Recent Advances in Intelligent Information Systems
ISBN 978-83-60434-59-8, pages 443–456

A Comparison of Structural Distance Measures for Causal Bayesian Network Models

Martijn de Jongh[1] and Marek J. Drużdżel[1,2]

[1] Decision Systems Laboratory, School of Information Sciences and Intelligent Systems Program, University of Pittsburgh, Pittsburgh, PA 15260, USA
[2] Faculty of Computer Science, Białystok Technical University, Wiejska 45A, 15-351 Białystok, Poland

Abstract

We compare measures of structural distance between both, Bayesian networks and equivalence classes of Bayesian networks. The main application of these measures is in learning algorithms, where typically the interest is in how accurately a gold standard structure is retrieved by a learning algorithm. Structural distance measures can be especially useful when looking for causal structures. We discuss desirable properties of measures, review existing measures, and show some of our empirical findings concerning the performance of these metrics in practice.

Keywords: Bayesian networks, causal discovery, structure learning, structural distance.

1 Introduction

Bayesian networks (BNs) (Pearl, 1988) are efficient representations of joint probability distributions. They factorize the joint distribution over a graph structure into a product of conditional probability distributions. This can significantly decrease the number of necessary parameters and make probabilistic modeling and inference in large problem domains feasible. Because they are acyclic directed graphs, they offer a convenient representation of causal relations. BNs have been applied to a wide variety of domains, including medical diagnosis, computer vision, hardware fault diagnosis, and speech recognition.

One way of creating a Bayesian network is by learning it from a data set. There exist learning algorithms for both graph structure and probability parameters, of which learning the structure is harder and, arguably, more critical.

A typical approach to evaluating the performance of a BN structure learning algorithm is to start with an existing network, generate a data set from the joint probability distribution that it represents, and, subsequently, use the algorithm to retrieve the original structure. The main advantage of this approach is that the result generated by a structure learning algorithm can be compared with the structure of the original network, the gold standard, that generated the data set.

A critical element of this approach is a measure of structural distance between two graphs: the smaller this distance, the better the algorithm was in retrieving

the original structure. When causality comes into play, the orientation of edges in a learned network becomes extremely important. The assumption in this case is that when there is an edge from node a to node b, a is a cause of b. Causal BNs allow for prediction of the effects of manipulation of variables. When a variable is forced to a certain value, this will normally affect only its descendants in the graph.

When establishing causal relations is not of essence, one can be more relaxed about edge orientations in the structure. For example, in a classical machine learning application like classification, it is immaterial what the exact BN structure looks like, as long as the BN performs well as a classifier. The exact topology of the network will merely have an impact on the number of necessary parameters and the quality of predictions, tested typically by KL divergence (Friedman, 1998), cross-entropy (Singh, 1997) or cross-validation (Madden, 2003).

2 Independence Structures

Bayesian networks can be viewed as consisting of two parts: (1) an acyclic directed graph (ADG) describing conditional independence relations between modeled variables, and (2) conditional probability distributions (CPDs) over each variable given its parents[1] in the graph.

Any joint probability distribution can be factored into a product of conditional probability distributions. This can be achieved by applying the chain rule:

$$P(x_1, x_2, \ldots, x_n) = P(x_1)\, P(x_2 | x_1) \cdots P(x_n | x_1 x_2 \ldots x_{n-1}). \tag{1}$$

This factorization can be represented by an ADG. If a variable x_i is represented by a node x_i in the graph, the conditioning variables of x_i are represented as parents of the node x_i. If two variables x_i and x_j are (conditionally) independent of each other, neither of the two will appear in the other's conditioning set and, thus, neither can be a parent of the other in the ADG. Removing a variable x_j from the conditioning set of a variable x_i is represented in an ADG by removing the directed edge from node x_j to x_i. If an ADG represents a factorization of a joint probability distribution, then this joint probability distribution can be recovered by applying Eq. 2.

$$P(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} P(x_i | Pa(x_i)), \tag{2}$$

where $Pa(x_i)$ denotes the parents of x_i in the graph.

Generally, the more independencies present in the probability distribution are represented in the graph of the Bayesian network, the fewer parameters will be needed to represent the full joint probability distribution.

