Rezumat

Despre Eficiența Testelor de Independență Condiționată și Utilizarea lor cu Acuratețe în Algoritmii pentru Descoperirea Granițelor Markov

Teză de doctorat
Camil Băncioiu

Keywords: deducția cauzalității, selecția atributelor, teoria informației, rețele bayesiene, independență condiționată, teste statistice, testul G, reutilizarea rezultatelor, criteriul de d-separație

Această teză prezintă contribuții noi și originale pentru algoritmii care descoperă granițe Markov (Markov Boundary Discovery, MBD), concentrându-se pe felul în care acești algoritmi folosesc testele de independență condiționată (conditional independence, CI). Algoritmii MBD sunt o clasă versatilă de algoritmi care dezvăluie informația cauzală din seturi de date. Acest fapt face ca algoritmii MBD să fie aplicabili și în probleme de deducție de cauzalitate, precum și în probleme de selecție a atributelor.

Datorită importanței majore pe care informația cauzală o prezintă în marea majoritate a domeniilor științei, extragerea sa din seturi de date în mod eficient și corect cu algoritmi MBD a fost cercetată intens în trecut și rămâne în continuare un subiect foarte activ de cercetare. Cu toate acestea, testele de independență condiționată atât de necesare acestor algoritmi nu au primit aceeași atenție. Drept urmare, mulți algoritmi MBD au împrumutat abordări de teste CI unul de la celălalt, cu variații sumare. Aceasta este o problemă serioasă, pentru că testele CI constituie un blocaj de performanță sever pentru algoritmi, consumând aproximativ 97% din timpul lor total de rulare, iar acuratețea și corectitudinea testelor CI este decisivă în funcționarea lor.

Contribuțiile prezentate în această teză se adresează acestor neajunsuri. Ele se concentrează pe două aspecte esențiale ale testelor CI, anume eficiența lor de calcul și metodologia de evaluare a acurateții lor. Teoria pe care aceste contribuții se bazează este fundamentală, simplă și bine cunoscută, însă în ciuda simplității lor, îmbunătățirile aduse sunt extrem de eficace, cum este detailat în această teză și în articolele publicate.

Prima contribuție: Un studiu al algoritmului lui Koller și Sahami

Capitolul 3 al tezei conține studiul de caz al algoritmului lui Koller și Sahami [23], astăzi depășit, însă foarte influent. Este considerat ca fiind primul algoritm care să folosească conceptul de graniță Markov din teoria informației, cu aceasta deschizând drumul pentru o nouă clasă de algoritmi. Trebuie accentuat faptul că algoritmul KS a fost inițial publicat

ca un algoritm de selecție a atributelor, însă poate fi aplicat și în probleme de deducție a cauzalității, dacă este necesar.

Acest studiu de caz se concentrează nu doar pe algoritmul KS în sine, ci și pe un **experiment comparativ original** efectuat pe algoritmul KS și un alt algoritm fundamental, Information Gain Thresholding (IGt). De asemenea, va fi discutată și o colecție de **optimizări noi** pentru algoritmul KS, optimizări care au reieșit din studiul și implementarea algoritmului. Proiectarea și dezvoltarea acestor optimizări au fost pietre de temelie pentru dezvoltarea contribuțiilor ulterioare, discutate în capitolele următoare din teză. Experimentul comparativ menționat mai sus a fost publicat în [9], în timp ce optimizările originale pentru algoritmul KS au fost publicate în [10].

