Advances in Artificial Intelligence

Research in Computing Science

Research in Computing Science, Año 19, Volumen 149, No. 3, marzo de 2020, es una publicación mensual, editada por el Instituto Politécnico Nacional, a través del Centro de Investigación en Computación. Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othon de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, Ciudad de México, Tel. 57 29 60 00, ext. 56571. https://www.rcs.cic.ipn.mx. Editor responsable: Dr. Grigori Sidorov. Reserva de Derechos al Uso Exclusivo del Título No. 04-2019-082310242100-203. ISSN: en trámite, ambos otorgados por el Instituto Politécnico Nacional de Derecho de Autor. Responsable de la última actualización de este número: el Centro de Investigación en Computación, Dr. Grigori Sidorov, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othon de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738. Fecha de última modificación 30 de agosto de 2019.

Las opiniones expresadas por los autores no necesariamente reflejan la postura del editor de la publicación.

Research in Computing Science, year 19, Volume 149, No. 3, March 2020, is published monthly by the Center for Computing Research of IPN.

The opinions expressed by the authors does not necessarily reflect the editor's posture.

Advances in Artificial Intelligence

Héctor-Gabriel Acosta-Mesa
Efrén Mezura-Montes
Luis Enrique Sucar
Nicandro Cruz-Ramírez
Felipe Orihuela-Espina
Samuel Montero
Jenny Betsabé Vázquez-Aguirre (eds.)

SEP
SECRETARÍA DE
EDUCACIÓN PÚBLICA

Instituto Politécnico Nacional
“La Técnica al Servicio de la Patria”

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2020

ISSN: 1870-4069

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional "Adolfo López Mateos", Zacatenco
07738, México D.F., México

http://www.rcs.cic.ipn.mx
http://www.ipn.mx
http://www.cic.ipn.mx

The editors and the publisher of this journal have made their best effort in preparing this special issue, but make no warranty of any kind, expressed or implied, with regard to the information contained in this volume.

Indexed in LATINDEX, DBLP and Periodica

Electronic edition

ISSN 1870-4069

Table of Contents

Page

Thematic Section: New Trends in Computational Intelligence and Applications

Guest Editors: Héctor-Gabriel Acosta-Mesa, Efrén Mezura-Montes

Thematic Section: Causal Reasoning

Guest Editors: Luis Enrique Sucar, Nicandro Cruz-Ramírez, Felipe Orihuela-Espina, Samuel Montero, Jenny Betsabé Vázquez-Aguirre

Regular Papers

Thematic Section
“New Trends in Computational Intelligence and Applications”

Héctor-Gabriel Acosta-Mesa
Efrén Mezura-Montes (eds.)

ISSN 1870-4069

Data-Driven Bayesian Network Learning: Towards a Bi-Objective Approach to Address the Bias-Variance Decomposition

Vicente Josué Aguilera Rueda, Nicandro Cruz Ramírez,
Efrén Mezura Montes

Universidad Veracruzana,
Centro de Investigación en Inteligencia Artificial (CIIA),
Mexico

{vaguilera,ncruz,emezura}@uv.mx

Abstract. We present a novel bi-objective approach to address the data-driven learning problem of Bayesian networks. Both the log-likelihood and the complexity of each candidate Bayesian network are considered as objectives to be optimized by our proposed algorithm named Non-dominated Sorting Genetic Algorithm for learning Bayesian networks (NS2BN) which is based on the well-known NSGA-II algorithm. The core idea is to reduce the implicit selection bias-variance decomposition while identifying a set of competitive models using both objectives. Numerical results suggest that, in stark contrast to the single-objective approach, our bi-objective approach is useful to find competitive Bayesian networks with a balanced trade-off between accuracy and complexity.

Keywords: Bayesian networks, bias-variance, NSGA-II.

1 Introduction

A way to build a Bayesian Network (BN) is adopting a data-driven inductive approach; in this case, the learning task is framed as a combinatorial optimization problem with two components: a metric to assess the quality of each BN candidate, and a search procedure to move through the space of candidate networks.

In data-driven BN learning, it is common to implement metrics in the form of a penalized log-likelihood (LL) function, as minimum description length (MDL). While adding an edge to a BN never decreases the likelihood –and hence irrelevant arcs should be discarded– adding extra arcs leads to two main problems: the overfitting problem and densely connected networks. To avoid complex networks, a penalty term is used. However, complex networks may have a low LL score value but overfit the model, while a high penalty term may incur in underfitting. Thus, it is desirable to have networks with a suitable balance between the goodness of fit (accuracy) and complexity.

Some researchers point out that the trade-off between accuracy and complexity should be featured as a bi-objective problem [5, 11] however, to the best of our knowledge, estimation of those values has not been previously used in a bi-objective approach for model selection in BN. Our work addresses this combination. The remainder of this paper is structured as follows: Section 2 describes related work. In Section 3, background about BN, MDL and bi-objective optimization problem are presented. In Section 4, our proposed algorithm is described. Section 5 presents the experiments and discusses the results. The concluding section summarizes the findings and gives an account for future work.

2 Related Work

There exist two main approaches to the use of crude MDL to learn BN: to find the true model (that has given rise to the data), known as the *gold-standard* network [8] and to find a model with a good trade-off between the accuracy and complexity [6]. Accuracy in this context refers to the computation of the log-likelihood of the data given a BN structure; it should not be confused with classification accuracy (see Equation 1).

Cruz-Ramírez et al. [3], performed an exhaustive experiment with four-node networks. Therefore, eventhough these results show how crude MDL produces well-balanced models in terms of complexity and log-likelihood, those experiments have a limited scope of four-node networks and they left for future work to explore the search procedure.

Previous studies have addressed the BN model selection problem using evolutionary algorithms, for instance, see [2, 13, 10]. However, none of them has tackled the problem in a multi-objective way.

Lastly, the work of Ross and Zuviria [12] uses a multi-objective genetic approach to induce dynamic BNs from data with a trade-off between likelihood and complexity. This work is focused on the modeling of biological phenomena that typically requires low-connectivity networks. However, to the best of our knowledge, this work is the only one with multi-objective criteria learning. Although, is in the context of dynamic BN.

In summary, the learning problem of BN using MDL as a metric has been dealt with mainly as a single-objective problem. However, it is possible that one objective tends to dominate the search procedure and will also add bias-variance decomposition to the kind of result obtained.

3 Background

3.1 Bayesian Networks

A BN is a graphical model that represents a joint probability distribution over a set of random variables $\{X_1, \ldots, X_n\}$. BNs are represented as a pair (G, Θ), where the directed acyclic graph (DAG) is represented by $G = (U, E_G)$; U is the set of nodes or random variables, and E_G is the set of arcs that represent the probabilistic relationship among these variables. The parents of X_i are denoted PA_i; X_i is independent of its non-descendant variables given its parents.

Thus, Θ is a set of parameters which quantify the network. The joint probability distribution can be recovered from local conditional probability distributions as:
$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(x_i|PA_i)$.

3.2 Minimum Description Length

The crude definition of MDL [6] is of the form:

$$MDL = -logP(D|\Theta) + \frac{k}{2}\, log\, n, \quad (1) \qquad\qquad k = \sum_{i=1}^{m} q_i(r_i - 1), \quad (2)$$

where D is the dataset, Θ represents the parameters of the model, k is the dimension of the model, and n is the sample size. The parameter Θ is the corresponding local probability distribution for each node in the network. The dimension of the model (k) is given by Equation 2.

