Abstract

On the Efficiency of Conditional Independence Tests and Their Accurate Use in Markov Boundary Discovery Algorithms

PhD Thesis

Camil Băncioiu

Keywords: causal inference, feature selection, information theory, Bayesian networks, conditional independence, statistical tests, G-test, computation reuse, d-separation criterion

This thesis presents novel contributions to Markov Boundary Discovery (MBD) algorithms, specifically concerning their usage of conditional independence (CI) tests. MBD algorithms form a versatile class of algorithms which reveal the causal information present in data sets. This makes MBD algorithms applicable in both causal inference problems and in feature selection problems.

Due to the great importance of causal information in virtually all fields of science, extracting it efficiently and accurately from data sets using MBD algorithms has been researched intensively in the past and it is a topic of active research. However, the tests of conditional independence required by these algorithms have not been studied as intensively, in spite of being central to the algorithms. As a consequence, many MBD algorithms have borrowed CI test design from one another without much variation. This is a serious issue because the CI tests are a severe computational bottleneck for the algorithms, consuming around 97% of their total running time and the accuracy of the CI tests is decisive for their functioning.

The contributions presented in this thesis aim to alleviate this shortcoming. They target two essential aspects of conditional independence tests, namely their computational efficiency and the methodological evaluation of their accuracy. The theory on which the contributions rely is fundamental, simple and well-established, yet in spite of their simplicity, the improvements they bring are extremely effective, as shown in detail in the thesis and in the published articles.

The first contribution: A study of Koller and Sahami's algorithm

Chapter 3 of the thesis contains a case-study of the obsolete but influential algorithm of Koller and Sahami (KS) [23]. It is widely regarded to be the first algorithm to employ the information-theoretic concept of Markov boundaries, thus establishing a new class of algorithms. It must be emphasized that the KS algorithm was originally published as a feature selection algorithm, but it can also be applied in causal inference problems if desired.

This case-study will focus not only on the KS algorithm itself, but also on the **original comparative experiment** performed on the KS algorithm and another fundamental

algorithm, Information Gain Thresholding (IGt). Also, a collection of **novel optimizations** will be discussed, optimizations which emerged from studying and implementing KS. Designing and developing these optimizations was an essential stepping stone for the development of the later contributions discussed in the following chapters of the thesis. The aforementioned comparative experiment has been published in [9], while the optimizations for the KS algorithm have been published in [10].

The KS algorithm is presented in detail and its two phases are discussed, namely **Gamma Calculation** and **Iterative Feature Removal**. Both of these phases are designed around two heuristics, γ and δ, originally expressed by Koller and Sahami as Kullback-Leibler divergences, but they can be easily rewritten as conditional mutual information instead. The γ heuristic is calculated for each pair of variables in the data set, but only once, before the algorithm starts. Once the algorithm starts, it will use the values of γ in each iteration to assemble approximate Markov boundaries for every variable that has not yet been removed by previous iterations. The δ heuristic is then used to find the strongest Markov boundary in that iteration. The variable corresponding to the strongest boundary is removed from the data set. This way, the data set decreases in size with each iteration, up to a preconfigured size of Q variables, at which the algorithm stops. Apart from the Q parameter, the KS algorithm also takes a parameter K, which specifies the number of variables to assemble the approximate Markov boundaries with.

The Information Gain Thresholding (IGt) algorithm is also presented and parallels are drawn between IGt and KS. However, because IGt is so simple, it was used only as a comparative baseline for the accuracy of KS.

The experiment that compares KS to IGt was performed on binary document-term matrices constructed from the Reuters Corpus Volume 1. The algorithms were tasked to select the variables (matrix columns) most relevant to a selected class, as specified by the Corpus. After the algorithms made their selection, the reduced data sets were passed to Naive Bayes Classifiers and their accuracy was measured with respect to the selected class.

In effect, the comparative experiment evaluates how capable the algorithms are at capturing the classification relevance of the variables in the data set, as would be consumed by a simple classifier. This experiment design was chosen because both algorithms are usable as feature filters, namely they are algorithms that filter the features (variables) in a data set in advance of a classifier, in order to increase the accuracy of the classifier, reduce its complexity and make it consume less computational resources.

The experiment also performed Design Space Exploration on the two parameters of the KS algorithm, K and Q. The results show that KS outperforms IGt in almost all configurations, as was expected.