Algoritmul KS este prezentat în detaliu în acest capitol și cele două faze ale sale sunt discutate, anume **Gamma Calculation** (Calculul lui Gamma) și **Iterative Feature Removal** (Eliminarea Iterativă a Atributelor). Amândouă faze ale algoritmului sunt concepute în jurul a două euristici, γ și δ, inițial descrise de Koller și Sahami înșiși ca fiind divergențe Kullback-Leibler, însă pot fi rescrise foarte ușor ca și informație mutuală condiționată. Euristica γ este calculată pentru toate perechile de variabile (atribute) ale setului de date, dar numai o singură dată, înainte de începerea algoritmului. După ce algoritmul începe, el va folosi valorile γ în fiecare iterație pentru a asambla granițe Markov aproximative pentru fiecare variabilă ce n-a fost încă eliminată de iterațiile anterioare. Apoi, euristica δ este folosită pentru a găsi cea mai "puternică" graniță Markov în acea iterație. Variabila care corespunde celei mai puternice granițe este eliminată din setul de date, astfel setul de date scade în mărime cu fiecare iterație până la o mărime preconfigurată de Q variabile, când algoritmul se oprește. În afară de parametrul Q, algoritmul KS mai are un parametru numit K, care specifică numărul de variabile care să fie alese pentru a forma granițele Markov aproximative în timpul iterațiilor.

Algoritmul Information Gain Thresholding (IGt) este de asemenea prezentat în acest capitol și sunt prezentate paralele între IGt și KS. Totuși, deoarece IGt este foarte simplu, a fost folosit doar ca un reper minim de acuratețe pentru KS.

Experimentul care compară algoritmii KS și IGt a fost efectuat pe matrici binare de tip document-termen, construite din Reuters Corpus Volume 1. Cei doi algoritmi au fost puși să selecteze care variabile (coloane ale matricilor) sunt cele mai relevante pentru o variabilă de clasificare, specificată de Corpus. După ce algoritmii au finalizat selecția, seturile de date astfel reduse au fost folosite pentru a antrena și evalua clasificatoare bayesiene naive.

În esență, acest experiment comparativ evaluează cât de capabili sunt algoritmii în a capta relevanța pentru clasificare a variabilelor din setul de date, așa cum va fi consumată de un clasificator simplu. Această formă de experiment a fost aleasă pentru că ambii algoritmi sunt aplicabili ca filtre de atribute, adică sunt algoritmi care filtrează atributele (variabilele) dintr-un set de date înaintea aplicării unui clasificator, în scopul de a crește acuratețea clasificatorului, de a-i reduce complexitatea și de a-l face să consume mai puține resurse de calcul.

Experimentul a efectuat și explorare de spațiu de parametri, variind valorile celor doi parametri ai lui KS, anume K și Q. Rezultatele arată că algoritmul KS este cu mult superior lui IGt în aproape toate configurațiile, așa cum era de așteptat.

În timpul implementării algoritmilor și a experimentului, au fost descoperite noi oportunități de optimizare pentru algoritmul KS, așa cum este descris în a doua parte a capitolului. Patru astfel de optimizări au fost implementate: **Gamma Decomposition** (Descompunerea lui Gamma), **Removed Features Database** (Baza de date a Atributelor

Eliminate), **In-iteration Parallelism** (Paralelism Intraiterație) și **Iteration Cache** (Cache de Iterație). Dintre aceste patru optimizări, Removed Features Database este o **îmbunătățire infrastructurală**, în timp ce celelalte trei sunt **optimizări de eficiență**, reducând timpul necesar algoritmului KS pentru a ajunge la finalul rulării, însă fără a-i schimba rezultatele.

Aceste optimizări originale au fost evaluate în trei experimente individuale, fiecare experiment comparând algoritmul KS neoptimizat cu o variantă a sa modificată cu una din optimizări, cu excepția Removed Features Database, care a fost pornită în permanență pentru a înregistra comportamentul algoritmului.

Experimentele conțin implementări originale ale algoritmilor KS și IGt și a celor patru optimizări pentru KS. Descrierea formală a algoritmului KS prezentată în acest capitol este de asemenea originală și a fost publicată în [10].

Cea ma eficace optimizare a fost Iteration Cache, care operează în cea de-a doua fază a algoritmului și i-a redus durata la o medie de 0.55%, adică de aproximativ 180 mai rapid, o remarcabilă accelerare. In-iteration Parallelism s-a comportat comparativ mai modest, doar înjumătățind durata celei de-a doua faze, în timp ce Gamma Decomposition a înjumătățit durata primei faze. Este important de observat că Iteration Cache este bazată pe o idee menționată de Koller și Sahami înșiși, însă ideea lor nu a fost niciodată explorată până acum.