For the case of Equation 2, m is the number of variables, q_i is the number of possible configurations of $PA_i; X_i$ and r_i is the number of values of the variable.

The first term of Equation 1 measures the accuracy of the model (f_1) and the second term measures the complexity (f_2). The complexity of a BN is proportional to the number of arcs, as shows the Equation 2.

3.3 Multi-Objective Optimization Problem

According to Deb [4], a multi-objective optimization problem (MOOP) can be seen as a search problem that aims to minimize or maximize two (or more) objectives that are usually in conflict. Without loss of generality, a MOOP can be defined as: $\vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \ldots, f_l(\vec{x})]$ where $\vec{x} = [x_1, \ldots, x_n] \in N^n$ is an n-variable decision vector, $\vec{f}$ is the set of objective functions to be minimized or maximized, and l is the number of objectives.

According to this idea, the following definitions are provided: a) a solution x_1 dominates a solution x_2 (denoted by $x_1 \preceq x_2$) if the solution x_1 is not worse than x_2 in all objectives and it is better than x_2 in at least one objective. In MOOPs there is not a single optimal solution, conversely, we can find a set of solutions that have no other solution which dominates them when all objectives are currently considered. Hence, the set of non-dominated solutions is called *Pareto optimal* set, and the evaluations of each non-dominated solution in each objective function are known as the *Pareto front* [4].

4 Non-Dominated sorting Genetic Algorithm (NSGA-II)

NSGA-II is a fast elitist multi-objective evolutionary algorithm proposed by Deb et al. [4]. In NSGA-II the individuals are ordered into non-dominated sets called fronts. A rank based on the number of the front is assigned to each individual. To know how close an individual is to its neighbors, the crowding distance is computed for each individual.

Algorithm 1 NS2BN

```
G=0 {Generation}
Generate a population $P$ of random solutions $\vec{x_i}, \forall i, i = 1, \ldots, POP\_SIZE$
Repair cycles of each $\vec{x}_i\ \forall i, i = 1, \ldots, POP\_SIZE$
Evaluate the fitness functions using the first and the second term of the Eq. 1 of each $\vec{p}_i\ \forall i, i = 1, \ldots, POP\_SIZE$
while $G \leq G_{max}$ do
    Crate offspring population $Q$ using: binary tournament selection, one-point crossover and bit inversion mutation.
    Repair cycles
    Evaluate the fitness functions using the first and the second term of the Eq. 1 of each $\vec{x}_i\ \forall i, i = 1, \ldots, POP\_SIZE$
    Combine parents and offspring population $R = P \cup Q$
    Sort using non-dominated criterio
    Replacement
    $G = G + 1$
end while
```

The selection of parents is performed by using a binary tournament based on the rank and the crowding distance. The selected parents generate offsprings through crossover and mutation operators.

The pseudocode of the proposed approach named Non-dominated Sorting Genetic Algorithm for learning BN (NS2BN) is presented in Algorithm 1.

For the carried out of NS2BN: i) the representation of the individual is adjacency matrix, and ii) a repair operator that replaces values randomly when a cycle is identified.

5 Experiments and Results

5.1 Experimental Setup

This section presents the experimental setup. Firstly, we proposed four golden-standard networks with 6-nodes and the following characteristics: i) two of them with 8 arcs each one, and ii) two of them with 7 and 9 arcs, respectively; we call them A RDP, B RDP, C LED, and D LED, accordingly. In the four networks, all the random variables are binary, since this does not produce any qualitative impact on results in comparison to non-binary variables [1]. Then, we generate the datasets through these networks in instances of 1000, 5000, and 10000 cases. The first two of these databases were generated using a random probability distribution and the next two were generated with distribution $p = 0.1$ that according to [1] changing the parameters to be high or low tends to produce low-entropy distributions which have more potential for data compression.

Additionally, we include the following datasets: Asia that has 8-nodes and 8 arcs and Car Diagnosis that has 18-nodes and 20 arcs. Both networks were tested using the dataset with 1000, 5000 and 1000 instances.

Ten independent runs were made by each algorithm per database, with $20,000$ evaluations. A single objective Genetic Algorithm [9] (GA) was carried out for comparison propose. The individual representation consists of an adjacency matrix; the fitness function is the crude MDL, as described in the previous subsection (3.2).

In this algorithm, binary tournament parent selection, one-point crossover and bit inversion mutation are employed. The GA finds a single network for each execution, the network with the best MDL is chosen as the "genetic solution", meanwhile, in NS2BN the result of a run is a set of solutions with a variety of accuracy and structural complexity measurements. To choose an overall best solution from the Pareto front is scientifically invalid due to all the solutions are equally desirables, and normally the decision corresponds to a high-level expert knowledge in the modeling field.

In this work, to have a comparison between the multi-objective approach and the single-objective approach, from the accumulated Pareto front of ten executions, the solution nearest to a reference point which is $(0, 0)$ is chosen. To find this solution, all of them were normalized and the Euclidean distances were computed between the reference point and each Pareto solution. The solution with the shortest Euclidean distance is referred to as the "chosen solution" in this work.

The experimentation is presented in two parts 1) the comparison against the *gold-standard* network, the genetic solution and the chosen solution in terms of the Kullback-Leibler divergence (KLD) computed as the log_2 of the ratio of gold-standard network/chosen solution or genetic solution, according to the case, and 2) the analysis of the plots of the accumulated Pareto fronts.

The parameters setting employed by NS2BN and the GA were tuning empirically. The parameters are the follows: $POP\_ZIZE = 100$, $G_{max} = 200$, $C = 0.9$ and $M = 0.3$.