Typically, a joint probability distribution can be represented by several Bayesian networks. A trivial example of this are the graphs and probability distribution decompositions that can be acquired by applying Eq. 1. There are $n!$ possible

[1] In an ADG, a node a is a parent of a node b if there is a directed edge from a to b.

orderings of n variables and, thus, there are $n!$ possible Bayesian networks that can represent this distribution. This is formalized by the concept of structural equivalence:

Definition 1 (Equivalence) (Chickering, 1995)**.** *Two network structures are equivalent if the set of distributions that can be represented by one of the ADGs is identical to the set of distributions that can be represented by the other.*[2]

A Bayesian network structure encodes certain conditional independencies. For another Bayesian network to be considered equivalent, its structure must encode the same independencies. However, this does not mean that the structures of the networks will be identical. If the graph of a Bayesian network represents the (conditional) independencies of a joint probability distribution, it is called an I-map (independency mapping) (Pearl, 1988) for the distribution. Fig. 1 shows three simple examples of BN graph structures that are all I-maps for the same distribution.

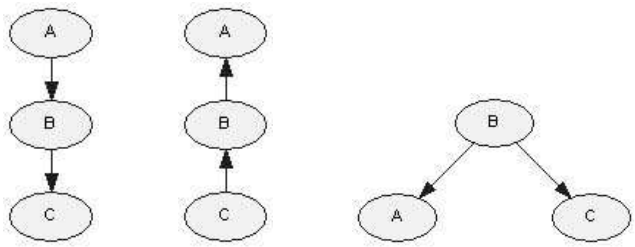


FIGURE 1: Example Bayesian networks that are I-maps for the same distribution.

The possibility of different graph structures representing the same independence relations is troubling when learning structures from data. Since, in general, there exist multiple structures that can accurately represent the independence relations found in the data, one must be chosen. When looking for a causal structure, this problem becomes serious. In the causal context, an edge signifies a causal influence of one variable on another and its direction is important. The inability to distinguish between graphs is known as statistical indistinguishability (Spirtes *et al.*, 1993). It is not possible, in general, to discover the orientation of all edges of a network, since the orientation of some edges may be immaterial to the set of independencies that the graph represents. Some substructures of graphs are recognizable. Fig. 2 shows a network structure, known as a v-structure, which is uniquely identifiable and is a sole member of its equivalence class. Finding v-structures is the key of the constraint-based search approach to structure learning (Spirtes *et al.*, 1993). V-structures are also the key to defining equivalence classes over Bayesian network structures. The following theorem summarizes this.

Theorem 2 (Verma and Pearl, 1991)**.** *Two ADGs are equivalent if and only if they have the same skeletons (both graphs have the same set of edges, disregarding*

[2]Originally proposed by Verma and Pearl (1991), although differently phrased.

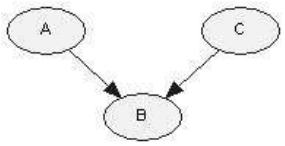


FIGURE 2: A v-structure

exact edge orientation) and the same v-structures.

An equivalence class can be represented by a graph structure. These are called partial directed acyclic graphs (PDAGs) or *patterns* (Verma and Pearl, 1991). The edges that belong to a v-structure are directed edges, and other edges, where the orientation cannot typically be determined, are represented by undirected edges.

3 Properties of Distance Measures

Traditionally, a metric is defined as a measure of distance d on a set $\mathbf{X}$.

$$d : \mathbf{X} \times \mathbf{X} \to \mathbb{R}$$

To be considered a metric, a distance measure must have the following four properties for all x, y, and z in X:

$$d(x, y) \geq 0 \tag{3}$$
$$d(x, y) = 0 \Leftrightarrow x = y \tag{4}$$
$$d(x, y) = d(y, x) \tag{5}$$
$$d(x, z) \leq d(x, y) + d(y, z) \tag{6}$$

We prove a property that is specifically suitable for distance measurements for equivalence classes of Bayesian network structures:

Proposition 1. *Let d be a proper distance metric that has properties (3) through (6). If x and y are Bayesian networks that belong to the same equivalence class, then their distance $d(x, y)$ must be 0. Furthermore, when the BN structure of z is compared with the structures of x or y then the distances $d(x, z)$ and $d(y, z)$ should be*

$$d(x, z) = d(y, z) .$$

Proof. From (6), the distances between BNs x and z and y and z are:

$$d(x, z) \leq d(x, y) + d(y, z) ,$$
$$d(y, z) \leq d(y, x) + d(x, z) .$$

Now, if we assume that x and y belong to the same equivalence class, their distance must be 0, i.e.,

$$d(x, y) = d(y, x) = 0 ,$$

we have

$$\begin{cases} d(x,z) \leq 0 + d(y,z) \\ d(y,z) \leq 0 + d(x,z) \end{cases},$$

which is satisfied if and only if

$$d(x,z) = d(y,z).$$

□

Comparing a learned structure against a gold standard network may, in general, call for asymmetry, i.e., the learned structure is compared to the gold standard but not vice versa. Using measures that violate property (5) can be acceptable in this case. But, in general, when comparing two arbitrary structures, with the same set of variables, this asymmetry is not present and there is no specific order of comparison available. In general, using proper metrics is advisable, since they are more universally applicable.

4 Related Work

We have observed that typically there are two approaches of comparing a learned Bayesian network structure to a gold standard: (1) comparing two ADG structures, and (2) comparing equivalence classes, represented by *patterns*. The former is more widespread than the latter, even though there are sound theoretical reasons to prefer *pattern* comparisons over ADG comparisons. We review the two approaches in Sections 4.1 and 4.2 respectively.

As an example, consider the Asia network (Lauritzen and Spiegelhalter, 1988). Fig. 3(a) shows the network, and Fig. 3(b) shows the *pattern* that represents the equivalence class that the network belongs to. The top three edges are undirected in the *pattern*. The existence of undirected edges in the *pattern* means that the equivalence class has more than one member, exactly six in this case. Although there are eight possible combinations, two of them would cause a new v-structure to be created, which would bring the resulting network outside of the equivalence class. Fig. 3(c) shows another member of the equivalence class Asia belongs to. If the two ADG structures in Fig. 3 would be compared with each other, focussing on differences in edge direction, a distance of 2 would be found. This is because of the two edges in Fig. 3(c) that are reversed. If first the *pattern* structures are derived from both the ADGs, and then the *patterns* are compared, a distance of 0 would be found. Since both ADG structures belong to the same equivalence class they will have the same *pattern* structure.

This simple example shows that discrepancies are possible between ADG and *pattern* scores. If Fig. 3(c) would be the result of a structure learning algorithm, and another algorithm would produce an ADG that when compared with the original would have a distance of 1, can we say that this second algorithm performs better? Because of statistical indistinguishability (Section 2), the difference may be incidental, and dependent on the exact data set used for learning the structure. When the *patterns* would have been compared, both would have had distance 0,

and, when considering only the edge direction measure, the algorithms would have to be considered equivalent.