Optimizarea Gamma Decomposition nu a fost cea mai impresionantă, însă a pus bazele la ceea ce va deveni optimizarea $dcMI$, a doua și cea mai semnificativă contribuție a tezei, cu creșteri de eficiență depășind cu mult orice așteptări.

Chiar dacă Iteration Cache crește radical eficiența algoritmului KS din punctul de vedere al resurselor de calcul, e important de reținut că acest algoritm este considerat depășit din motive bune: este departe de a fi cel mai exact algoritm MBD și nu poate garanta corectitudinea rezultatelor sale deoarece se bazează pe euristici și aproximări. De asemenea, are nevoie de specificarea valorilor a doi parametri, care nu pot fi determinate ușor. Toate aceste probleme au fost rezolvate de algoritmi ulteriori.

A doua contribuție: Accelerarea unei clase întregi de algoritmi

Capitolul 4 al tezei prezintă optimizarea $dcMI$, o nouă și originală optimizare aplicabilă tuturor algoritmilor care calculează în mod repetat informația mutuală condiționată a unor permutări de variabile dintr-un set de date. Deoarece testul statistic G, cu largă utilizare în domeniu, este el însuși informație mutuală condiționată, impactul potențial al lui $dcMI$ este semnificativ.

Optimizarea $dcMI$ este surprinzător de simplă: informația mutuală condiționată este rescrisă ca o sumă de termeni de entropie comună, ale căror valori sunt păstrate după primul calcul și sunt refolosite intens pe întreg parcursul algoritmului în calculul informației mutuale condiționate. Acest proces se numește **decomposed conditional mutual information (dcMI)** (informație mutuală condiționată descompusă). A se observa că folosirea lui $dcMI$ nu schimbă rezultatul calculelor – $dcMI$ **nu este o aproximare**, ci este o **descompunere echivalentă**.

Mulți algoritmi MBD se bazează pe testul statistic G pentru a stabili independența condiționată între variabile în timpul operării. Și pentru că acești algoritmi trebuie să aplice testul G în mod sistematic pe multe permutări ale acelorași variabile din setul de

date (perechea de variabile testate, împreună cu variabilele din mulțimea condițională din test), optimizarea *dcMI* are un efect semnificativ asupra eficienței lor.

Acest capitol include și o analiză originală a factorului de reutilizare al unui set de date, exploatabil de *dcMI*. Această analiză demonstrează că reutilizarea termenilor de entropie comună crește **polinomial** în numărul de variabile ale setului de date, deci câștigul de eficiență dat de *dcMI* **crește cu cât setul de date e mai mare**, o proprietate remarcabilă.

De asemenea este inclusă în capitol și o metodă alternativă și originală de a calcula gradele de libertate pentru un test G (degrees of freedom), metodă mai eficientă și mai compatibilă cu structuri de optimizare decât metoda des folosită de algoritmii MBD.

Pentru a demonstra empiric câștigul de eficiență adus de *dcMI*, a fost efectuat un experiment: IPC-MB, un algoritm MBD eficient și exact, a fost configurat să aplice testul G optimizat cu *dcMI* pe seturi de date sintetizate din rețele Bayesiene accesibile public. Eficiența acestei configurații de IPC-MB a fost comparată direct cu eficiența lui IPC-MB folosind testul G neoptimizat, dar și cu IPC-MB configurat cu implementări de test G îmbunătățite cu arbori omnidimensionali (AD–tree), implementări care extrăgeau distribuțiile de probabilitate necesare testului fie din arbori omnidimensionali statici preconstruiți, fie din arbori omnidimensionali construiți în timpul rulării.

