Widened Learning of Bayesian Network Classifiers

Oliver R. Sampson and Michael R. Berthold

Chair for Bioinformatics and Information Mining
Department of Computer and Information Science
University of Konstanz, Germany

Abstract. We demonstrate the application of *Widening* to learning performant Bayesian Networks for use as classifiers. Widening is a framework for utilizing parallel resources and diversity to find models in a hypothesis space that are potentially better than those of a standard greedy algorithm. This work demonstrates that widened learning of Bayesian Networks, using the Frobenius Norm of the networks' graph Laplacian matrices as a distance measure, can create Bayesian networks that are better classifiers than those generated by popular Bayesian Network algorithms.

1 Introduction

Widening [2,18] formalizes a method for executing a greedy learning algorithm in parallel while using diversity to guide the parallel refinement paths through a hypothesis[1] space. This enables the system as a whole to avoid local optima and potentially find better models than the greedy learning algorithm would otherwise find. Previous work [29,13] has demonstrated its viability on real world algorithms. This work builds on that with an application to the superexponentially-sized [28] hypothesis space of learning Bayesian Networks. Bayesian Networks [26] are probabilistic graphical networks, which describe relationships of conditional dependence between the features of a dataset. Perhaps the best known of these graphical networks is the network defined by the Naïve Bayes algorithm [11,23]. This paper describes the application of Widening to the learning of Bayesian Networks for use as classifiers.

The ultimate goal of Widening is not just to provide better solutions using parallel resources, but to provide better solutions in the same time or less than the canonical greedy algorithm. To enable this, *communication-free* Widening would allow the model refinement paths, separated by some measure of diversity, to be followed through the solution space until some stopping criterion is met. The difficulty in that effort has been finding a suitable measure of distance, i.e., diversity. Here, we show that the Frobenius Norm of Bayesian Networks' graph

[1] We freely mix the use of "solution space" and "hypothesis space" throughout this paper, referring essentially to the same space, but drawing attention to whether it is the evaluation of the hypothesis or the hypothesis itself that is important.

Laplacians is a useful measure of diversity for comparing Bayesian networks in the Widening framework, albeit not in a communication-free framework.

2 Background

2.1 Learning and Scoring Bayesian Networks

A Bayesian network, B, derived from a dataset, $\mathcal{D}$, is a triple, $\langle \mathcal{X}, G, \Theta \rangle$, where $\mathcal{X}$ is the set of features or random variables in the dataset, G is a directed-acyclic-graph (DAG), and Θ is the set of conditional probability tables (CPT) for the features in $\mathcal{X}$. The graph $G = (\mathcal{X}, \mathcal{E})$, is an ordered pair, where each node, $X \in \mathcal{X}$, is a feature from the dataset and where each edge, $E = \{X_i, X_j\} \in \mathcal{E}$, is directed according to the dependency of one feature on another.

There are four general categories of algorithms for learning Bayesian networks: *search-and-score*, *constraint-based*, *hybrid* [19] and *evolutionary algorithms* [20]. Search-and-score methods such as K2 [9] and Greedy Equivalence Search (GES) [8] rely on heuristics to sequentially add, remove, or change the direction of the edges in the graph, G, to which a scoring method is applied. Edges that improve the score are kept in the graph for the next iteration of add, delete, or change. Constraint-based methods such as PC Algorithm [32] or CBL [7] rely on some assumptions about the dependency relationships of the features, from which a partially-directed-acyclic-graph is generated. This "skeleton" of a graph describes the neighbors of each of the feature nodes within the graph, but not necessarily the direction of the edges between the nodes. After determining the skeleton, search-and-score methods are used to find better networks, i.e., networks with a higher score, by flipping the direction of the edges and re-evaluating the score. Hybrid methods such as Max-Min Hill-Climbing [35] incorporate techniques from both the search-and-score and the constraint-based methods. Algorithms based on evolutionary techniques randomly change and combine networks and evaluate them with a fitness function.