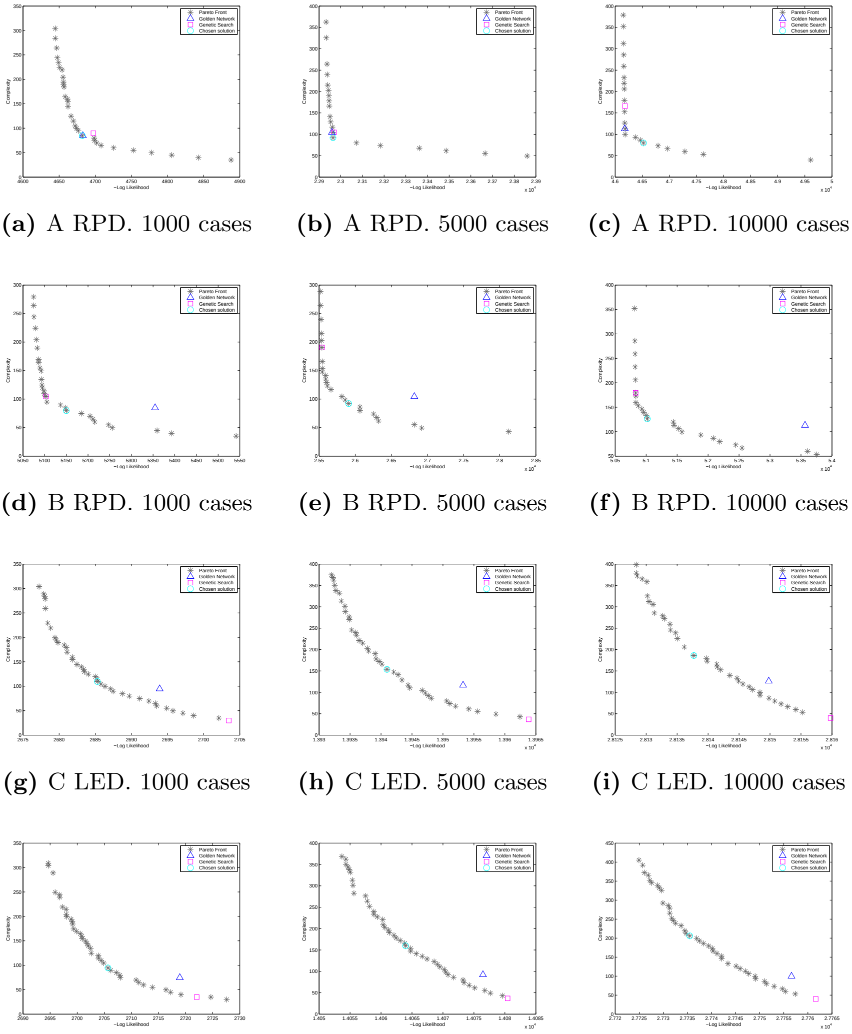
5.2 Results

Table 1 shows the results of the computation of the KLD. According to such a test, there were in ten databases significant differences in favor of the chosen solution that means that the chosen solution is closest to the gold-standard network concerning the subjacent distribution.

Since the genetic algorithm is searching for the minimum value of MDL, the genetic solutions show a minor MDL in sixteen databases. However, one of the objectives is punished in those results. The Figures 1d to 1f show how the genetic solution tends to choose solutions with a smaller log-likelihood but more complex, a similar situation occurs in the Figures 1g to 1i where the genetic algorithm chooses solutions less complex but with a worst log-likelihood value.

Regarding the sample size, Grünwald [6] points that crude MDL does not work well when the sample size is small or moderate and Hastie et al. [7] point out that a metric like crude MDL, in a finite sample, tends to select models less complex. However, these results agree with Grünwald and in contrast to Hastie's et al. our work, show a bias when the sample size is greater in the Genetic Solution, which is used a weighted sum, since this solution tends to select a more complex model (see Figures 1b, 1c, 1e, 1f, 2b, 2c, 2e and 2f).

The experiments generated by a low-entropy distribution show, as was pointed by Cruz-Ramírez et al. [3] that the presence of noise rate affects the behavior



(a) A RPD. 1000 cases **(b)** A RPD. 5000 cases **(c)** A RPD. 10000 cases

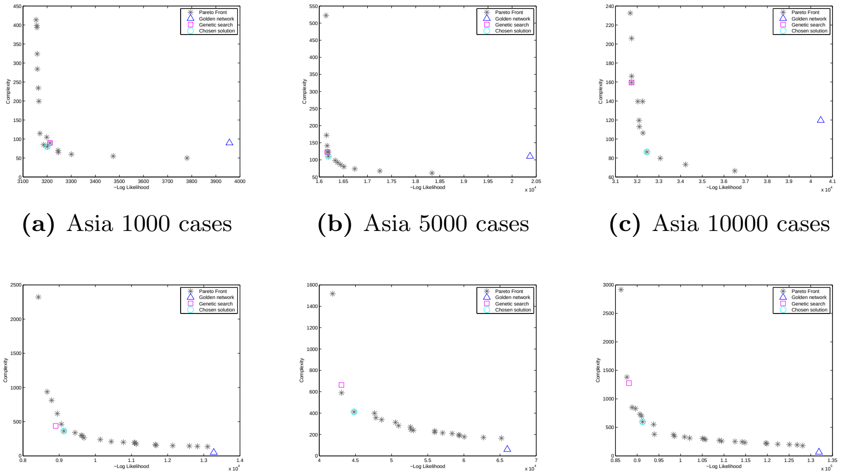
(d) B RPD. 1000 cases **(e)** B RPD. 5000 cases **(f)** B RPD. 10000 cases

(g) C LED. 1000 cases **(h)** C LED. 5000 cases **(i)** C LED. 10000 cases

(j) D LED. 1000 cases **(k)** D LED. 5000 cases **(l)** D LED. 10000 cases

Fig. 1. Accumulated Pareto front of the twelve first databases with 6-nodes with random probability distribution (RPD) and low-entropy probability distribution (LED). Gray stars - the accumulated front obtained by five runs of NSGA-II. Blue triangle - the golden-standard network. Pink square - the genetic solution and then green circle - the chosen solution from the Pareto front.

of MLD which tends to prefer the less complex models, even a network with no arcs.



(a) Asia 1000 cases **(b)** Asia 5000 cases **(c)** Asia 10000 cases

(d) Car diagnosis 1000 cases **(e)** Car diagnosis 5000 cases **(f)** Car diagnosis 10000 cases

Fig. 2. Accumulated Pareto front of the well-known benchmark databases with the different number of cases. Gray stars - the accumulated front obtained by five runs of NSGA-II. Blue triangle - the golden-standard network. Pink square - the genetic solution and the green circle - the chosen solution from the Pareto front.

However, these results show, independent of the sample size, solutions with better values in both terms (see Figures 1g to 1l).

6 Conclusion and Future Work

In this paper, a novel evolutionary bi-objective optimization approach for model selection of BN was presented.

The accuracy and the complexity, which are related to bias and variance respectively, were adopted as the objectives to be optimized to obtain models with an acceptable generalization performance. A set of trade-off solutions was obtained per database. A solution nearest to the origin was chosen as a competitive solution with a suitable trade-off between the objectives. This chosen solution was compared with a single-objective solution. The chosen solution achieved competitive results, especially in the complexity. It is important to note, that one of the main advantages of this approach is the set of trade-off solutions and that the selection of a model can be a high-level decision and must be performed by a domain expert of the modeling phenomenon. Additional advantages are that the proposed method can be applied to a database from different domains and can be extended to other models. As future work, different

Table 1. Kullback-Leibler divergence computed between the gold-standard network with the genetic search solution and the gold-standard network with the chosen solution of the Pareto front. Values in **boldface** mean the best value found.

Golden-network	Genetic search	Chosen solution
A RPD. 1000 cases	0.006256036	**0.000412874**
A RPD. 5000 cases	0.000735484	**0.000166667**
A RPD. 10000 cases	**0.000622825**	0.010558429
B RPD. 1000 cases	**0.5008542**	0.512832286
B RPD. 5000 cases	**0.50817743**	0.527715617
B RPD. 10000 cases	**0.501635069**	0.506660672
C LED. 1000 cases	0.006859061	**0.000558415**
C LED. 5000 cases	0.001254388	**8.84927E-06**
C LED. 10000 cases	0.000630321	**0.000231126**
D LED. 1000 cases	0.005505678	**0.001674059**
D LED. 5000 cases	0.001196043	**0.0007695**
D LED. 10000 cases	0.000561088	**0.000529102**
Asia 1000 cases	0.184669176	**0.183903387**
Asia 5000 cases	0.279944777	**0.277977466**
Asia 10000 cases	0.272191288	**0.262362486**
Car diagnosis 1000 cases	**0.161505741**	0.278079726
Car diagnosis 5000 cases	**0.160725004**	0.192815203
Car diagnosis 10000 cases	**0.200548739**	0.223971025

methods can be used to evaluate accuracy and complexity. Also, alternatives to reduce the computational cost of the algorithm can be included.