Un arbore omnidimensional este o structură de date specială care reține numărul de eșantioane al oricăror combinații posibile de variabile dintr-un set de date, sau doar a unor combinații anume, în funcție de tipul de arbore. **Arborii omnidimensionali statici** conțin toată informația necesară construirii oricărei distribuții de probabilitate a variabilelor din setul de date, ceea ce permite calculul extrem de rapid al testului G, dar cu costuri de memorie aproape intractabile. Acest tip de arbore omnidimensional este construit într-o singură etapă indivizibilă și trebuie finalizat înainte de orice interogare a sa. În schimb, **arborii omnidimensionali dinamici** sunt mult mai eficienți cu memoria consumată pentru că sunt expandați doar la nevoie, în timpul interogărilor făcute de testul G la rulare. Construirea la rulare are impact minimal asupra eficienței de interogare. Consumul de memorie rămâne totuși ridicat, însă este mult mai acceptabil decât pentru un arbore omnidimensional static. Ambele tipuri de arbori mai au un dezavantaj: cu toate că implementarea lor în cod-sursă nu e tocmai dificilă, dezvoltarea unei implementări *eficiente* este o cu totul altă problemă, necesitând mult mai mult efort și dedicație.

Așadar, patru configurații ale testului G au fost evaluate: neoptimizat, optimizat cu arbore omnidimensional static, optimizat cu arbore omnidimensional dinamic și optimizat cu *dcMI*. Evaluarea a fost efectuată pe seturi de date sintetizate din două rețele Bayesiene publice, ALARM și ANDES, conținând 37 de variabile și respectiv 223 variabile.

Cum era de așteptat, optimizarea *dcMI* a prezentat cel mai mare câștig de eficiență. Totuși, rezultatele au fost atât de extreme încât au depășit orice așteptare: *dcMI* a cauzat **o accelerare a testului G de 21 de ori**, în timp ce consuma **de 3.6 ori mai puțină memorie** față de următoarea configurație ca eficiență, arborele omnidimensional dinamic. Dată fiind dimensiunea setului de date pe care aceste rezultate extreme au fost observate, anume 223 de variabile și $16,000$ de eșantioane, *dcMI* este într-adevăr remarcabil. A se reține că toate configurațiile din experiment au ajuns la același rezultat pentru fiecare test G efectuat. Nicio configurație nu a folosit vreo aproximare sau estimare.

Acest experiment conține implementări originale ale algoritmului IPC-MB, ale celor patru configurații de test G, inclusiv a implementării lui *dcMI* cu structura sa de date caracteristică, Joint Entropy Table (JHT, Tabel de Entropie Comună). Ambele tipuri

de arbore omnidimensional, static și dinamic, au fost implementate special pentru acest experiment. Este foarte probabil că aceste implementări de arbore omnidimensional să fie cele mai eficiente implementări publice în Python. Pentru a folosi rețele Bayesiene în mod direct, a fost implementat un interpretor original de Bayesian Interchange Format bazat pe o gramatică formală, împreună cu un sintetizator de eșantioane aleatoare.

A treia contribuție: Evaluarea algoritmilor MBD în condiții ideale

Capitolul 5 al tezei conține descrierea unei îmbunătățiri metodologice originale specifică studiului, dezvoltării, evaluării și validării algoritmilor MBD: folosirea criteriului de d-separație ca și test de independență condițională în algoritmi, calculat direct pe rețele Bayesiene, în opoziție cu metodologia de evaluare folosind teste CI statistice pe seturi de date sintetice. În condiții de laborator, în care rețelele Bayesiene sunt ușor accesibile, criteriul d-separației se comportă ca un **test CI perfect**, oferind algoritmilor informații ideale, în contrast cu informațiile extrase dintr-un set de date supus dezechilibrelor probabilistice, aleatorului și lipselor de eșantioane. Această contribuție își are originea în munca depusă pentru implementarea algoritmului IPC-MB din a doua contribuție, descrisă în Capitolul 4. În timpul implementării lui IPC-MB, a devenit evident că este necesar un mod de a calcula teste CI perfecte pentru a putea scrie teste automate.