Several scoring functions have been proposed for the use of learning Bayesian networks. For an extensive overview and comparison, the reader is referred to [6]. Scoring functions can be grouped into two categories, *Bayesian* and *information-theoretic*. Bayesian scoring methods calculate the posterior probability distribution based on the prior probability distribution conditioned on $\mathcal{D}$. Some examples of Bayesian scoring functions are K2 [9], Bayesian Dirichlet (BD) [17], BD with equivalence assumption (BDe) [9], and BD with equivalence and uniform assumptions (BDeu) [5].

Information-theoretic score functions are based on Shannon entropy and the amount of compression possible for a Bayesian network. The Log-Likelihood (LL) score is based on the logarithm of the likelihood of $\mathcal{D}$ given B, i.e., $\log(P(\mathcal{D}|B))$. The LL score is better, in general, for complete networks, and for this reason alternate scoring functions have been proposed that penalize the LL according to some factor. The Minimum Description Length [34] (MDL), the Akaike Information Criterion [1] (AIC), and Bayesian Information Criterion [30] (BIC)

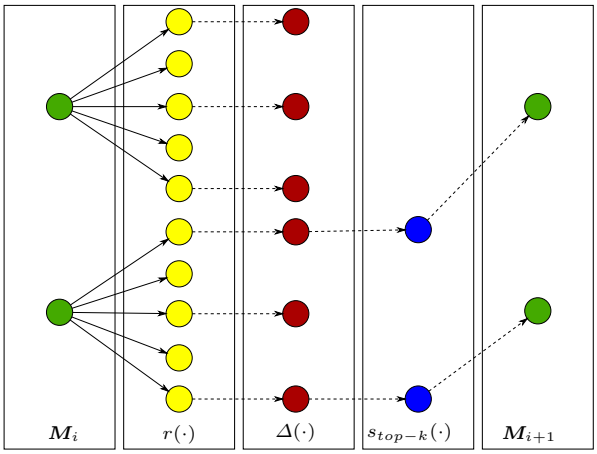


Fig. 1: Widening. Each $m_{k,i} \in \boldsymbol{M}_i$ (green) is refined to five models (yellow). In each of these sets, the three most diverse from another are determined (red). The two best performing models (blue) are selected and used for the next iteration, $m_{k,i+1} \in \boldsymbol{M}_{i+1}$ (green).

(roughly equivalent to MDL) all adjust the LL by different proportions of the network complexity.

Less commonly, and more often associated with evolutionary algorithms such as [27] and [31], the performance of the networks as a classifier is used as the scoring, i.e., fitness, function. In this vein, the work described in this paper also uses accuracy as the scoring function.

When used as a classifier, the relevant portion of the network contains the parents of the target node, the children of the target node, and the children's other parents. This is termed the *Markov blanket* [26, p. 97].

2.2 Widening

The Widening framework [2,18] (See Figure 1.) describes a general process for improving greedy learning algorithms where models, $m \in \mathcal{M}$, are iteratively refined and scored in parallel. Each refinement path follows a different route through the hypothesis space. The models at each refinement step are separated using a diversity measure, Δ, which enforces a distance between the models' respective refinement paths.

More formally, a refinement operator, $r(\cdot)$, applied to a model, m, generates a set of models, $\boldsymbol{M'}$, from the set of all possible models, $\mathcal{M}$, in the hypothesis space. A selection operator, $s(\cdot)$, when applied to a set of models, selects a subset according to a performance metric. In Widening's most rudimentary form, the best k performing models from a refinement step are selected, $s_{\text{top}-k}(\boldsymbol{M'})$, which in turn are further refined and selected until a stopping criterion is met. The $s_{\text{top}-k}(\cdot)$ operator has a similarity to a Beam Search, [22] but instead of a

selection operator based solely on performance, the selection is also based on diversity, due to the refinement operator.