References

1. Allen, T.V., Greiner, R.: Model Selection Criteria for Learning Belief Nets: An Empirical Comparison. In: Int Conf Mach Learn. pp. 1047–1054 (2000)
2. Blanco, R., Inza, I., Larrañaga, P.: Learning Bayesian networks in the space of structures by estimation of distribution algorithms. International Journal of Intelligent Systems 18(2), 205–220 (2003)
3. Cruz-Ramírez, N., Acosta-Mesa, H.G.G., Mezura-Montes, E., Guerra-Hernández, A., Hoyos-Rivera, G.d.J., Barrientos-Martínez, R.E.E., Gutiérrez-Fragoso, K., Nava-Fernández, L.A.A., González-Gaspar, P., Novoa-del Toro, E.M.M., Aguilera-Rueda, V.J., Ameca-Alducin, M.Y.: How good is crude MDL for solving the bias-variance dilemma? An empirical investigation based on Bayesian networks. PloS one 9(3) (2014)
4. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast Elitist Multi-Objective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6, 182–197 (2000)
5. Gräning, L., Jin, Y., Sendhoff, B.: Generalization improvement in multi-objective learning. In: International Joint Conference on Neural Networks. p. 9893–9900. IEEE Press (2006)
6. Grünwald, P.D.: The Minimum Description Length Principle (Adaptive Computation and Machine Learning). The MIT Press (2007)

7. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer Series in Statistics, Springer New York Inc., New York, NY, USA (2001)
8. Heckerman, D.: A Tutorial on Learning with Bayesian Networks. In: Jordan, M.I. (ed.) Learning in Graphical Models, pp. 301–354. MIT Press, Cambridge, MA, USA (1999)
9. Holland, J.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, MI, USA (1975)
10. Li, G., Xing, L., Chen, Y.: A New BN Structure Learning Mechanism Based on Decomposability of Scoring Functions, pp. 212–224. Springer Berlin Heidelberg, Berlin, Heidelberg (2015)
11. Rosales, A., Escalante, H.J., C. A. Reyes, J.A.G., Coello, C.A.: Bias and Variance Optimization for SVMs Model Selection. In: Proceedings of the Twenty-Sixth International Florida Artificial Intelligence Research Society Conference. pp. 136–141. St. Pete Beach, Florida, USA (2013)
12. Ross, B.J., Zuviria, E.: Evolving dynamic Bayesian networks with Multi-objective genetic algorithms. Applied Intelligence 26(1), 13–23 (Feb 2007)
13. Xiao-Lin, L., H., X.D., Chuan-Ming, C.: A method for learning Bayesian networks by using immune binary particle swarm optimization. In: Slezak, D., Kim, T.h., Zhang, Y., Ma, J., Chung, K.i. (eds.) Database Theory and Application, Communications in Computer and Information Science, vol. 64, pp. 115–121. Springer Berlin Heidelberg (2009)

ISSN 1870-4069

Digit-Based Speaker Verification in Spanish Using Hidden Markov Models

Juan Carlos Atenco Vazquez[1], Juan Carlos Moreno Rodriguez[1],
Rene Arechiga Martinez[2], Juan Manuel Ramirez Cortes[1],
Pilar Gomez Gil[1], Rigoberto Fonseca Delgado[3]

[1] National Institute of Astrophysics, Optics and Electronics,
Mexico

[2] New Mexico Tech., Electrical Engineering Department,
USA

[3] Yachai Tech.,
Ecuador

{atencovaz, jmram, jmram, pgomez}@inaoep.mx,
rene.arechiga@nmt.edu, rfonseca@yachaytech.edu.ec

Abstract. In this paper we propose a digit-based text-dependent speaker verification system (SVS) in Spanish. The system uses word level Hidden Markov Models (HMM) as classifiers and Frequency Cepstral Coefficients (MFCC) with Cepstral Mean Subtraction as features. The system was developed considering a gender independent Universal Background Model (UBM) within an HMM-UBM framework. The training data set was pooled with both genders equally represented for the UBM. A Target Speaker Model (TSM) was generated using Maximum A Posteriori (MAP) to adapt the UBM's parameters to the acoustic characteristics of the target speakers. Every target speaker (TS) has a 4 digits password. Robustness of the system was tested using adaptive noise cancellation as a speech enhancement scheme; the speech signals were corrupted with additive white Gaussian noise (AWGN) at different values of signal to noise ratio. Generation of a digit-database, which we have named as BIOMEX-DB, is described. The main contribution of this work is a robust SVS in Spanish language tested with 4-digits passwords, which can be easily adapted to different lengths or text-prompted mode. Obtained results showed an equal error rate (EER) in the range of 1.0567-1.4465 % when 50 subjects in the database were considered.

Keywords: speaker verification, HMM, voice biometrics, universal background model, speaker adaptation, voice database.

1 Introduction

Nowadays there is an increasing number of applications that require verification of a user's identity, such as access to facilities, internet applications, or bank services. In that sense, biometric systems are widely used as a viable solution for most security problems.

A biometric modality with a good balance of properties within the universe of desired conditions is the use of voice signals, known as speaker verification, which constitutes an active research area. Many approaches focused on attacking the inherent problems of this biometric modality have been recently proposed [11]. In a speaker verification task there are two relevant modalities: text dependent and text independent. In the first scheme, the words spoken by a speaker for recognition are limited to a specific vocabulary, in the second approximation there is no limitation in the words that can be uttered [17]. There is an increasing interest to capitalize on the advantages of text-dependent systems while allowing for the flexibility of the text-independent domain [4]. A review of text dependent modality can be consulted in [6].

HMM and Gaussian Mixed Models (GMM) have been widely used in speaker recognition approaches. In recent years these techniques have been used to obtain statistics for a new representation known as i-vector feature extraction [19, 20]. In [16] HMM's are used for data segmentation and the resulting statistics are incorporated in a system based on Joint Factor Analysis (JFA). In other works these models have been used as classifiers aiming to capture the temporal information of voice signals and improve accuracy through different types of UBMs [14, 13]. In [7] a performance comparison is made between HMM, GMM and i-vector showing competent results in different scenarios. Another important aspect of speaker recognition is the choice of a feature extraction technique that improves accuracy and captures the information of the speech signal. Several techniques for representing spectral features, such as Linear predictive cepstral coefficients (LPCC), Perceptual Linear Predictive (PLP) and Mel Frequency Cepstral Coefficients (MFCC) have been used for different speech processing tasks, mainly speech and speaker recognition. Reference [8] presents details on the use of LPC and MFCC with several classifiers for speech recognition. In [3] the performance of a HMM based speech recognition system trained with noisy speech samples using several spectral feature extraction techniques is compared. In the field of speaker recognition, spectral features have a widespread use, in [1] spectral features are used to train several classification models for a text independent identification task.