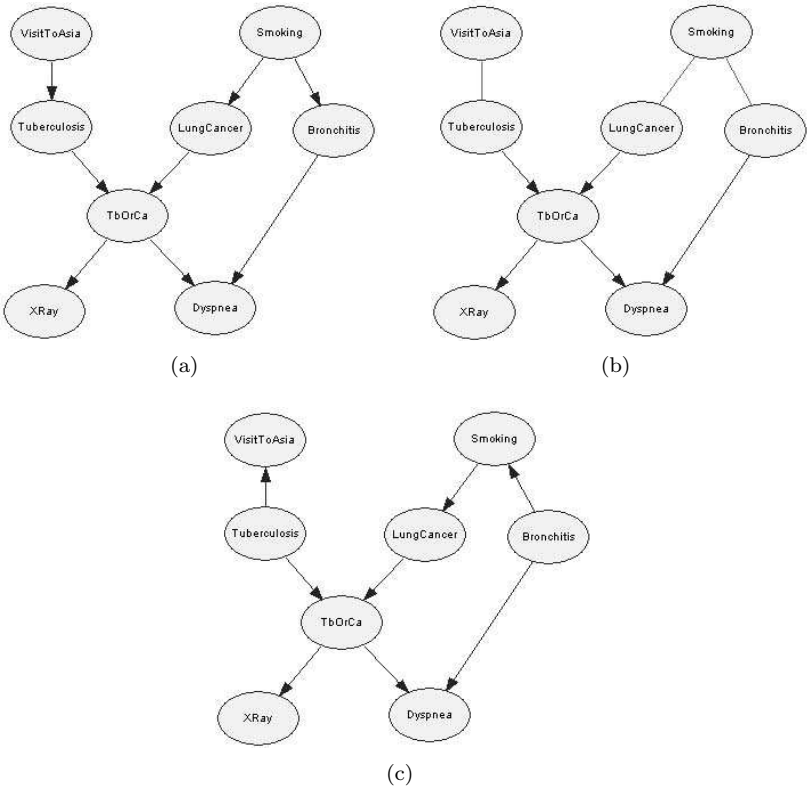


FIGURE 3: The Asia network, its corresponding *pattern*, and another instance of the equivalence class the Asia network belongs to.

We review several measures reported in the literature below.

4.1 ADG Measures

There are many examples of measures of structural distance that rely on comparing ADGs. Cooper and Herskovits (1992), for instance, compare the performance of their K2 algorithm for different sample sizes and look at added edges and missing edges, edges in the discovered graph there either were not present in the original or should have been present in the discovered graph but were missing. A similar approach was taken by Monti and Cooper (1997). Heckerman *et al.* (1995) mention a structural difference metric (symmetric difference of parents of a node in gold standard and learned structure). Colace *et al.* (2004) describe multiple metrics and introduce two normalized versions of ADG metrics. Recently, Cruz-Ramírez *et al.* (2006) evaluated the performance of the Bayesian Information Criterion (BIC) and the Minimum Description Length (MDL) principle as model selection metrics. In

one part of their evaluation, they compared learned network structures against gold standard networks.

4.2 Pattern Measures

In an example involving their PC algorithm, Spirtes *et al.* (1993) mention metrics similar to extra and missing edges: they call them edge commission and omission, but they base their metrics on *patterns*, not on ADGs. Recently Abellán *et al.* (2006) and Cano *et al.* (2008) have used similar metrics for ADGs in their evaluation of variations on the PC algorithm, They calculate their metrics before the edge orientation phase of the PC algorithm. To evaluate their hybrid algorithm, Tsamardinos *et al.* (2006) compared the output of their algorithm against a gold standard network. They specifically compare equivalence class structures and not ADGs. They defined a metric called the *Structural Hamming Distance* (SHD), which is similar to the metric proposed by Acid and de Campos (2003). They compute the SHD between two directed graphs after first converting them into *patterns* by using the approach of Chickering (1995). Perrier *et al.* (2008) proposed a small modification to the SHD metric, assigning a smaller penalty to incorrect orientation of edges.

5 Empirical Evaluation

While there are some theoretical grounds, as discussed in Section 2, for preferring distance measures between patterns, a researcher testing structure learning algorithms is left with the choice of measure. To our knowledge, no systematic comparison of different measures has ever been performed. In this section, we report the results of a series of experiments that focus on empirical comparison of various measures of structural distance.

We created a framework that allowed us to generate data from a gold standard network, learn structures from these data, and compare the structures using multiple measures of distance. While the exact algorithm that we used in our experiments is not important, we used a Bayesian search algorithm to learn the structures, which were always ADGs.[3] This is important for our evaluation, since we want to compare measures used for both ADGs and *pattern* structures. An ADG can be converted into the *pattern* representing its equivalence class, but not vice versa.

5.1 Methodology

We chose four different existing Bayesian networks of varying sizes to use as gold standard networks: Asia (Lauritzen and Spiegelhalter, 1988), Alarm (Beinlich *et al.*, 1989), Hailfinder (Abramson *et al.*, 1996), and HEPAR (Onisko, 2003).

[3]The PC algorithm, based on constraint based search, returns *patterns*. Bayesian search algorithms typically only return ADGs, although Chickering (2002) extended a Bayesian search algorithm to learn equivalence classes represented by *patterns*, rather than ADGs.

For every network, we generated multiple data sets of three different sample sizes: 1,000, 5,000, and 10,000. We generated 100 data sets of each sample size to be able to acquire useful statistics for the different measures under study.