2.3 Related Work

Learning Bayesian Networks for classification, either by modifying networks created by Naïve Bayes or by the generation of networks through completely different methods, is a very active research area. An excellent survey can be found in [4]. In [14], Friedman et al. describe Tree Augmented Naïve Bayes Network (TAN) where edges are added between child nodes of a Naïve Bayes network in a greedy search using the MDL scoring function, and whose structure is limited to that of a tree. The authors also describe learning an "unrestricted" Bayesian Network Augmented Naïve Bayes (BAN), but these networks do not include networks with nodes as parents for the target nodes, but rather just more complex relationships among the child nodes. Cheng et al. in [7] describe an algorithm (CBL) for finding General Bayesian Networks (GBN) based on conditional independence tests using Mutual Information (MI).

In [25] Nielsen et al. present k-Greedy Equivalence Search (KES) which is a modification to the GES, where a random subset of models from the entire set is chosen and evaluated. They describe this as a method specifically to avoid the local optima encountered by GES in [8].

Su and Zhang describe in [33] what they call Full Bayesian Networks (FBN), which are TANs where all child nodes of the target are connected to a maximal subset of the other child nodes based on an ordering using MI. This structure is in turn used to learn a Decision Tree-like structure for learning CPT-Trees.

The work presented here is similar to the TAN in [14], in that we perform a greedy search for better networks starting with a network generated by Naïve Bayes. It is similar to the work in [25], in that a subset of models is chosen and evaluated specifically to avoid local optima. It differs from these two, in that 1) any configuration of Bayes Network is allowed, 2) diversity between networks rather than randomness is used to select models, and 3) classification accuracy is employed for the scoring function.

3 Widened Bayesian Networks

3.1 Application of the Widening Framework

The simplest search-and-score method (Hill-Climbing or Greedy Search) refines a Bayesian Network model by changing a randomly or heuristically chosen edge, E, and scores the network according to one of the scoring functions discussed in Section 2.1. The algorithm greedily keeps the changed edge if it improves the score. Using the Widening notation, the greedy search-and-score method is $B_{i+1} = s_{\text{top}-k=1}(r(B_i))$, where i refers to the current search-and-score iteration. The process stops when no further improvement is seen.

The application of Widening to this process is to refine a set of different Bayesian networks at each stage, $\boldsymbol{B}_{i+1} = r(\boldsymbol{B}_i)$. Each model is refined to a

number, l, of refinements. From this set, k models are selected by the selection operator, $s_{\text{top}-k}(\cdot)$. $k \times l$ models are generated during each refinement iteration, with the exception of the initial one. Additionally, the application of a diversity measure, Δ, is used by the refinement operator, and therefore notated as $r_{\Delta}(\cdot)$. The refinement operator ensures that the models are different enough to explore disparate regions of the hypothesis space.

Scoring Bayesian Networks by using classification accuracy is common only with the evolutionary algorithms, even though, for example, Friedman et al. in [14] explicitly say that one of the reasons that their TAN ALGORITHM did not always provide superior solutions was that the structural score may not have been a good analog for the use of the network in its role as a classifier.

In summary, each step in the top-k Widening process is described as

$$\boldsymbol{B}_{i+1} = s_{\text{top}-k}(r_{\Delta}(\boldsymbol{B}_i)) \tag{1}$$

3.2 Refinement Operator

The refinement operator creates a list of all possible pairs of nodes, i.e., all possible edges. Each edge is compared with the current model and up to two additional models are created based on the edge. (See Figure 2.)

1. If it is possible to add the edge to the initial model (Figure 2a), i.e., its presence would not contravene the definition of DAG by creating a loop, it is added. (See Figure 2f.)
2. If it is present in the model, it is removed. (See Figures 2b and 2d.)
3. If it is present in the model, and the reversal of its direction would not create a loop, it is reversed. (See Figures 2c and 2e.)