In this work, a HMM-based speaker verification system using a single 4 digits password in Spanish is presented. Verification is done through the matching of the spectral characteristics of the voice using Universal Background Models (UBM) and Target Speaker Models (TSM), obtaining a score based on Log Likelihood Ratio (LLR), testing speaker correct (SC) and Impostor correct (IC) for the assessment of identity verification. The MFCC feature extraction technique was used together with the Delta (first derivative) and Delta-Delta (second derivative) coefficients. Cepstral Mean Normalization (CMN) technique was applied to compensate for spectral effects caused by the recording channel. Each HMM is trained at word level representing a digit, so there is a set of UBMs composed of the 10 models plus 1 silence model, and a target-speaker set using the same scheme. Robustness of the system in several noise conditions was tested using an adaptive LMS filter approach.

Results are analyzed using ROC (Receiver Operating Characteristic) curves, presenting an equal error rate (EER) in the range of 1.0567-1.4465 % with a population of 50 subjects. The rest of the paper is organized as follows: section 2 describes theoretical concepts; section 3 presents a description of the BIOMEX-DB database which was generated for this experiment. In section 4, experimental development and performed testing on the SVS are explained. Tests results are described in section 5 and concluding remarks are presented in section 6.

2 Speaker Verification System

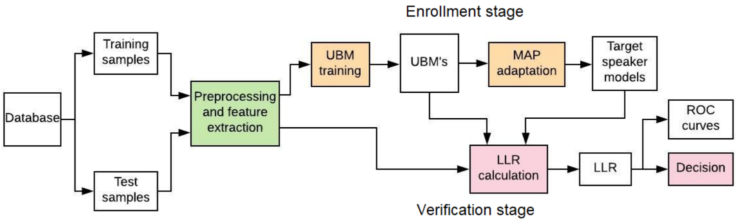


Fig. 1. Speaker verification process.

Figure 1 shows the block diagram of the SVS indicating the enrollment and verification stages. Feature extraction consists of the generation of MFCC [5], incorporating Delta and Delta-Delta in the feature vector. The Enrollment stage consists of training of HMM-based UBMs using the Baum-Welch algorithm for parameter estimation[18], and after a MAP adaptation TSMs are generated. In the verification stage, LLR is calculated to qualify the match between the features of a test speech sample and both UBM and TSM [12]. Log likelihood ratio is defined in Equation 1.

$$\Lambda(\mathbf{X}) = \log p(\mathbf{X}|\lambda_S) - \log p(\mathbf{X}|\lambda_{UBM}), \tag{1}$$

where $p(\boldsymbol{X}|\lambda_S)$ is the likelihood that the feature vector $\mathbf{X}$ has been generated by the speaker model λ_S, while $p(\boldsymbol{X}|\lambda_{UBM})$ is the likelihood that $\mathbf{X}$ was generated by the UBM. Verification is then carried out by comparing the LLR with a predetermined threshold and based on that comparison the SVS accepts or rejects the claimed identity of a test speaker.

2.1 Hidden Markov Models (HMM)

Figure 2 shows a block diagram of the HMM structure used in this work. An HMM represents a doubly embedded stochastic dynamical process through a set

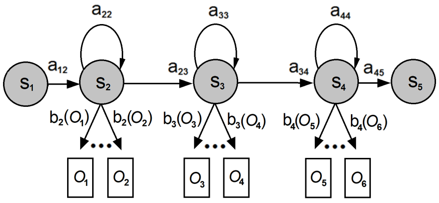


Fig. 2. Hidden Markov Model (HMM) representation.

of states $S : \{S_1, S_1, ..., S_N\}$, a set of possible observations $O : \{O_1, O_1, ..., O_M\}$, a transition probability matrix $A = [A_{ij}]$, and an output probability distribution $B = [B_{ij}]$. The stochastic nature of HMMs is contained in the transition of states defined in the transition matrix A, and in generation of the observations B, being both probabilistic events [10]. HMM have been widely used in speech processing due to the ability to capture temporal information of the speech, taking into account its non-stationary nature. In this work, the HTK (Hidden Markov Model Toolkit) software tools [18] were used to build the UBMs and TSMs conforming the SVS. Parameter estimation and frame alignment in HMM were implemented using Baum-Welch and Viterbi algorithms respectively [18].

2.2 Speaker Adaptation

A TSM contains the acoustic characteristics of a TS. The process followed in this work was the adaptation of the parameters of previously trained UBMs with the target speaker's speech information. MAP parameter adaptation is defined according to equation 2:

$$\widehat{\mathbf{u}}_{jm} = \frac{\mathcal{N}_{jm}}{\mathcal{N}_{jm} + \tau}\overline{\mathbf{u}}_{jm} + \frac{\tau}{\mathcal{N}_{jm} + \tau}\mathbf{u}_{jm}, \tag{2}$$

where $\widehat{\mathbf{u}}_{jm}$ is the adapted mean vector, $\overline{\mathbf{u}}_{jm}$ is the mean vector of the adaptation data and $\mathbf{u}_{jm}$ is the mean vector of the speaker independent model. $\mathcal{N}_{jm}$ is the likelihood that the adaptation data was generated by the Gaussian component m in the j state, while τ is the weight of the adaptation data. In [2] it is shown that adapting other parameters, mean not included, decreases the accuracy of an SVS, furthermore, it is also shown that a very high value of τ can also decrease the accuracy. In this work, the TSMs were generated by adapting only the means, with $\tau = 12$ established by experimentation.

3 BIOMEX-DB Database Generation

The experiments described in this work were carried out with a database created specifically for biometric purposes, focused in digit-based text dependent speaker verification in Spanish language. The speech signals were obtained from 51 volunteers, 26 men and 25 women to ensure a balanced representation of both genders and voice variability. Table 1 shows the demographic distribution of volunteers within the database.

Table 1. Demographic distribution of subjects in the database.

	Gender	
Age	Male	Female
Less than 21	3	2
21 - 30	18	14
31 - 40	2	6
41 - 50	1	1
51 - 60	0	2
Older than 60	2	0

The speech database consists of audio files containing strings of digits randomly ordered with a short segment of silence between utterances. The speech database is divided into two parts: The first part consists of 10 strings of 10 digits each, giving a total number of 5100 digits pronounced by 51 subjects. The second part consists of 10 strings of 4 digits each. Each string is considered a 4 digits password assigned to a specific speaker. Similarly, each digit is pronounced once per string. All speech samples of each speaker were recorded consecutively in 15-20 minutes sessions. The recording was carried out with a microphone Sennheiser model MD 421-II connected to a deskstop computer through a Yamaha amplifier model MG06X. A MATLAB script was used to display the digits to be pronounced and to generate the transcripts of each audio file. The audio signals were recorded with a sampling frequency of 16 KHz, a resolution of 16 bits per sample, and stored in wav format.

4 Experimental Setup

The evaluation was conducted using two gender balanced population sizes of 40 and 50 speakers. These population sizes were chosen according to the maximum amount of speakers available in the database and to evaluate the impact of different population sizes in the results. With a population of 40 speakers, 24 were employed to train the UBM, 8 target speakers and 8 impostors. With 50 speakers, 30 were employed to train the UBM, 10 target speakers and 10 impostors, using gender balance in all cases. The MFCC features were extracted using signal framing of 25 ms Hamming windows with an overlapping of 10 ms. The number of states of the HMM was the same in every UBM and TSM models, with one Gaussian component per state. Three cases were analyzed according to the number of states in the HMM structure: 5, 8 and 12 states. Testing was carried out using a jack-knife-like iterative scheme [9].