For every combination of network and sample size, we ran a Bayesian search algorithm called Greedy Thick Thinning (Dash and Druzdzel, 2003) and compared the result of the algorithm with the gold standard twice: (1) we compared the ADG structure that the algorithm returned against the ADG structure of the gold standard, and (2) we compared the equivalence class structures, which are obtained by applying the algorithm described by Chickering (1995) on the ADG structures that we found earlier. We recorded the values calculated by the various metrics. We examined the following metrics and measures:

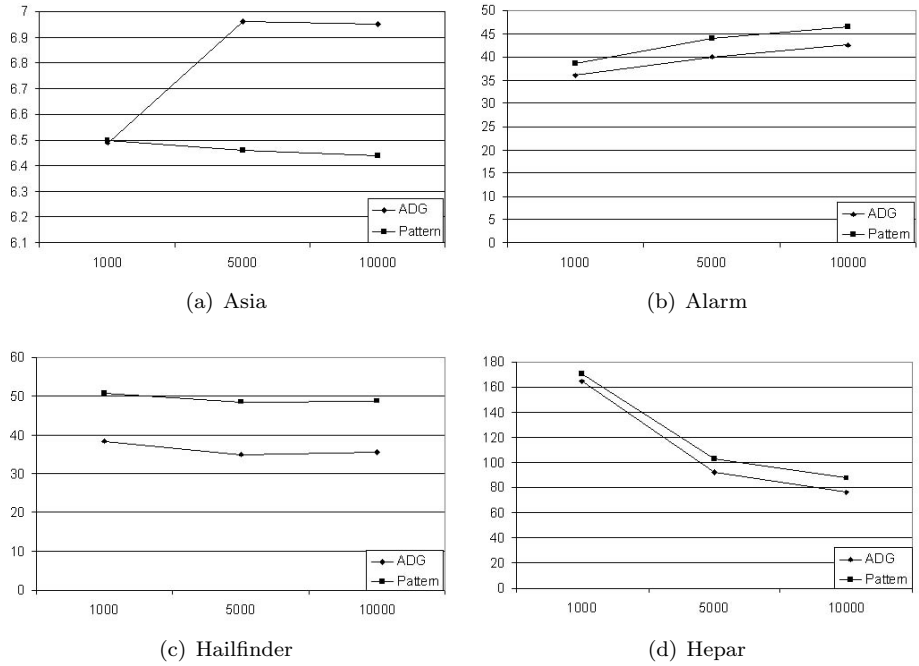
1. **Missing_Edges**: counts edges that are present in the original structure but are missing in the learned structure. (lower is better)
2. **Extra_Edges**: counts edges that are found in the learned structure but are not present in the original structure. (lower is better)
3. **Correct_Edges**: counts edges, regardless of their orientation, that are present both in the original structure and the learned structure. (higher is better)
4. **Correct_Type_Edges**: counts edges, taking into account their orientation, that are present both in the original structure and the learned structure. (higher is better)
5. **Correct_Edge_Direction**: counts directed edges in the learned structure that are oriented correctly. (higher is better)
6. **Incorrect_Edge_Direction**: counts directed edges in the learned structure that are oriented incorrectly. (lower is better)
7. **Topology**: measure was proposed by Colace *et al.* (2004) and can be summarized as a normalized, weighted combination of measures 1,2, and 3. (higher is better)
8. **Global**: like the previous measure, was proposed by Colace *et al.* (2004), is similar to the topology metric, combining measures 1,2,4, and 5.(higher is better)
9. **Hamming_Distance**: Describes the number of changes that have to be made to a network for it to turn into the one that it is being compared with. It is the sum of measures 1, 2, and 6. Versions of the Hamming distance metric have been proposed by Acid and de Campos (2003), Tsamardinos *et al.* (2006), and Perrier *et al.* (2008). (lower is better)

We stored the results per experiment (network, size) and calculated the mean, variance, standard deviation, minimum value, and maximum value. We plotted several series of averages to look for trends. Examples of examined series are: comparing metric values for ADG or *pattern* while varying data set size, comparing the ADG and *pattern* metric for one network while varying data set size, and comparing the ADG and *pattern* metric for one data set size while varying the network.

5.2 Results

We do not show results of all the metrics that we examined but will illustrate our findings, which apply in general to all metrics, using some of the more common metrics used for comparison. Included are the Hamming distance and its components: missing and added edges, and correct and incorrect edge orientation.