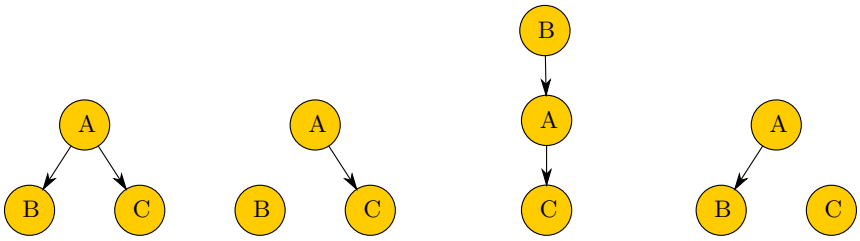
A distance matrix of all distances between network model pairs is then calculated.

3.3 Diversity

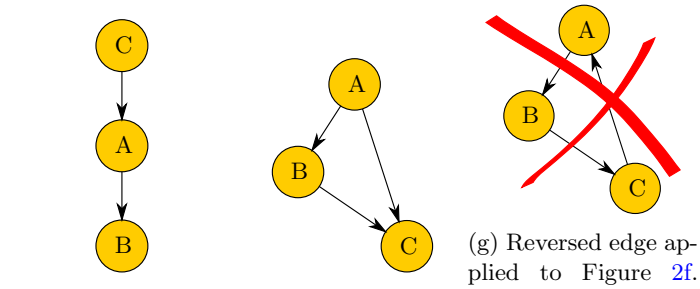
There are a variety of measures for comparing two labeled DAGs. Early experiments indicated that the Hamming distance [16] does not measure diversity in a way that scales well to larger networks. For this work, we have chosen the Frobenius Norm of the difference between the graphs' Laplacian matrices. The Frobenius Norm is sometimes referred to as the Euclidean norm and provides a "measure of distance on the space of matrices." [15] The Frobenius Norm for a matrix, $A \in \mathbb{R}^{m \times n}$ is defined as

$$||A||_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2}[15] \tag{2}$$

where, a_{ij} are elements of matrix A.



(a) Initial model. (b) $A \to B$ removed. (c) $A \to B \Rightarrow B \to A$ (d) $A \to C$ removed.



(e) $A \to C \Rightarrow C \to A$ (f) $B \to C$ added. (g) Reversed edge applied to Figure 2f. Disallowed; not DAG.

Fig. 2: Example possible refinements for the three edges $\{\langle A, B\rangle, \langle A, C\rangle, \langle B, C\rangle\}$

The Laplacian matrix of a graph is given by the formula $L = D - A$, where D is the out-degree matrix, and A is the adjacency matrix. Here we use the Frobenius Norm of the difference of each pair of Bayesian networks' Laplacian matrices, i.e.,

$$\Delta_{\text{Frobenius}} = ||L_{B_p} - L_{B_q}||_F : B_p, B_q \in \boldsymbol{B}_i \tag{3}$$

and $\boldsymbol{B}_i$ is the set of refined Bayesian networks from Equation 1.

The p-Dispersion Problem describes selecting a subset of points from a larger set, where the subset's minimum pairwise distances are maximized. There are several diversity measures used commonly with the p-Dispersion Problem, including *sum* and *min-sum*. *p-dispersion-sum* simply maximizes the sum of the distances between any two points in the subset, whereas *p-dispersion-min-sum* maximizes the sum of the minimum distances between two points. *p-dispersion-sum* has the property of pushing the resultant subset to the margins of the original set, whereas the subset derived using *p-dispersion-min-sum* is more representative of the dataset as whole [24]. Because of this property, and based on the results in [29], we favor *p-dispersion-min-sum* as the diverse subset selection method.