Robustness of the system in the presence of noise was tested through a series of experiments in which the speech signals were corrupted with additive white

Gaussian noise at different signal-to-noise values. For that purpose, a normalized least mean square (NLMS) filter was incorporated [15].

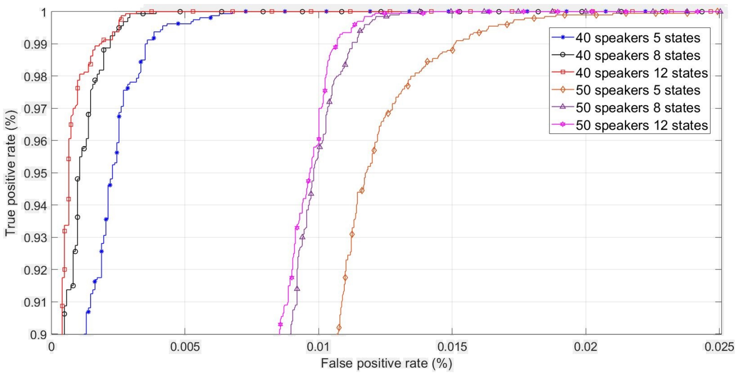


Fig. 3. Speaker Verification results; ROC curves.

5 Results and Discussion

Fig. 3 shows the ROC curves obtained with two population cases, 40 and 50 subjects, and three different HMM models with 5, 8 and 12 states. This figure has been zoomed to the upper left corner in order to highlight details. These results are concentrated in table 2 showing the evaluation parameters EER and Area under the Curve (AUC).

The obtained EER is located in the ranges 0.2516-0.4353 % and 1.0567-1.4465 % for the cases of 40 and 50 speakers, respectively. Results obtained in the proposed work are comparable with those presented in recent works, although differences in population sizes, corpora, training conditions and language do not allow a direct evaluation. In [13] the authors report HMM-UBM and GMM-UBM systems with an EER ranging from 7.12 to 0.79 %.

Reference [7] presents different HMM and GMM systems with an EER in the range 9.94-0.59 %. In [14], HMM-UBM and GMM-UBM systems with an EER of 5.56-0.009 % are reported. Results show that the more states the HMMs have the better the verification results are, since there are more states the acoustic modeling improves, however 12 states HMMs don't show significant improvements in comparison with 8 states because there isn't enough training data to further improve the results.

A trade off between the computational cost for both training and testing and the number of states must be take into consideration, more states increases the computational cost, therefore the 8 states HMMs were considered good choices. It is evident that the best results come from the 40 subjects population, it is well documented that performance decreases with big populations.

Table 2. Speaker verification results; EER and AUC.

HMM states	Population			
	P=40		P=50	
N	EER %	AUC	EER %	AUC
5	0.4353	0.9995	1.4465	0.9943
8	0.2992	0.9998	1.1378	0.9951
12	0.2516	0.9999	1.0567	0.9952

Table 3 shows the values of EER and AUC obtained when the speech signals are corrupted with AWGN at the following SNR values: -10 dB, -5 DB, and 0 dB, using a HMM structure of 8 states. As expected, the system performance decreases compared to the noiseless case, however, the obtained EER lies in the range of 0.7467-0.8746 % which it still a satisfactory system behavior.

Table 3. EER and AUC results with signals corrupted by AWGN; N=8 HMM states, P=40 subjects.

SNR (dB)	EER (%)	AUC
Noiseless	0.2992	0.9999
-10	0.7467	0.9997
-5	0.8107	0.9997
0	0.8746	0.9997

6 Conclusions

A HMM-UBM based speaker verification system using a four-digit password pronounced in Spanish language, has been presented. Several experiments were carried out with different number of emitting states and two population sizes: 40 and 50 speakers. The more emitting states the better obtained accuracy, however, the training time increases and more computational resources are needed.

The experiments showed that the best results in terms of EER, AUC, and computational cost considerations were obtained with a HMM structure of 8 states. This structure was further tested in noise conditions adding white Gaussian noise at several SNR values, showing good noise tolerance. Results showed a degradation on EER from 0.2992 % without noise to 0.8746 % with a SNR=0 dB which corresponds to a noise power level in the same magnitude of the voice signal. In practical applications is not possible to operate in noiseless environments, so adaptive filtering is an affordable alternative to improve the robustness of a HMM-UBM based SVS. Additional experiments with real and typical noise conditions are currently in progress. In conclusion, the system showed high accuracy to discriminate between correct target speakers and impostors.

References

1. Charan, R., Manisha, A., Karthik, R., Kumar, M.R.: A text-independent speaker verification model: A comparative analysis. In: 2017 International Conference on Intelligent Computing and Control (I2C2). pp. 1–6. IEEE (2017)
2. Gauvain, J., Barras, C.: Feature and score normalization for speaker verification of cellular data. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, Proceedings (ICASSP'03). vol. 2, pp. II–49. IEEE (2003)
3. Këpuska, V.Z., Elharati, H.A.: Robust speech recognition system using conventional and hybrid features of mfcc, lpcc, plp, rasta-plp and hidden markov model classifier in noisy conditions. Journal of Computer and Communications 3(06), 1 (2015)
4. Kofi, B.: Speaker recognition in the text-independent domain using keyword hidden markov models. Masters Report, University of California at Berkeley (2005)
5. Kopparapu, S.K., Laxminarayana, M.: Choice of mel filter bank in computing mfcc of a resampled speech. In: 10th International Conference on Information Science, Signal Processing and their Applications (ISSPA 2010). pp. 121–124. IEEE (2010)
6. Larcher, A., Lee, K.A., Ma, B., Li, H.: Text-dependent speaker verification: Classifiers, databases and rsr2015. Speech Communication 60, 56–77 (2014)
7. Liu, Y., He, L., Tian, Y., Chen, Z., Liu, J., Johnson, M.T.: Comparison of multiple features and modeling methods for text-dependent speaker verification. In: 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). pp. 629–636. IEEE (2017)
8. Madan, A., Gupta, D.: Speech feature extraction and classification: A comparative review. International Journal of computer applications 90(9) (2014)
9. Martın-Donas, J.M., López-Espejo, I., González-Lao, C.R., Gallardo-Jiménez, D., Gomez, A.M., Pérez-Córdoba, J.L., Sánchez, V., Morales-Cordovilla, J.A., Peinado, A.M.: Secuvoice: A spanish speech corpus for secure applications with smartphones (2016)
10. Nguyen, L.: Tutorial on hidden markov model. Applied and Computational Mathematics 6(4-1), 16–38 (2017)
11. Poddar, A., Sahidullah, M., Saha, G.: Speaker verification with short utterances: a review of challenges, trends and opportunities. IET Biometrics 7(2), 91–101 (2017)
12. Reynolds, D.A., Quatieri, T.F., Dunn, R.B.: Speaker verification using adapted gaussian mixture models. Digital signal processing 10(1-3), 19–41 (2000)
13. Sarkar, A.K., Tan, Z.H.: Text dependent speaker verification using un-supervised hmm-ubm and temporal gmm-ubm. In: interspeech. pp. 425–429 (2016)
14. Sarkar, A.K., Tan, Z.H.: Incorporating pass-phrase dependent background models for text-dependent speaker verification. Computer Speech & Language 47, 259–271 (2018)
15. Sil, R., Bharath, K., Karthik, R., Kumar, M.R.: Nlms-loess algorithm for adaptive noise cancelation. In: Microelectronics, Electromagnetics and Telecommunications, pp. 65–74. Springer (2019)
16. Stafylakis, T., Alam, M.J., Kenny, P.: Text-dependent speaker recognition with random digit strings. IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP) 24(7), 1194–1203 (2016)
17. Tirumala, S.S., Shahamiri, S.R., Garhwal, A.S., Wang, R.: Speaker identification features extraction methods: A systematic review. Expert Systems with Applications 90, 250–271 (2017)