E important de accentuat că *testarea automată este cel puțin la fel de importantă pentru software-ul științific* cum este pentru software-ul comercial. Din acest motiv, cercetătorii care lucrează la algoritmi ar trebui să ia în considerare folosirea criteriului de d-separație când își validează implementările. Acest capitol mai menționează doi algoritmi MBD care au fost publicați cu comportament incorect, o situație evitabilă dacă ar fi fost folosit criteriul d-separației în timpul implementării și validării.

Prin configurarea unui algoritm MBD cu criteriul d-separației ca test CI, aplicat pe o rețea Bayesiană anume, se ajunge la condiții ideale de laborator în care să se studieze și să se dezvolte algoritmi MBD noi sau deja existenți. E important de menționat că eliminarea aleatorului caracteristic seturilor de date sintetice face ca aceste condiții să fie *complet repetabile*, deoarece d-separația este deterministică și nu sunt implicate date aleatoare. Acest capitol din teză discută această îmbunătățire metodologică, așa cum a fost publicată în [8].

Folosirea d-separației în dezvoltarea și evaluarea algoritmilor MBD are patru avantaje: simplificarea semnificativă a **testării automate** pentru implementarea algoritmului; cercetătorii au acces la **feedback rapid** al comportamentul algoritmului, deoarece d-separația este mult mai ușor de calculat pe o rețea Bayesiană decât un test statistic peste un set de date; este dezvăluit **comportamentul real al algoritmului**, neafectat de aleatorul și dezechilibrele din seturile de date sintetice sau cele culese din lumea reală; devine posibilă proiectarea de **metrici noi de performanță**, utile în studiul anumitor trăsături ale algoritmilor.

Pentru a exemplifica avantajele metodologice ale folosirii d-separației, au fost efectuate două experimente. **Primul experiment** a fost efectuat direct pe rețele Bayesiene, fără a necesita vreun set de date. Acest experiment a comparat numărul absolut de teste CI și mărimea medie a mulțimilor condiționale din testele CI efectuate de către doi algoritmi, IPC-MB și IAMB. Astfel se dezvăluie adevăratul comportament intrinsec al algoritmilor, neafectat de factori aleatori externi. **Al doilea experiment** a fost efectuat combinând rețelele Bayesiene cu seturi de date sintetizate din acestea pentru a evalua

eficiența informațională (data efficiency) a celor doi algoritmi. Eficiența informațională a fost evaluată măsurând mărimea medie a mulțimilor condiționale din testele CI pentru ambii algoritmi, dar a mai fost calculat procentul de teste CI statistice **corecte** efectuate de algoritmi, adică acele teste în acord cu criteriul d-separației corespunzător, calculat simultan cu testul CI statistic.

Așadar, pe lângă îmbunătățirea metodologică de folosire a d-separației, acest capitol descrie și trei metrici propuse, ușor calculabile când criteriul d-separației este folosit dar foarte dificile în absența sa, anume: numărul total de teste CI perfecte efectuate de algoritmi; mărimea medie a mulțimii condiționale din testele CI perfecte; procentul de teste CI statisice **corecte** efectuate, conform validării cu testul CI perfect (d-separație).

Aceste două experimente au necesitat o implementare originală a algoritmului IAMB; IPC-MB era deja disponibil din experimentul prezentat în Capitolul 4.

Rezumat al contribuțiilor

Prima din contribuții este minoră în sine, dar formează baza peste care sunt construite contribuțiile ce urmează. Ea constă într-un studiu de caz al algoritmului lui Koller și Sahami (KS), foarte influent la vremea sa, propus în 1995, însă depășit în prezent de alți algoritmi. Totuși, datorită simplității sale, algoritmul KS este un subiect bun de studiu, examinare și discuție. Studiind acest algoritm, a fost găsit un truc simplu care accelerează calculul uneia din euristicile sale. Acest truc de calcul nu este remarcabil în sine, dar a fost generalizat și extins, producând o optimizare de calcul cu largă aplicabilitate, foarte eficace și care constituie subiectul celei de-a doua contribuții a acestei teze. Acest studiu de caz al algoritmului KS se regăsește în Capitolul 3 al tezei.