Definition 1 ***p-dispersion-min-sum.***[2] *Given a set* $\boldsymbol{B} = \{B_1, \cdots, B_n\}$ *of* n *distinct Bayesian networks and* p*, where* $p \in \mathbb{N}$ *and* $p \leq n$*, and a distance*

[2] In this application, it would be correctly termed "l-dispersion-min-sum," but the notation is written here as "p" to be consistent with the literature.

measure $d(B_i, B_j) : B_i, B_j \in \boldsymbol{B}$ *between Bayesian networks* B_i *and* B_j*, the* p*-diversity-min-sum problem is to select the set* $\hat{\boldsymbol{B}} \subseteq \boldsymbol{B}$*, such that*

$$\begin{aligned} \hat{\boldsymbol{B}} &= \underset{\substack{\boldsymbol{B}' \subseteq \boldsymbol{B} \\ |\boldsymbol{B}'|=p}}{\operatorname{argmax}} f(\boldsymbol{B}'), \text{ where} \\ f(\boldsymbol{B}') &= \sum_{i=1}^{p} \min_{1 \le i,j \le n, i \neq j} d(B_i, B_j) : B_i, B_j \in \boldsymbol{B} \text{[24]} \end{aligned} \tag{4}$$

The p-Dispersion Problem is known to be $\mathcal{NP}$-complete, and when adjusting the diversity criterion to be *min-sum*, the problem is $\mathcal{NP}$-hard [12].

3.4 Selection Operator

The selection operator presented in this work is simply the performance metric of the Bayesian network as a classifier, similar to that of [31]. When compared to the scoring methods described in Section 2.1, this has the advantage of being directly related to the network's use as a classifier, and networks that perform poorly as classifiers are eliminated from the refinement paths. The calculation for determining the target winner is similar to that of Naïve Bayes, except the probabilities of the parents of the target node and of the other parents of the target's child nodes are considered.

$$\hat{C} = \underset{j=1,\ldots,|C|}{\operatorname{argmax}} P(C_j, \boldsymbol{X}_m) = \underset{j=1,\ldots,|C|}{\operatorname{argmax}} P(C_j | \boldsymbol{pa}(C)) \prod_{i=1}^{m} P(x_i | \boldsymbol{pa}(x_i)) \text{[4]} \tag{5}$$

where $\boldsymbol{X}_m \subseteq \mathcal{X}$ is the subset of features contained in the Markov-blanket of the target node, C, and $\boldsymbol{pa}(\cdot)$ is the set of parents of a child of C in the Bayesian network.

4 Experimental Results

The experiments were performed in KNIME Analytics Platform [3]. The datasets from the UCI Machine Learning repository[3] [21] were discretized using the LUCS-KDDN software.[4] Unlike algorithms such as K2 or CBL, no assumptions were made concerning the ordering of the features within the dataset. Datasets with missing values or continuous values were not considered, because we are interested in testing the Widened learning process and not the robustness of the algorithm to various data types. The refinement operator placed no restrictions on the number of parents a node may have. The stopping criterion was set to stop the iterations when improvement in the best model compared to its performance in the previous iteration was less than 0.01%. The records in the datasets were shuffled between each widening trial of a different breadth and width.