During the development of the algorithms and the experiment, novel optimization opportunities for the KS algorithm were discovered, as discussed by the second part of the chapter. Four such optimizations were implemented: **Gamma Decomposition**, the **Removed Features Database**, **In-iteration Parallelism** and the **Iteration Cache**. Of these four optimizations, the Removed Features Database is an **infrastructural improvement**, while the other three are **efficiency optimizations**, reducing the time needed by the KS algorithm to complete but without changing its output.

These original optimizations were evaluated in three individual experiments, each experiment comparing the unoptimized KS with a variant of KS containing an optimization, with the exception of the Removed Features Database, which was permanently enabled to record the behavior of KS.

The experiments contain original implementations of the KS algorithm, of IGt and of the four optimizations for KS. The formal description of the KS algorithm presented in this chapter is also original and has been published in [10].

The most effective optimization was the Iteration Cache, operating in the second phase of the algorithm, which reduced the duration of KS to an average of 0.55%, which is about 180 times faster, a remarkable acceleration. In comparison, In-iteration Parallelism performed modestly, halving the duration of the second phase, while Gamma Decomposition halved the duration of the first phase. It is important to note that the Iteration Cache is based on an idea mentioned by Koller and Sahami themselves, but their idea was never explored before.

The Gamma Decomposition optimization was not the most impressive one, but it formed the foundation of what was to become the *dcMI* optimization, the second and most significant contribution of the thesis, with efficiency gains far beyond expectations.

While the Iteration Cache makes the KS algorithm very efficient with respect to computational resources, it must be emphasized that KS has been obsoleted for good reasons: it is far from being the most accurate MBD algorithm and it cannot guarantee that its output is correct because it relies on heuristics and approximations. It also requires the specification of two parameters which cannot be determined straightforwardly. All these issues have been addressed by subsequent algorithms.

The second contribution: Accelerating an entire class of algorithms

Chapter 4 of the thesis presents the *dcMI* optimization, a novel and original optimization applicable to all algorithms that rely on conditional mutual information computed repeatedly on permutations of the variables of a data set. Because the widely used statistical G-test itself is conditional mutual information, the potential impact of *dcMI* is wide.

The *dcMI* optimization is surprisingly simple: conditional mutual information is rewritten as a sum of joint entropy terms, which are then cached and intensively reused across the computation of conditional mutual information for the variables of a data set. This process was called **decomposed conditional mutual information (dcMI)**. Note that using *dcMI* does not change the result of the conditional mutual information – it is **not an approximation**, but an **equivalent decomposition**.

Many Markov Boundary Discovery algorithms rely on the G-test to determine conditional independence in their operation. And because the algorithms must systematically apply the G-test on many permutations of variables (the two tested variables and the variables in the conditioning set), the *dcMI* optimization has a significant effect on their efficiency.

This chapter also includes an original analysis of the reuse factor of a data set, exploitable by *dcMI*. The analysis proves that the reuse of the joint entropy terms scales **quadratically** with the number of variables in the data set, therefore the efficiency gains of *dcMI* **become higher as the data set size increases**, a remarkable property.

Also included is an original alternative method of computing the degrees of freedom for a G-test, more efficient and more compatible with optimization structures than the method widely used by MBD algorithms.

In order to empirically demonstrate the efficiency gains brought by *dcMI*, an experiment was performed: the highly efficient and accurate IPC-MB algorithm was configured to apply the G-test optimized with $dcMI$ on data sets that were synthetically generated from publicly available Bayesian networks. The efficiency of this configuration of IPC-MB was compared directly with the efficiency of IPC-MB with an unoptimized G-test, but also with IPC-MB configured with implementations of the G-test enhanced with AD–trees, which retrieve the needed probability distributions from either pre-built static AD–trees or from dynamic AD–trees built at runtime.

The AD–tree is a special data structure which stores the sample counts of all the possible combinations of variables from the data set, or of only a subset thereof, depending on the type of AD–tree. **Static AD–trees** contain all the information needed to construct every possible probability distribution of samples from the data set, which allows for extremely fast G-tests but at the cost of nearly prohibitive memory consumption. This type of AD–tree is built in one pass and must be completed before any query. On the other hand, **dynamic AD–trees** are much more memory-efficient because they only expand as needed by the queries made by the G-test, with only minimal loss in time efficiency. The memory consumption remains relatively high, but it is far more manageable than a static AD–tree. Both types of AD–trees share another important shortcoming: while implementing them in source code is not particularly difficult, doing so *efficiently* is a different matter, requiring much more effort and commitment.