18. Young, S., Evermann, G., Gales, M., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., Woodland, P.: The htk book version 3.4 manual. Cambridge University Engineering Department, Cambridge, UK (2006)
19. Zeinali, H., Sameti, H., Burget, L., Cernockỳ, J., Maghsoodi, N., Matejka, P.: i-vector/hmm based text-dependent speaker verification system for reddots challenge. In: InterSpeech. pp. 440–444 (2016)
20. Zeinali, H., Sameti, H., Burget, L., et al.: Text-dependent speaker verification based on i-vectors, neural networks and hidden markov models. Computer Speech & Language 46, 53–71 (2017)

ISSN 1870-4069

Pareto Explorer for Solving Real World Applications

Oliver Cuate[1], Oliver Schütze[1,2]

[1] CINVESTAV, Computer Science Department,
Mexico

[2] Universidad Autónoma Metropolitana Cuajimalpa,
Mexico

schuetze@cs.cinvestav.mx, ocuate@computacion.cs.cinvestav.mx

Abstract. An important characteristic of the Multi-objective Optimization Problems (MOPs) is that their solution sets typically form a $(k-1)$-dimensional object where k is the number of objectives involved in the MOP. Thus, it is only possible to approximate the entire set of interest for a relatively few numbers of objectives (say, $k = 3$ or 4). In this work, we address the numerical treatment of MOPs with more than four objectives which are termed as Many Objective Optimization Problems (MaOPs). Such problems have recently caught the interest in the industry as the decision-making processes are getting more and more complex. The recently proposed Pareto Explorer (PE) method raises as a solution for the MaOPs, it is conceived as a global/local exploration tool which consists of two principal phases: obtaining a global optimal solution for a given MaOP, and the local exploration of optimal solutions based on the preferences of a decision-maker. In this work, we demonstrate the effectiveness of PE for solving real-world applications.

Keywords: many objective optimization, interactive method, decision making, continuation method.

1 Introduction

In many applications, several objectives have to be optimized concurrently leading to a *multi-objective optimization problem* (MOP). Due to the increasing complexity of practical problems, decision-making processes are getting more and more sophisticated. Motivated by the advances in the design of algorithms for the numerical treatment of MOPs [13] with few objectives and their huge success in applications, there is a recent trend to include more objectives into the optimization process. Due to this reason, MOPs with more than four objectives are often termed *many objective problems* (MaOPs) in the literature as they require a different numerical treatment than problems with two to four objectives.

However, there exist real-world problems where the decision-maker (DM) has some knowledge about the problem or she/he wants to obtain optimal solutions with specific characteristics instead of a vast set of alternatives. Reference point methods are useful for this scenario, where the idea is to get the closest solution to a given vector, usually infeasible, which is a guess of the DM. This kind of methods, where the DM has active participation in the solution process, are called interactive methods, and they differ from each other according to what type of information they ask the DM [10].

We can find different alternatives which consider only one reference point to get a solution, some of them include reference point method [17], light beam search [8], GUESS [3], and even the r-NSGA-II [4], which can work with a set of points. On the other hand, the learning-oriented methods (a different class of interactive methods), exploit the preferences of the DM to direct the search, and reduce the number of solutions to consider. Such methods are useful when the set of optimal solutions is huge, for example, for many-objective optimization problems. A wide variety of these interactive methods have been developed [2], for example, Pareto Navigator [5], NIMBUS [12], and Nautilus [11].

Recently, continuation methods have been used to solve the multiobjective optimization problem. These methods have the advantage that they move through the Pareto front. To achieve this, we need an initial optimal solution, starting from this point we compute a predictor, which is a movement according to specific criteria, and then with a corrector we obtain a new optimal solution. The change both in the predictor as in the corrector, gives rise to different methods as they are Hillermeier method [7], Pareto Tracer [9], and Zigzag [16].

The method used this paper, called Pareto Explorer, is a continuation method that was recently proposed and which takes into account the preferences of the DM to calculate the predictor. It is in spirit an interactive method and even more. Here we show how to use PE to solve real world applications.

2 Background

A continuous multi-objective optimization problems (MOP) is mathematically expressed as:

$$\min_{x \in D} F(x) = [f_1(x), \ldots, f_k(x)]^T, \tag{1}$$

where $D \subset \mathbb{R}^n$ is the domain and $F : D \subset \mathbb{R}^n \to \mathbb{R}^k$ is called the objective function, where k is the number of objectives and n is the number of variables.

The optimality of an MOP is defined by the concept of *strict dominance.* Let $v, w \in \mathbb{R}^k$, the vector v is *less than* w ($v <_p w$), if $v_i < w_i$ for all $i \in \{1, \ldots, k\}$; the relation $\leq_p$ is defined analogously. A vector $y \in D$ is *dominated* by a vector $x \in D$ ($x \prec y$) with respect to (1) if $F(x) \leq_p F(y)$ and $F(x) \neq F(y)$, else y is called non-dominated by x. A point $x^* \in \mathbb{R}^n$ is Pareto optimal to (1) if there is no $y \in D$ that dominates x. The set of all the Pareto optimal points P_D is called the Pareto set and its image $F(P_D)$ is called the Pareto front. Typically, i.e., under certain mild smoothness assumption on the model, both Pareto set and front form at least locally $(k-1)$-dimensional objects.

2.1 Pareto Tracer

Here we briefly state the core elements of PT for unconstrained problems, for details including constraint handling we refer to [9].

In [7] we find a continuation method for the MOPs context by considering $\hat{F} : \mathbb{R}^{n+k} \to \mathbb{R}^{n+1}$:

$$\hat{F}(x,\alpha) = \begin{pmatrix} \sum_{i=1}^{k} \alpha_i \nabla f_i(x) \\ \sum_{i=1}^{k} \alpha_i - 1 \end{pmatrix} = 0. \tag{2}$$

The set of KKT points of (1) is contained in the zero set of $\hat{F}$ which motivates the continuation along $\hat{F}^{-1}(0)$.