[3] http://archive.ics.uci.edu/ml/

[4] http://www.csc.liv.ac.uk/~frans/KDD/Software/LUCS_KDD_DN/

Dataset	$\|\mathcal{D}\|$	$\|\mathcal{X}\|$	$\|C\|$	Widened Bayes	MMHC	Tabu	Hill-Climbing
`ecoli`	336	7	8	$\mathbf{0.747 \pm 0.032}$	0.430 ± 0.123	0.593 ± 0.119	0.647 ± 0.057
`flare`	1389	10	9	$\mathbf{0.843 \pm 0.015}$	$\mathbf{0.843 \pm 0.013}$	$\mathbf{0.843 \pm 0.013}$	$\mathbf{0.843 \pm 0.013}$
`glass`	214	9	7	$\mathbf{0.649 \pm 0.137}$	0.457 ± 0.151	0.564 ± 0.133	0.536 ± 0.111
`nursery`	12960	8	8	$\mathbf{0.935 \pm 0.047}$	0.570 ± 0.150	0.621 ± 0.214	0.632 ± 0.246
`pageBlocks`	5473	10	5	0.898 ± 0.015	$\mathbf{0.913 \pm 0.011}$	$\mathbf{0.913 \pm 0.004}$	0.910 ± 0.023
`pima`	768	8	2	0.710 ± 0.068	0.721 ± 0.136	$\mathbf{0.757 \pm 0.098}$	0.736 ± 0.143
`waveform`	5000	21	3	$\mathbf{0.790 \pm 0.025}$	0.342 ± 0.014	0.619 ± 0.020	0.620 ± 0.021
`wine`	178	13	3	$\mathbf{0.939 \pm 0.091}$	0.746 ± 0.150	0.798 ± 0.116	0.747 ± 0.184

Table 1: Accuracy ($\mu \pm 2\sigma$) comparison of all tested algorithms with 5-fold cross-validation.

The initial state could be any network configuration that satisfies the definition of a DAG, including a network without any edges. Because our effort is to prove the ability of Widening to find superior solutions to traditional greedy methods, we chose a Naïve Bayes configuration, where all of the non-target features are dependent on the target variable, as the initial state. This was a pragmatic decision in the sense that finding a network out of all possible networks that is tuned to the target node is impractical. Additionally, Naïve Bayes performs remarkably well given its simplicity for a large number of datasets and is a measuring stick for many new algorithms.

We tested eight datasets, `ecoli`, `flare`, `glass`, `nursery`, `pageBlocks`, `pima`, `waveform` and `wine` against three standard Bayesian Network learning algorithms, Max-Min Hill-Climbing (MMHC), Tabu, and Hill-Climbing, from the **R** `bnlearn`[5] package, version 3.8.1. MMHC and Hill-Climbing used parameters `test` = `mi`, `restart` = 100, and `perturb` = 100. These values were chosen experimentally as values that provide good results for all datasets.

Widened Bayesian Networks (WBN) significantly outperformed the other three reference implementations in five of the eight datasets, tied in one, and performed slightly worse in two.

The results in Figure 3 show a two responses to Widening. In general, with Widening we expect a gradual improvement of average performance with the width, i.e., the number of parallel paths in the solution space. Additionally, we expect a decrease in the variance of the results as the many paths push themselves towards better solutions. `ecoli`, `glass`, `nursery`, `pima`, `waveform`, and `wine` show this behavior nicely. `pageBlocks` and `flare` demonstrate how some solution space topologies cannot be explored with the refine-and-select process presented here, even though the results for the comparison algorithms for `flare` indicate that the resultant Bayesian network is a best fit. The non-responsive nature of `pageBlocks` however, invites further research into other refining-and-select strategies and/or diversity measures.

