	0	1	0.25
1	0	0	0.40

(b) $\kappa(X, Y, Z)$ - table `kappa`

X	Y	Z	W	probability
0	0	0	1	0.20
1	0	0	1	0.40
0	0	1	1	0.12
0	0	1	0	0.03
1	0	1	1	0.20
1	0	1	0	0.05

(c) Representation of $(\kappa \triangleright \lambda)(X, Y, Z, W)$ - table `composition`

Table 3: Process of table composition

X	Y	Z	W	Probability
0	0	0	1	0.060
1	0	0	1	0.120
0	0	1	1	0.060
0	0	1	0	0.015
1	0	1	1	0.100
1	0	1	0	0.025

Table 4: $\kappa \cdot \lambda$

3.2 Compositional model

Every compositional model is fully defined using its *generating sequence* - the sequence of low-dimensional probability distributions that composed together from left to right using operator of composition $\triangleright$ create a *multidimensional probability distribution.* To represent the compositional model in a computer memory, it is enough and desirable to keep its generating sequence. In the case of e.g. marginalization, conditioning, perfectization (conversion of a model to a perfect one representing the same multidimensional distribution), etc. of the model, all computations are made locally. E.g. algorithm to perform marginalization of a compositional model locally using its generating sequence can be found in [9]. In case of conditioning, it has been proven that the conditioning process is easy if the compositional model is perfect (see [2]) and decomposable [3]. Note that decomposability is a structural property - the model is decomposable if the sequence variables meet the running intersection property.

To simplify the calculations, we keep the structure of the model aside. The reason is simple. For example, the marginalization process of a compositional model employs several heuristics that depend on structure only [10].

So far, we have implemented the following methods to work with compositional models:

- *marginalization* - removal of given variables from the given model
- *perfectization* - conversion of the model to a perfect one
- to perform *interventions*
- conversion to a *decomposable model*
- *conditioning* by a variable value

3.2.1 Learning

The first step in using the compositional model in case of data-mining is learning it from data. The learning process can be split into two parts. In the first part, a model structure is found. In this particular case, we have used hill climbing (HC) [11] algorithm. The main reason is that its implementation is a good trade-off between CPU requirements, the accuracy of the obtained model, and ease of implementation. Note that this method guarantees to obtain a minimal independence relations map and therefore it is especially appropriate to deal with high dimensional domains.

The second part of the compositional model learning process states in the estimating of the low-dimensional probabilistic distributions over a given set of variables (from the already learned model structure). In this case, we simply use frequencies of given states.

4 Web application

In this paper, we present a new semi-supervised web application called MUDIM online where acronym MUDIM stands for a system for MULti DImensional Models. Thanks to the web architecture of the system, the researchers will always have a possibility to influence the data-based model construction process from any place of the world without the need of any restriction. The application can be found at `http://gogo.utia.cas.cz/mudim`.