Thus, four configurations of the G-test were evaluated: unoptimized, optimized with static AD–trees, optimized with dynamic AD–trees and optimized with $dcMI$. The evaluation was performed on data sets instantiated from two public Bayesian networks, ALARM and ANDES, consisting of 37 variables and 223 variables respectively.

As expected, the $dcMI$ optimization exhibited the greatest efficiency gains. However, it was not expected that these gains would be as extreme as observed: $dcMI$ caused the G-test to be computed **21 times faster** while simultaneously consuming **3.6 less memory** than the next most efficient configuration, the G-test with dynamic AD–tree. Given the dimensions of the data set on which these extreme results were observed, namely 223 variables with $16,000$ samples, $dcMI$ is indeed remarkable. Note that all configurations in the experiment yielded the same values for every G-test performed. No configuration in the experiment used any approximations or estimations.

This experiment contains original implementations of the IPC-MB algorithm, of the four G-test configurations, including the implementation of $dcMI$ with its characteristic data structure, the Joint Entropy Table (JHT). Both static and dynamic AD–tree implementations were developed specifically for this experiment. These are very likely the most efficient publicly available Python implementations of AD–trees. In order to use Bayesian networks directly, an original grammar-based reader for the Bayesian Interchange Format was implemented, along with a full-valued random sampler for Bayesian networks.

The third contribution: Evaluating MBD algorithms in ideal conditions

Chapter 5 of the thesis contains the description of an original methodological improvement specific to the study, development, evaluation and validation of Markov Boundary Discovery algorithms: the usage of the d-separation criterion as the conditional independence (CI) test of the algorithms, computed directly on Bayesian networks, as opposed to synthesizing random data sets from the networks and then applying statistical CI tests. In laboratory conditions, where Bayesian networks are readily available, the d-separation criterion acts as a **perfect CI test**, providing ideal information to the algorithms, as opposed to information extracted a data set subject to biases, randomness and sample insufficiencies. This contribution originates from the work done on implementing the IPC-MB algorithm for the second contribution, described in Chapter 4. During the implementation of IPC-MB, it became obvious that a method of computing perfect CI tests is necessary, in order to write proper automatic tests.

It is important to emphasize that *automatic testing is at least as important in scientific software* as it is in commercial software. For this reason alone, algorithm researchers should consider using the d-separation criterion when validating their implementations. This chapter will also mention two MBD algorithms which were published with incorrect behavior, an avoidable situation had the d-separation criterion been used during their implementation.

By configuring an MBD algorithm to use the d-separation criterion as a CI test, applied on a selected Bayesian network, the ideal laboratory conditions are achieved for the study and development of existing or novel MBD algorithms. It is important to note that removing the randomness of the synthetic data sets makes these conditions *fully repeatable*, because d-separation is deterministic and there is no random data set involved. This chapter of the thesis discusses this specific methodological improvement, as published by Băncioiu and Brad [8].

Using d-separation when developing and evaluating MBD algorithms has four important advantages: greatly simplifies **automatic testing** for the algorithm implementation; provides **quick feedback** to the researcher, because d-separation is much faster to compute on a Bayesian network than a statistical test on a data set; it reveals the **true behavior of the algorithm**, unaffected by randomness or biases present in a synthetic or real-world data set; allows for the design of **novel performance metrics**, useful to study particular traits of the algorithms.

To exemplify the methodological advantages of using d-separation, two experiments were performed. **The first experiment** was performed directly on Bayesian networks, necessitating no data sets whatsoever. This experiment compared the absolute number of CI tests and the average size of the conditioning sets in the CI tests performed by two MBD algorithms, IPC-MB and IAMB. This reveals the true intrinsic behavior of the algorithms, unaffected by external randomness. **The second experiment** was performed by combining the Bayesian networks with data sets synthesized from them, in order to reliably evaluate the data efficiency of the two algorithms. Data efficiency was evaluated by measuring the average conditioning set size in CI tests for both algorithms, but also by calculating the percentage of statistical CI tests performed by the algorithms **that are correct**, i.e. in agreement with the d-separation criterion computed simultaneously with the statistical CI test.