[5] http://www.bnlearn.com/

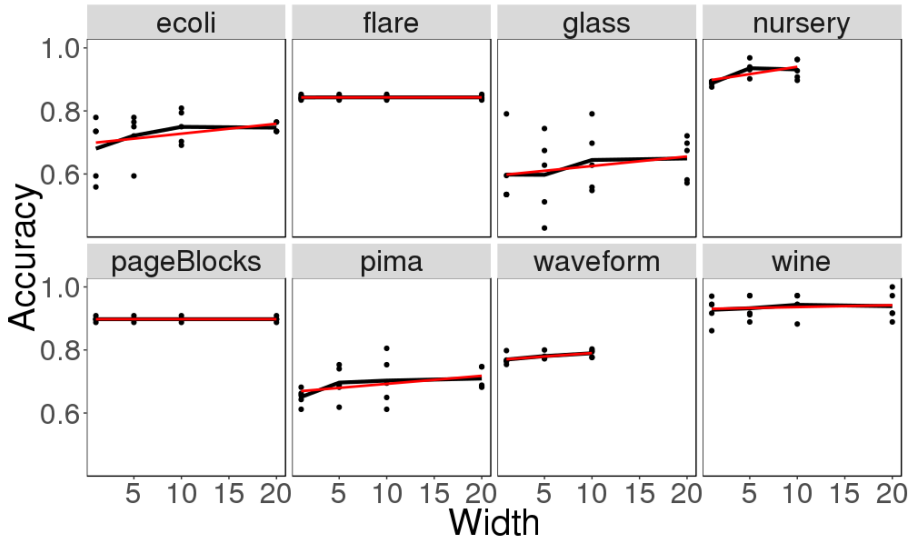


Fig. 3: Widened Bayesian Networks accuracy progression verses *width* with 5-fold cross-validation.

5 Conclusion and Future Work

This paper demonstrates the successful initial application of Widening to learning Bayesian Networks for use classifiers and demonstrates classification scoring techniques with the search-and-score greedy heuristic. The technique was able to find superior solutions when compared to standard Bayesian Network learning algorithms from the **R** `bnlearn` package. Although the results are similar or superior to established Bayesian Network learning algorithms on some datasets, the execution time does not meet the specified goal of finding better solutions in the same time or less as the greedy algorithm. The primary impediment to this goal, as it is demonstrated here, is the use of *p-dispersion-min-sum* for finding a maximally diverse subset of networks for refinement. Methods that allow for diverse subsets to be calculated without communication between the parallel workers would be better. (See [18] for details.) Additionally, the refinement operator considers the entire space of possible networks, where only the refinements to the Markov blanket are actually necessary. Significantly, the use of the Frobenius Norm of the difference of the Bayesian networks's graph Laplacians is very encouraging and suggests further research into distance measures based on graph features such as those derived from Spectral Graph Theory. Experiments with alternate starting states based on conditional information, in a manner similar to the PC Algorithm and CBL, or constraint-based algorithms like Incremental Association or HITON, or even to those claiming to find the exact network structure [10] could also be promising.

References

1. Hirotugu Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.
2. Zaenal Akbar, Violeta N. Ivanova, and Michael R. Berthold. Parallel data mining revisited. Better, not faster. In *Proceedings of the 11th International Symposium on Intelligent Data Analysis*, pages 23–34, 2012.
3. Michael R. Berthold, Nicolas Cebron, Fabian Dill, Thomas R. Gabriel, Tobias Kötter, Thorsten Meinl, Peter Ohl, Christoph Sieb, Kilian Thiel, and Bernd Wiswedel. KNIME: The Konstanz Information Miner. In Christine Preisach, Hans Burkhardt, Lars Schmidt-Thieme, and Reinhold Decker, editors, *Data Analysis, Machine Learning and Applications - Proceedings of the 31st Annual Conference of the Gesellschaft für Klassifikation e.V. (GfKL 2007)*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 319–326, Berlin, Germany, 2007. Springer.
4. Concha Bielza and Pedro Larranaga. Discrete Bayesian network classifiers: a survey. *ACM Computing Surveys (CSUR)*, 47(1):5, 2014.
5. Wray Buntine. Theory refinement on Bayesian networks. In *Proceedings of the Seventh conference on Uncertainty in Artificial Intelligence*, pages 52–60. Morgan Kaufmann Publishers Inc., 1991.
6. Alexandra M. Carvalho. Scoring functions for learning Bayesian networks. Technical Report INESC-ID Tec. Rep. 54/2009, Instituto superior Téchnico, Technical University of Lisboa, Apr 2009.
7. Jie Cheng, David A. Bell, and Weiru Liu. An algorithm for Bayesian belief network construction from data. In *Proceedings of AI & STAT '97*, pages 83–90, 1997.
8. David Maxwell Chickering. Optimal structure identification with greedy search. *The Journal of Machine Learning Research*, 3:507–554, 2002.
9. Gregory F. Cooper and Edward Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine learning*, 9(4):309–347, 1992.
10. Cassio P. De Campos, Zhi Zeng, and Qiang Ji. Structure learning of Bayesian networks using constraints. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 113–120. ACM, 2009.
11. Richard O. Duda and Peter E. Hart. *Pattern classification and scene analysis*. John Wiley & Sons, New York, 1973.
12. Erhan Erkut. The discrete p-dispersion problem. *European Journal of Operational Research*, 46(1):48–60, 1990.
13. Alexander Fillbrunn and Michael R. Berthold. Diversity-driven widening of hierarchical agglomerative clustering. In *Advances in Intelligent Data Analysis XIV*, pages 84–94. Springer, 2015.
14. Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2-3):131–163, 1997.
15. Gene H. Golub and Charles F. van Loan. *Matrix computations*. The Johns Hobpkins University Press, 4th edition, 2013.
16. Richard Wesley Hamming. Error detecting and error correcting codes. *The Bell System Technical Journal*, 29(2):147–160, April 1950.
17. David Heckerman, Dan Geiger, and David M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243, 1995.
18. Violeta N. Ivanova and Michael R. Berthold. Diversity-driven widening. In *Proceedings of the 12th International Symposium on Intelligent Data Analysis (IDA 2013)*, 2013.