A doua contribuție este semnificativă și constă în optimizarea calculului unui pas comun al tuturor algoritmilor de interes pentru această teză, optimizare de o eficacitate spectaculoasă. Mai exact, este o optimizare pentru calculul informației mutuale condiționate, care se regăsește în esența testului statistic G. Această optimizare țintește teoria care stă la baza acestui pas de calcul, adesea ignorat datorită simplității sale (înșelătoare). Optimizarea însăși se bazează pe proprietăți ale entropiei informaționale și ale informației mutuale, elemente fundamentale ale teoriei informației, în scopul de a exploata o masivă reutilizare de termeni care apar după o simplă descompunere matematică. O evaluare experimentală folosind algoritmul MBD numit IPC-MB a dezvăluit creșteri ale eficienței ale $dcMI$ cu *două ordine de mărime* mai înalte decât ale următoarei optimizări ca eficiență, dintre optimizările cunoscute ca fiind comparabile cu cea discutată. Simultan, optimizarea propusă în contribuție a consumat mult mai puține resurse de calcul în general, fiind totodată ușor de implementat în cod-sursă. Cum a fost menționat anterior, această optimizare își are originea în studiul de caz al algoritmului KS. Optimizarea a fost numită $dcMI$, o abreviere a "informației mutuale condiționate descompuse" (decomposed conditional mutual information). Capitolul 4 discută optimizarea $dcMI$ în detaliu.

A treia contribuție nu are ca subiect funcționarea algoritmilor MBD, în schimb se concentrează pe felul în care ele sunt proiectate, testate, validate și evaluate. Așadar, aceasta este o îmbunătățire metodologică. Pe scurt, această contribuție constă în introducerea unor etape speciale de evaluare și validare în dezvoltarea algoritmilor MBD, etape în care algoritmii rulați și studiați primesc informații perfecte provenite de la criteriul de separație direcțională (d-separație, d-separation), un criteriu central în teoria rețelelor Bayesiene. Folosind criteriul de d-separație în timpul testării și evaluării algoritmilor MBD,

cercetătorii pot descoperi erori în algoritmii lor cu mult înainte de publicare și pot înțelege mai bine comportamentul algoritmilor lor în timpul rulării. Asta nu înseamnă că până acum cercetătorii nu își validau și evaluau algoritmii. Dar marea majoritate a algoritmilor publicați au fost validați folosind teste statistice, efectuate pe seturi de date sintetizate aleator, de varii dimensiuni, care sunt o sursă foarte bogată de impredictibilități aleatoare. Însă folosirea criteriului de d-separație înlătură cu totul orice formă de impredictibilitate și aleator din comportamentul algoritmilor, permițându-le să opereze în condiții ideale și să își atingă performanțele teoretice. Desigur, calcularea criteriului de d-separație necesită rețeaua Bayesiană originală, deci este în mare parte limitat la condiții de laborator. Capitolul 5 acoperă utilizarea d-separației întru acest scop.

Tot codul-sursă dezvoltat pentru experimentele prezentate în această teză a fost publicat sub GPLv3 cu numele de Markov Boundary Toolkit (MBTK) [6], o bibliotecă software pentru Python destinată studiului și dezvoltării algoritmilor MBD, la momentul actual conținând 16,000 de linii de cod. MBTK conține implementări originale ale algoritmilor KS, IGt, IPC-MB și IAMB, a optimizării $dcMI$, ale arborilor omnidimensionali statici și dinamici (static / dynamic AD–tree), precum și a unui simplu algoritm pentru calculul d-separației. MBTK mai conține și unelte pentru crearea experimentelor: un interpretor de Bayesian Interchange Format, un motor de eșantionare cu valori complete și componentele necesare procesului de rulare a experimentelor, incluzând procesarea rezultatelor.

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 Vinţan. A comparison between two feature selection algorithms. In *Proceedings of ICSTCC 2017*, pages 242–247, 2017.

[10] Camil Băncioiu, Maria Vinţan, and Lucian Vinţan. 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. Vinţan), 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.