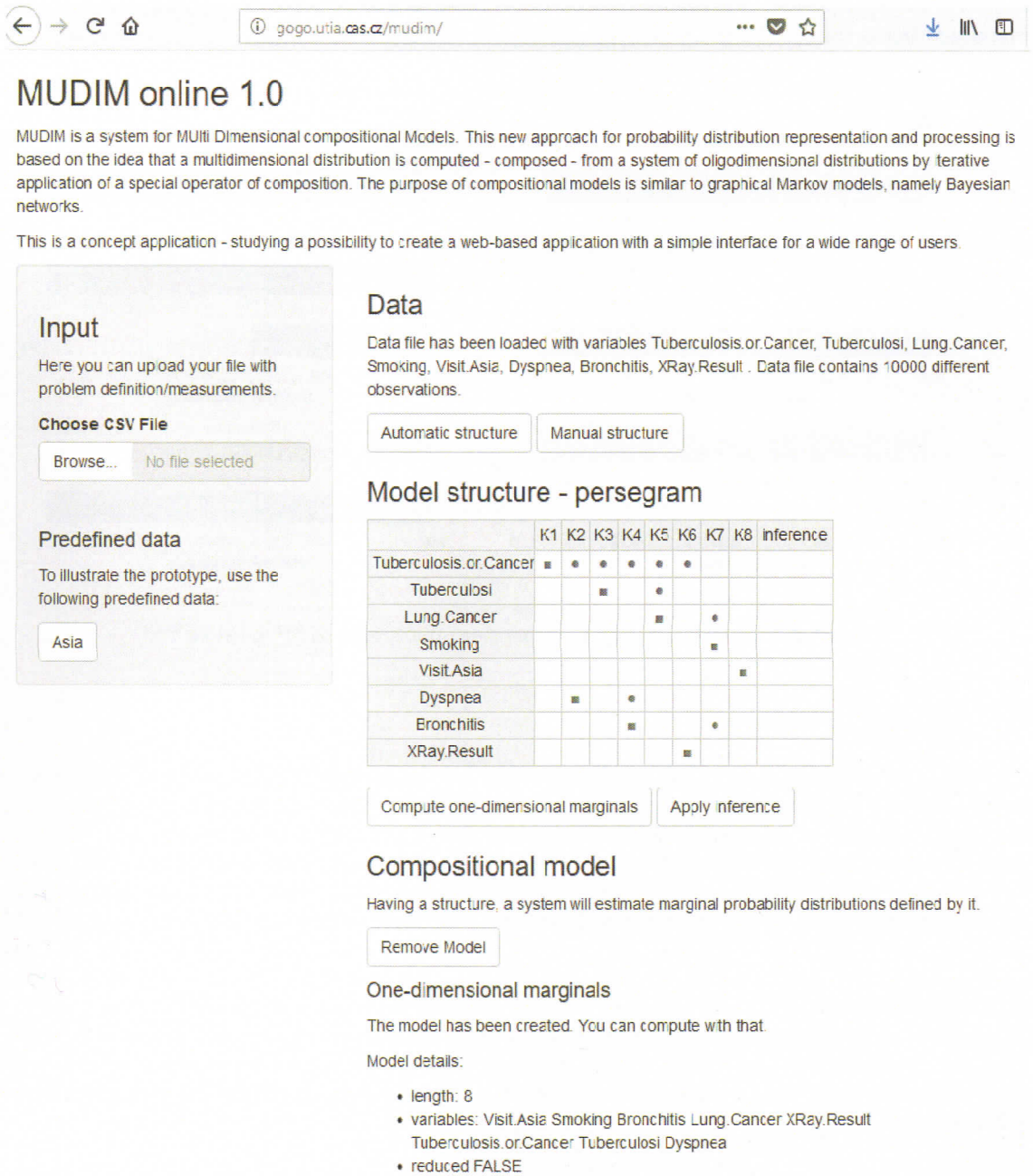

Figure 1: The look of MUDIM online web application

Having a compositional model, a data miner can easily apply an intervention on variables

of interest, fix values of other variables (conditioning), or to narrow the context of a problem (marginalization). To do that, the application expects a data file for observations/measurement at its input. The file has to be comma-separated values (CSV) file.

The basic interface of MUDIM online is visualized on Figure 1. The user can read that a data file has been loaded. The problem at hand has 8 random variables. To learn the structure of the model, the user can choose between automatic and manual mode. The structure is depicted using a persegram.

When the structure is finished, the model can be created. By creating the model we understand that corresponding low-dimensional probability distributions are estimated from data and aligned in a generating sequence in the order given by the structure. In case of the problem from Figure 1, the model has 8 low-dimensional probability distributions in its generating sequence. Reading the persegram, each distribution has dimension maximally 3. When the model is created the user is allowed to perform intervences, fix the value of a random variable by conditioning, display one-dimensional marginals (see Figure 2), etc.