Therefore, on top of the methodological improvement of using d-separation, the chapter also describes three proposed metrics, easily measurable when the d-separation criterion is used and very difficult without it: the total number of perfect CI tests performed by algorithms; the average conditioning set size of perfect CI tests; the percentage of **accurate** statistical CI tests performed by algorithms when validated against the corresponding perfect CI test (d-separation).

These two experiments required the original implementation of the IAMB algorithm; IPC-MB was already available from the experiment presented in Chapter 4.

Summary of contributions

The first contribution is a minor one, but it is built upon by the subsequent contributions. It consists of a case-study of the highly influential algorithm of Koller and Sahami (KS), first proposed in 1995 and now obsoleted by other algorithms. Due to its simplicity, it lends itself to straightforward examination and discussion. Studying the KS algorithm uncovered a simple trick that accelerated the computation of one of its heuristics. This trick is not remarkable on its own, but it was later generalized and expanded into a highly effective computational optimization, becoming the second contribution presented in this thesis. The case-study of the KS algorithm forms Chapter 3 of the thesis.

The second contribution is significant and consists of the spectacularly effective optimization of a computational step common to all the algorithms of interest, namely the computation of conditional mutual information found at the heart of the statistical G-test. It focuses on the theory underlying this computational step, often overlooked due to its (deceptive) simplicity. Specifically, the optimization relies on the properties of entropy and mutual information, the fundamental information-theoretic concepts, in order to exploit a massive reuse of terms that appear after a simple mathematical decomposition. An experimental evaluation, using the MBD algorithm named IPC-MB, revealed efficiency gains *two orders of magnitude* greater than the next-best known comparable optimization, while consuming far less computing resources and being simple to implement in source code. As mentioned above, this optimization stems from the case-study of the KS algorithm. It was named $dcMI$, an abbreviation of “decomposed conditional mutual information”. Chapter 4 discusses $dcMI$ in detail.

The third contribution is not concerned with how the Markov Boundary Discovery algorithms function, but with how they are designed, tested, validated and evaluated. Thus it is a methodological improvement. In short, this contribution consists of adding evaluation and validation steps that provide the algorithms with perfect information at runtime by relying on the d-separation criterion, a central aspect in the theoretical study of Bayesian networks. By using d-separation when testing and evaluating Markov Boundary Discovery algorithms, researchers can discover flaws in their algorithms far in advance of publication and can better understand the runtime behavior of their algorithms. This is not to say that researchers did not evaluate or validate their algorithms in the past. However, virtually all published algorithms were validated using statistical tests performed on randomly generated synthetic data sets of various sizes, which are a rich source of unwanted unpredictability. Using the d-separation criterion removes all unpredictability and randomness from the behavior of the algorithms, allowing them to operate in ideal conditions and perform at their theoretical best. Of course, computing d-separation requires the original Bayesian network, thus it is mostly limited to laboratory settings. Chapter 5 covers the use of d-separation for this specific purpose.

All the source code developed for the experiments presented in this thesis has been published under the GPLv3 as the Markov Boundary Toolkit (MBTK) [6], a Python library for the study and development of MBD algorithms, currently consisting of almost 16,000 lines of code. MBTK includes original implementations of the KS, IGt, IPC-MB and IAMB algorithms, of the $dcMI$ optimization, of static and dynamic AD–trees, of the G-test in four configurations (unoptimized, optimized with $dcMI$, optimized with static and dynamic AD–trees) and of a simple algorithm for d-separation. MBTK also includes tools for creating experiments: a Bayesian Interchange Format reader, a full-valued Bayesian network sampler and an experimental pipeline with result analysis.

Contents

Bibliography

[1] Lark - a parsing toolkit for python, 2020. URL https://github.com/lark-parser/lark.

[2] Constantin F Aliferis, Ioannis Tsamardinos, and Alexander Statnikov. Hiton: a novel Markov blanket algorithm for optimal variable selection. In *AMIA Annual Symposium Proceedings*, volume 2003, page 21. American Medical Informatics Association, 2003.

[3] Constantin F Aliferis, Alexander Statnikov, Ioannis Tsamardinos, Subramani Mani, and Xenofon D Koutsoukos. Local causal and markov blanket induction for causal discovery and feature selection for classification part i: algorithms and empirical evaluation. *Journal of Machine Learning Research*, 11(1), 2010.

[4] Iain Bancarz. *Conditional-Entropy Metrics for Feature Selection*. PhD thesis, University of Edinburgh. College of Science and Engineering. School of Informatics., 2005.