We found that generally, the distances reported by the metrics on either ADG structures or *patterns* are similar. This can be seen in Fig. 4, which shows for each of the networks the Hamming distance metric as a function of the size of the data set. The exception here is Fig. 4(a), but Asia is a small network with only 8 nodes and the differences in the values of the Hamming distance metric are most likely due to variance.



(a) Asia (b) Alarm

(c) Hailfinder (d) Hepar

FIGURE 4: Similarity of values of the Hamming distance metric

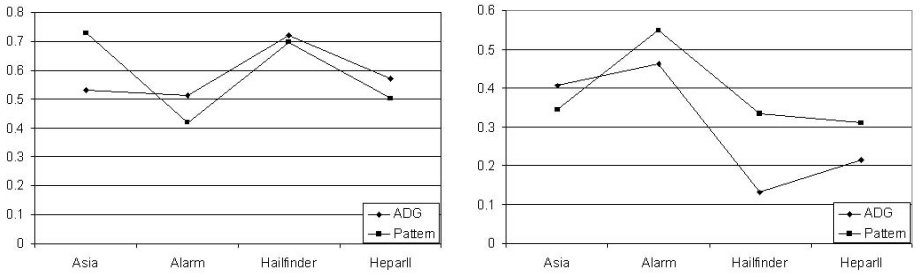
Second, we have found that there are situations where it does matter whether the distance is measured between ADG or *pattern* structures. We show examples of this in Fig. 5, which contains plots of normalized values of two measures, correct edge directions (Fig. 5(a)) and incorrect edge directions (Fig. 5(b)). The values of the two measures are normalized by the total number of edges in the original network (incorrect edge direction), and the total number of directed edges in the original network (correct edge direction).

Depending on the network that we used to generate the training data, the structures that the learning algorithm retrieved had quite different average values for the counts of correct and incorrect edge directions for the ADG and the *pattern*

cases. This is apparent especially in the Hailfinder network. On the average, both show that the learning algorithm retrieved about 70% of the directed edges. But there is a large discrepancy between the counts of the incorrect edge directions.

There is an explanation for these differences. The correct edge direction measure, examines only directed edges and the incorrect edge direction measure examines directed and undirected edges. The correct interpretation of the graphs is that in both structures 70% of all the directed edges are retrieved. But for the ADG, this means 70% of all edges and for the *pattern* this means 70% of the directed edges. These edges are either parts of, or originate from v-structures. Since the same structure is compared under two circumstances, we can say that the directed edges included in the *pattern* are a subset of the directed edges that are present in the ADG. Since an ADG will usually, and certainly for the Hailfinder network, have more directed edges than the corresponding *pattern* representation, some of these will be statistically indistinguishable. Some correct edges in the ADG may have been correctly identified by coincidence. This is most visible in the incorrect edge direction, where the lower *pattern* score could be explained by the fact that *patterns* have three different possible edges and ADGs only have two. If two randomly generated graphs with the exact same skeleton are compared, ADGs would have a 50% chance for each individual edge to be correct. For *patterns*, this would only be 33%, since there are three possibilities: directed edge from a to b, directed edge from b to a, or an undirected edge. More errors are simply to be expected.

This is counterintuitive from a theoretical point of view. An equivalence class can contain many ADGs, and, because in *patterns* edges can be undirected, there should be fewer errors due to statistical indistinguishability. But even when all correct edges in a *pattern* are counted, directed and undirected, there is still a huge gap between ADGs and *patterns*. We found for the Hailfinder network that on the average about 52% of all edges in the original network are retrieved correctly versus 72% in the ADGs. The data suggests that the learned ADG structure appears closer to the ADG of the original network than the respective equivalence classes that they belong to.



(a) Correct Edge Direction (b) Incorrect Edge Direction

FIGURE 5: Normalized values of measures for networks learned with 10,000 samples

Third, we observed that sometimes, when more data were available for learning, the distance to the original network would increase. This is clearly visible in Fig. 4(b), where the Hamming distance increased when more data was available

for learning. Further investigation showed that factors contributing to the increase of the Hamming distance were: extra edges and incorrect edge directions.