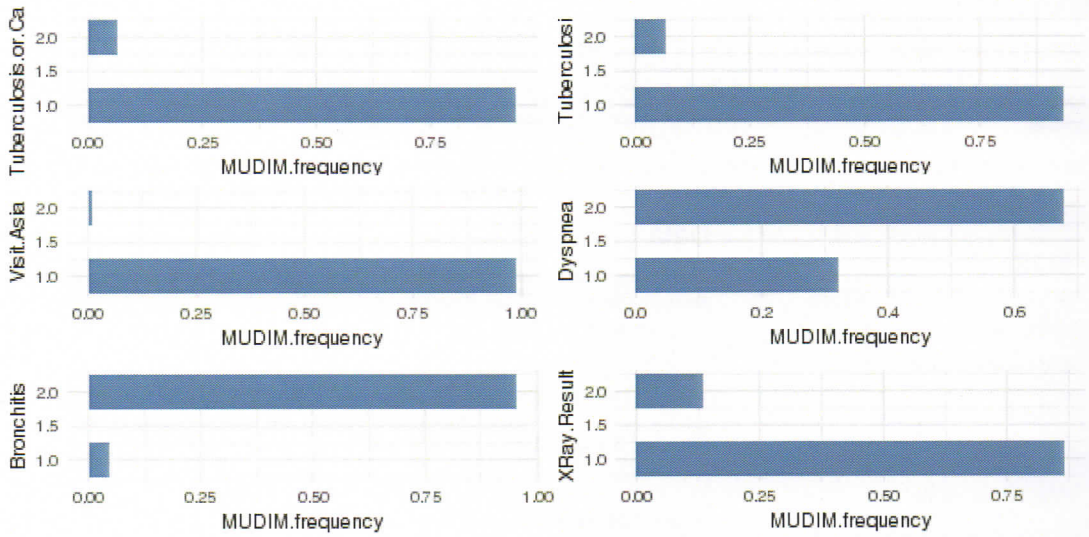

Figure 2: Visualisation of one-dimensionl marginals in MUDIM online

5 Future plans

There is a lot of work to do. First of all, we would like to implement all theoretical foundations known so far. We would like to implement more methods of learning the structure from data, information-theoretic notions like Kullback-Leibler divergence, multi-information, etc. We have several ideas on how the model learning process can be controlled to avoid overfitting of data - the system needs that for sure. Last but not least we hope that the application of this methodology to practical problems will open new problems that will be an inspiration for further theoretical research.

Acknowledgement
This work was supported by the Czech Science Foundation project 16-12010S. The authors would also like to thank the Czech Academy of Sciences and the Taiwanese Ministry of Science and Technology for their financial support of bilateral project "**Compositional Models for Data Mining**".

References

[1] Radim Jiroušek, Václav Kratochvíl, and Tozng-Ru Lee. Compositional models for data mining: an example. In *Proceedings of the 21 Czech-Japan Seminar on Data Analysis and Decision*

Making, 2018.

[2] Radim Jiroušek. Foundations of compositional model theory. *International Journal of General Systems*, 40(6):623–678, 2011.

[3] Radim Jiroušek and Václav Kratochvíl. Foundations of compositional models: structural properties. *International Journal of General Systems*, 44(1):2–25, 2015.

[4] Radim Jiroušek. Brief introduction to causal compositional models. In *Causal Inference in Econometrics*, pages 199–211. Springer, 2016.

[5] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2018.

[6] Winston Chang, Joe Cheng, JJ Allaire, Yihui Xie, and Jonathan McPherson. *shiny: Web Application Framework for R*, 2018. R package version 1.1.0.

[7] Matt Dowle and Arun Srinivasan. *data.table: Extension of 'data.frame'*, 2018. R package version 1.11.4.

[8] Edgar F Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387, 1970.

[9] Radim Jiroušek. Marginalization in composed probabilistic models. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 301–308. Morgan Kaufmann Publishers Inc., 2000.

[10] Vladislav Bína and Radim Jiroušek. Marginalization in multidimensional compositional models. *Kybernetika*, 42(4):405–422, 2006.

[11] José A. Gámez, Juan L. Mateo, and José M. Puerta. A fast hill-climbing algorithm for bayesian networks structure learning. In Khaled Mellouli, editor, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 585–597, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.