[5] Gavin Brown, Adam Pocock, Ming-Jie Zhao, and Mikel Luján. Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *Journal of machine learning research*, 13(Jan):27–66, 2012.

[6] Camil Băncioiu. MBTK, a library for studying Markov boundary algorithms. https://github.com/camilbancioiu/mbtk, 2020.

[7] Camil Băncioiu and Remus Brad. Accelerating causal inference and feature selection methods through G-test computation reuse. *Entropy*, 23(11), 2021. ISSN 1099-4300. doi: 10.3390/e23111501. URL https://www.mdpi.com/1099-4300/23/11/1501.

[8] Camil Băncioiu and Remus Brad. Analyzing markov boundary discovery algorithms in ideal conditions using the d-separation criterion. *Algorithms*, 15(4), 2022. ISSN 1999-4893. doi: 10.3390/a15040105. URL https://www.mdpi.com/1999-4893/15/4/105.

[9] Camil Băncioiu and Lucian Vințan. A comparison between two feature selection algorithms. In *Proceedings of ICSTCC 2017*, pages 242–247, 2017.

[10] Camil Băncioiu, Maria Vințan, and Lucian Vințan. Efficiency optimizations for Koller and Sahami's feature selection algorithm. *Romanian Journal of Information Science and Technology*, 22(1):85–99, 2019. ISSN 1453-8245. URL https://romjist.ro/abstract-620.html.

[11] T. M. Cover and Joy A. Thomas. *Elements of information theory*. Wiley-Interscience, Hoboken, N.J, 2nd ed edition, 2006. ISBN 978-0-471-24195-9.

[12] Robert G Cowell. Conditions under which conditional independence and scoring methods lead to identical selection of bayesian network models. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 91–97, 2001.

[13] Shunkai Fu and Michel C Desmarais. Fast Markov blanket discovery algorithm via local learning within single pass. In *Conference of the Canadian Society for Computational Studies of Intelligence*, pages 96–107. Springer, 2008.

[14] Shunkai Fu and Michel C Desmarais. Markov blanket based feature selection: a review of past decade. In *Proceedings of the world congress on engineering*, volume 1, pages 321–328. Newswood Ltd, 2010.

[15] Shunkai Fu, Michel Desmarais, and Weibin Chen. Reliability analysis of Markov blanket learning algorithms (1996-2010). In *Proceedings of the International Conference on Data Mining (DMIN)*, page 1. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2011.

[16] Kenji Fukumizu, Francis R. Bach, and Michael I. Jordan. Kernel dimension reduction in regression. *The Annals of Statistics*, 37(4):1871–1905, 2009. ISSN 00905364, 21688966. URL http://www.jstor.org/stable/30243690.

[17] Tian Gao and Qiang Ji. Efficient score-based markov blanket discovery. *International Journal of Approximate Reasoning*, 80:277–293, 2017. ISSN 0888-613X. doi: https://doi.org/10.1016/j.ijar.2016.09.009. URL https://www.sciencedirect.com/science/article/pii/S0888613X1630161X.

[18] Ben Glocker, Mirco Musolesi, Jonathan Richens, and Caroline Uhler. Causality in digital medicine. *Nature Communications*, 12, 2021. doi: https://doi.org/10.1038/s41467-021-25743-9.

[19] Isabelle Guyon, editor. *Feature extraction: foundations and applications*. Number v. 207 in Studies in fuzziness and soft computing. Springer-Verlag, Berlin ; New York, 2006. ISBN 978-3-540-35487-1. OCLC: ocm70886217.

[20] Mark Andrew Hall. *Correlation-based feature selection for machine learning*. PhD thesis, University of Waikato, 1999.

[21] David Heckerman, Dan Geiger, and David M. Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 1995. doi: 10.1007/BF00994016.

[22] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, Cambridge, MA, USA, 2009.

[23] Daphne Koller and Mehran Sahami. Toward optimal feature selection. In *In 13th International Conference on Machine Learning*, pages 284–292, 1995.

[24] Paul Komarek and Andrew W. Moore. A dynamic adaptation of AD-trees for efficient machine learning on large data sets. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, page 495–502, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1558607072.

[25] Thuc Duy Le, Tao Hoang, Jiuyong Li, Lin Liu, Huawen Liu, and Shu Hu. A fast pc algorithm for high dimensional causal discovery with multi-core pcs. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 16(5):1483–1495, 2019. doi: 10.1109/TCBB.2016.2591526.