Finally, we want to comment on the Hamming distance metric. It is a proper metric, and it is a useful indicator of structural distance. But it is unwise to rely solely on this metric, or any single metric, in general. As an example, consider Fig. 6. Here, the Hamming metric shows a clear trend. As more data are available, the learned network structure seems to better resemble the original network. However, when we examine the individual components of the Hamming distance metric, we see that when more data becomes available, the skeleton of the structure improves, but the number of incorrectly oriented edges increases. Since the increase of incorrect edges is smaller than the combined decrease of missing and extra edges, this information is lost when considering only the Hamming distance. Specifically, when evaluating algorithms for causal discovery, such information might be vital and relying only on the Hamming distance as a measure of distance may be misleading.

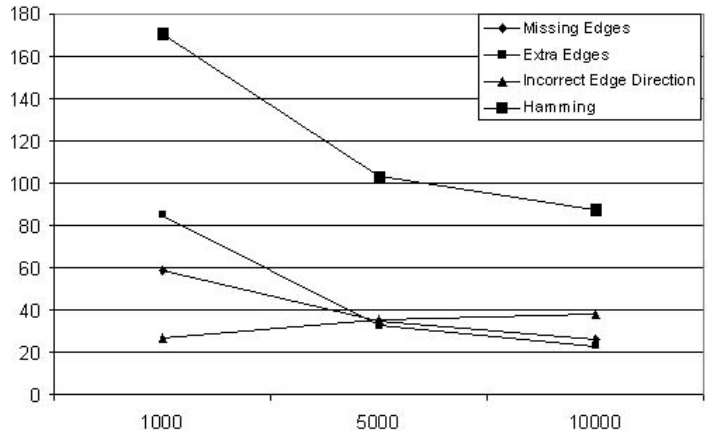


FIGURE 6: Hamming metric and its components on the Hepar network, 10,000 samples

6 Conclusion

Our experiments led to the following observations. (1) In general, the measurements performed either on ADGs or *patterns* are similar. They differ not too much, although *pattern* distances seem to be slightly higher in general. (2) There are situations, where distance measures applied to ADGs or *patterns* will differ more significantly. This may be due to a specific interpretation of some of the measures, but a possible explanation is that, in *patterns* more errors are made due to the fact that more different edges are possible. From a theoretical point of view, this is counterintuitive, since a pattern represents a BN equivalence class that can contain many ADG structures, which should nullify many small errors that should negatively impact measures performed on ADGs. A possible conclusion is that

ADG measures are underestimating the true distance between the learned and the original network. (3) We observed, surprisingly, that for several cases having more data available for the learning algorithm resulted in network structures that were more distant from the original network. Since this affected both ADG and *pattern* structures, we assume that the cause is that specific structure learning algorithm we used for the experiments. We intend to test this by trying other Bayesian search algorithms. Finally, (4) we do not recommend relying on only one measure. Some measures, like the Hamming distance, aggregate other measures, leading to loss of information.

On theoretical grounds, pattern measures should be preferred over ADG measures, especially when evaluating algorithms for causal discovery. Taking statistical indistinguishability into account, *pattern* comparison is a more conservative measure of distance, which should be preferred when comparing causal structures. In classification tasks, measures of structural distance become less important. Both, comparing ADGs or *patterns* will suffice, but for this application it is more common to compare performance with measures used in machine learning.

Acknowledgments

This work was supported by the Air Force Office of Scientific Research grant FA9550-06-1-0243 and by Intel Research. We thank Mark Voortman and Saeed Amizadeh for their useful suggestions.

Empirical part of the paper was performed using SMILE, an inference engine, and GeNIe, a development environment for reasoning in graphical probabilistic models, both developed at the Decision Systems Laboratory, University of Pittsburgh, and available at `http://genie.sis.pitt.edu/`.

References

J. Abellán, M. Gómez-Olmedo, and S. Moral (2006), Some Variations on the PC Algorithm, in *Proceedings of the Third European Workshop on Probabilistic Graphical Models*, pp. 1–8.

J. Abramson, B., W. Brown, A. Edwards, and R. Winkler Murphy (1996), Hailfinder: A Bayesian system for forecasting severe weather, *International Journal of Forecasting*, 12(1):57–72.