19. Timo J. T. Koski and John M. Noble. A review of Bayesian networks and structure learning. *Mathematica Applicanda*, 40(1):53–103, 2012.
20. Pedro Larrañaga, Hossein Karshenas, Concha Bielza, and Roberto Santana. A review on evolutionary algorithms in Bayesian network learning and inference tasks. *Information Sciences*, 233:109–125, 2013.
21. Moshe Lichman. UCI Machine Learning Repository, 2013.
22. Bruce T. Lowerre. *The HARPY speech recognition system.* PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 1976.
23. Melvin Earl Maron and John L. Kuhns. On relevance, probabilistic indexing and information retrieval. *Journal of the ACM (JACM)*, 7(3):216–244, 1960.
24. Thorsten Meinl. *Maximum-Score Diversity Selection.* PhD thesis, University of Konstanz, July 2010.
25. Jens D. Nielsen, Tomáš Kočka, and José M. Peña. On local optima in learning Bayesian networks. In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, pages 435–442. Morgan Kaufmann Publishers Inc., 2003.
26. Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers Inc., 1988.
27. Franz Pernkopf. Bayesian network classifiers versus k-NN classifier using sequential feature selection. In *AAAI*, pages 360–365, 2004.
28. Robert W Robinson. Counting unlabeled acyclic digraphs. In *Combinatorial mathematics V*, pages 28–43. Springer, 1977.
29. Oliver Sampson and Michael R. Berthold. Widened KRIMP: Better performance through diverse parallelism. In Hendrik Blockeel, Matthijs van Leeuwen, and Veronica Vinciotti, editors, *Advances in Intelligent Data Analysis XIII*, volume 8819 of *Lecture Notes in Computer Science*, pages 276–285. Springer International Publishing, October 2014.
30. Gideon Schwarz. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.
31. Basilio Sierra and Pedro Larrañaga. Predicting the survival in malignant skin melanoma using Bayesian networks. An empirical comparison between different approaches. *Artificial Intelligence in Medicine*, 14(1-2):215–230, 1998.
32. Peter Sprites, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search.* MIT Press, 1993.
33. Jiang Su and Harry Zhang. Full Bayesian network classifiers. In *Proceedings of the 23rd international conference on Machine learning*, pages 897–904. ACM, 2006.
34. Joe Suzuki. A construction of Bayesian networks from databases based on an MDL principle. In *Proceedings of the Ninth international conference on Uncertainty in artificial intelligence*, pages 266–273. Morgan Kaufmann Publishers Inc., 1993.
35. Ioannis Tsamardinos, Laura E. Brown, and Constantin F. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine learning*, 65(1):31–78, 2006.