[26] Changki Lee and Gary Geunbae Lee. Information gain and divergence-based feature selection for machine learning-based text categorization. *Information Processing & Management*, 42(1):155–165, January 2006. ISSN 03064573. doi: 10.1016/j.ipm.2004.08.006.

[27] E. L. Lehmann and Joseph P. Romano. *Testing statistical hypotheses*. Springer, third edition, 2005. ISBN 0-387-98864-5.

[28] David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5: 361–397, 2004. URL http://www.jmlr.org/papers/v5/lewis04a.html.

[29] Dimitris Margaritis and Sebastian Thrun. Bayesian network induction via local neighborhoods. In *Advances in neural information processing systems*, pages 505–511, 2000.

[30] Andrew Moore and Mary S Lee. Cached sufficient statistics for efficient machine learning with large datasets. *Journal of artificial intelligence research*, 8:67–91, 1998.

[31] Ionel Daniel Morariu. *Contributions to Automatic Knowledge Extraction from Unstructured Data*. PhD thesis, "Lucian Blaga" University of Sibiu (supervisor: Prof. L. Vințan), 2007.

[32] Teppo Niinimäki and Pekka Parviainen. Local structure discovery in bayesian networks. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, pages 634–643, 2012.

[33] E Pasero, A Montuori, W Moniaci, and Giovanni Raimondo. *An application of data mining to PM10 level medium-term prediction*. PhD thesis, International Environmental Modelling and Software Society, 2008.

[34] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Kaufmann, San Francisco, Calif, rev. 2. print., 12. [dr.] edition, 2008. ISBN 978-1-55860-479-7.

[35] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[36] Jose M Pena, Roland Nilsson, Johan Björkegren, and Jesper Tegnér. Towards scalable and data efficient learning of Markov boundaries. *International Journal of Approximate Reasoning*, 45(2):211–232, 2007.

[37] Stuart J. Russell and Peter Norvig. *Artificial intelligence: A Modern Approach*. Prentice Hall series in artificial intelligence. Prentice Hall, Upper Saddle River, 3rd ed edition, 2010. ISBN 978-0-13-604259-4.

[38] Marco Scutari. Bayesian network constraint-based structure learning algorithms: Parallel and optimized implementations in the bnlearn r package. *Journal of Statistical Software*, 77, 03 2017. doi: 10.18637/jss.v077.i02.

[39] Marco Scutari. bnlearn - an r package for Bayesian network learning and inference, 2020. URL https://www.bnlearn.com/bnrepository/.

[40] Tomi Silander and Petri Myllymäki. A simple approach for finding the globally optimal bayesian network structure. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, UAI'06, page 445–452, Arlington, Virginia, USA, 2006. AUAI Press. ISBN 0974903922.

[41] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search.* MIT press, 2000.

[42] Er Statnikov, Ioannis Tsamardinos, Laura E. Brown, and Constantin F. Aliferis. Causal explorer: A matlab library of algorithms for causal discovery and variable selection for classification, 2009.

[43] Ioannis Tsamardinos, Constantin Aliferis, Alexander Statnikov, and Er Statnikov. Algorithms for large scale Markov blanket discovery. In *In The 16th International FLAIRS Conference, St*, pages 376–380. AAAI Press, 2003.

[44] Ioannis Tsamardinos, Constantin Aliferis, Alexander Statnikov, and Er Statnikov. Algorithms for large scale Markov blanket discovery. In *In The 16th International FLAIRS Conference, St*, pages 376–380. AAAI Press, 2003.

[45] Ioannis Tsamardinos, Constantin F Aliferis, and Alexander Statnikov. Time and sample efficient discovery of Markov blankets and direct causal relations. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 673–678. ACM, 2003.

[46] Hal R. Varian. Causal inference in economics and marketing. *Proceedings of the National Academy of Sciences*, 113(27):7310–7315, 2016. doi: 10.1073/pnas.1510479113. URL https://www.pnas.org/doi/abs/10.1073/pnas.1510479113.

[47] Xingyu Wu, Bingbing Jiang, Yan Zhong, and Huanhuan Chen. Tolerant markov boundary discovery for feature selection. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 2261–2264, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368599. doi: 10.1145/3340531.3415927. URL https://doi.org/10.1145/3340531.3415927.

[48] Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In *ICML*, volume 97, pages 412–420, 1997.