Silvia Acid and Luis M. de Campos (2003), Searching for Bayesian network structures in the space of restricted acyclic partially directed graphs, *Journal of Artificial Intelligence Research*, 18:445–490.

Ingo Beinlich, Jaap Suermondt, Martin Chavez, and Gregory Cooper (1989), The ALARM Monitoring System: A Case Study with Two Probablistic Inference Techniques for Belief Networks, in *Second European Conference on Artificial Intelligence in Medicine*, pp. 247–256, London.

A. Cano, M. Gómez-Olmedo, and S. Moral (2008), A Score Based Ranking of the Edges for the PC Algorithm, in *Proceedings of the Fourth European Workshop on Probabilistic Graphical Models*, pp. 41–48.

David Maxwell Chickering (1995), A Transformational Characterization of Equivalent Bayesian Network Structures, in *Proceedings of the 11th Annual Conference on Uncertainty in Artificial Intelligence (UAI-95)*, pp. 87–98, Morgan Kaufmann, San Francisco, CA.

David Maxwell Chickering (2002), Learning equivalence classes of Bayesian-network structures, *Journal of Machine Learning Research*, 2:445–498.

Francesco Colace, Massimo De Santo, Mario Vento, and Pasquale Foggia (2004), Bayesian Network Structural Learning from Data: An Algorithms Comparison, in *ICEIS (2)*, pp. 527–530.

Gregory F. Cooper and Edward Herskovits (1992), A Bayesian Method for the Induction of Probabilistic Networks from Data, *Machine Learning*, 9(4):309–347.

Nicandro Cruz-Ramírez, Héctor-Gabriel Acosta-Mesa, Rocío-Erandi Barrientos-Martínez, and Luis-Alonso Nava-Fernández (2006), How Good Are the Bayesian Information Criterion and the Minimum Description Length Principle for Model Selection? A Bayesian Network Analysis, in *MICAI*, pp. 494–504.

Denver Dash and Marek J. Druzdzel (2003), Robust Independence Testing for Constraint-Based Learning of Causal Structure, in *Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence (UAI-03)*, pp. 167–174, Morgan Kaufmann.

Nir Friedman (1998), The Bayesian Structural EM Algorithm, in *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pp. 129–138, Morgan Kaufmann.

David Heckerman, Dan Geiger, and David M. Chickering (1995), Learning Bayesian Networks: The Combination of Knowledge and Statistical Data, volume 20, pp. 197–243, Kluwer Academic Publishers, Hingham, MA, USA.

Steffen. L. Lauritzen and David. J. Spiegelhalter (1988), Local computations with probabilities on graphical structures and their application to expert systems, *Journal of the Royal Statistical Society*, 50:157–224.

Michael G. Madden (2003), The performance of Bayesian network classifiers constructed using different techniques, in *Working notes of the ECML/PKDD-03 workshop on*, pp. 59–70.

Stefano Monti and Gregory F. Cooper (1997), Learning Bayesian belief networks with neural network estimators, in *Neural Information Processing Systems 9*, pp. 579–584, MIT Press.

Agnieszka Onisko (2003), *Probabilistic Causal Models in Medicine: Application to Diagnosis of Liver Disorders.*, Ph.D. thesis, Institute of Biocybernetics and Biomedical Engineering, Polish Academy of Science, Warsaw.

Judea Pearl (1988), *Probabalistic Reasoning in Intelligent Systems*, Morgan Kaufman, San Mateo.

Eric Perrier, Seiya Imoto, and Satoru Miyano (2008), Finding Optimal Bayesian Network Given a Super-Structure, *Journal of Machine Learning Research*, 9:2251–2286.

M. Singh (1997), Learning Bayesian Network from Incomplete Data, *AAAI*, '97:27–31.

Peter Spirtes, Clark Glymour, and Richard Scheines (1993), *Causation, Prediction, and Search*, MIT Press, 1st edition.

Ioannis Tsamardinos, Laura E. Brown, and Constantin F. Aliferis (2006), The max-min hill-climbing Bayesian network structure learning algorithm, *Mach. Learn.*, 65(1):31–78.

Thomas Verma and Judea Pearl (1991), Equivalence and synthesis of causal models, in *UAI '90: Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*, pp. 255–270, Elsevier Science Inc., New York, NY, USA.