The idea of Pareto Tracer [9] is to separate the decision and weight space:

$$\hat{F}'(x,\alpha)\begin{pmatrix} \nu \\ \mu \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{k} \alpha_i \nabla^2 f_i(x) & \nabla f_1(x) & \dots & \nabla f_k(x) \\ 0 & 1 & \dots & 1 \end{pmatrix}\begin{pmatrix} \nu \\ \mu \end{pmatrix}. \tag{3}$$

By the second equation of (3) we have that $\sum_{i=1}^{k} \mu_i = 0$, and it is possible to find a relationship between ν and μ, i.e., a relationship between the objective space and the variable space:

$$\nu_\mu = -W_\alpha^{-1} J^T \mu, \tag{4}$$

where $W_\alpha := \sum_{i=1}^{k} \alpha_i \nabla^2 f_i(x) \in \mathbb{R}^n$ and $J = J(x) = (\nabla f_1(x)^T, \dots, \nabla f_k(x)^T)^T \in \mathbb{R}^{k\times n}$. Finally, given a direction $d \in \mathbb{R}^k$ in objective space such that $J\nu_\mu = d$, this vector ν_μ can be obtained with the vector μ_d that solves:

$$\begin{pmatrix} -JW_\alpha^{-1}J^T \\ 1 \dots 1 \end{pmatrix} \mu_d = \begin{pmatrix} d \\ 0 \end{pmatrix}. \tag{5}$$

If the rank of J is $k-1$, we can compute the set of tangent vectors via a QR factorization of α, i.e. $\alpha = QR$. Let Q_2 denote the matrix formed by the last $k-1$ columns vectors of Q, this matrix is an orthonormal basis of the linearized Pareto front at $F(x)$.

3 Pareto Explorer

In case the number k of objective is too high, it is not possible to compute a suitable finite size approximation of the entire solution set any more. Instead, the Pareto Explorer [14] aims to find a solution in cooperation with the DM in two steps:

Step 1 Compute a solution x_0 of the MaOP.
Step 2 Explore the Pareto landscape around x_0 via performing movements into user specified directions.

Step 1 can be performed via a global heuristic such as an evolutionary reference point method. For Step 2, the above described PT has been adapted in [14] that allows to perform best fit movement along the Pareto set/front in directions defined in decision, objective, and weight space. The key for this is the fact that the tangent spaces of both the Pareto set at x as well as the Pareto front at $F(x)$ can be computed for every regular solution x which follows by the above discussion (computation of the predictor for PT). By doing so, Step 2 allows for a fine-tuning of the initial solution x_0 from Step 1.

Let $d_k \in \mathbb{R}^k$ be a given direction, if x is a solution of (1) with its corresponding α vector, then the best direction to move the point $F(x)$ on the Pareto front, according d_k, is given by the orthogonal projection of d_k on the linearization of the Pareto front at the point $F(x)$ i.e:

$$d = Q_2 Q_2^T d_k, \tag{6}$$

where Q_2 are the last $k-1$ columns of the QR factorization of α. We consider the normalization of d, i.e. $d = d/\|d\|$ and we can now compute the desired vector $\nu_d \in \mathbb{R}^n$ such that $J\nu_d = d$ using (4) and (5).

The normalization of d is useful in order to compute the step length t for the predictor, if we want that, for two consecutive solutions, $\|F(x_i) - F(x_{i+1})\| \approx \tau$, then t is given by:

$$t = \frac{\tau}{\|J\nu_d\|}. \tag{7}$$

The corrector is, as in the case of the Pareto Tracer, given by the Newton method for MOPs [6].

Figure 1 shows a hypothetical example for a best fit movement along the Pareto front from the image $F(x_i)$ at the current iterate x_i. Hereby, $d_y \in \mathbb{R}^k$ denotes the desired direction in objective space specified by the DM, and $d_y^{(i)} \in \mathbb{R}^k$ the direction projected to the linearized Pareto front at $F(x_i)$. The projected direction $d_y^{(i)}$ is used to perform a best fit movement along the Pareto front of the problem.

4 Applications

In this section, we illustrate the efficiency of our method via two real world applications, the industrial laundering and the plastic injection molding (PIM).

4.1 Industrial Laundering

The laundering process is influenced by the four parameters temperature, chemistry (amount of cleaner), time and mechanics (speed of rotation), which is described by Sinners' Circle [15]. These laundries are capable of washing up to 30 tons of laundry per day. Consequently, it is of great interest to increase the efficiency, which is beneficial both for ecological as well as economic reasons.

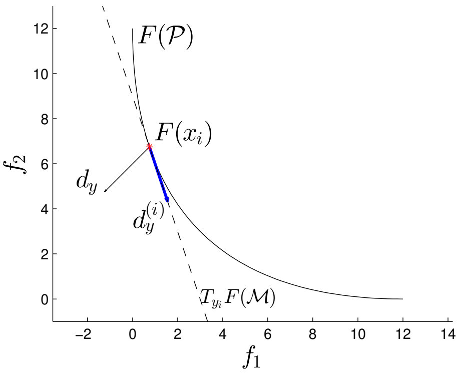


Fig. 1. Best fit direction $d_y^{(i)}$ for a given direction d_y in objective space for the Pareto Explorer.

The model was generated by fitting quadratic Ansatz functions to measurements. The decision variables are the temperature of the water(x_1), the amount of washing detergents (x_2), the washing time (x_3) and the rotating speed of the laundry (x_4). Thirteen of the objectives are related to the cleaning of specific types of contamination, i.e., f_1 tof_{13} represent the effectiveness on: wool grease in cotton, wool grease in polyester, red in cotton, sebum in cotton, sebum in polyester, curry in cotton, motor oil in cotton, petroleum in cotton, blood in cotton, egg in cotton, starch in cotton, cocoa, and vegetable grease, respectively; while the 14^{th} objective is related to the negative of the cost. All parameters are normalized with the reference point being at $(0, 0, 0, 0)$. The degree of cleaning varies between 0 (no cleaning) and 100 (perfect cleaning). This leads the following model:

$$\begin{aligned} \min_{x\in\mathbb{R}^4} \; & F(x) = [f_1(x), \dots, f_{14}(x)]^T, \\ \text{s.t} \; & -1.5 \le x_i \le 1.5, \; i = 1, 2, 4 \\ & 0 \le x_3 \le 1.5. \end{aligned} \tag{8}$$

We used Pareto Explorer to solve the problem of the washing machine (for more details see [14]). For this approach we define the direction in objective space as $d_k = -e_{14}$, i.e. we want to reduce as much as possible the value of the $14th$ function, which is the cost function. We took as initial point the optimal vector $x_0 = (1.0429, 0.8521, 1.3622, 1.5000)$ and $f_{14}(x_0) = 9.5637$. We obtained such a point, after applying the Newton method at the vector formed by the middle value for each variable.

In order to view more easily the obtained results, we introduce graphs which represent optimal solutions as polygons inscribed in the unit circle. In all cases, the center of the circle depicts the best values for each objective function. Then, the more a solution is far to the center, the best is its value. The first objective function is in the line which goes from the point $(0, 0)$ to the point $(1, 0)$, the rest of the objective functions are set in the counter-clockwise direction.

The result obtained by the Pareto Explorer is in Figure 2a (left), the method did 110 iterations and we show the solution for the initial point, $27th$ iteration, $55th$ iteration, and the final result. We can see that there is a reduction of the value for the function f_{14} until reach the best value. The values of f_{14} and the points considered in Figure 2a are:

- $x_0 = (1.0429, 0.8521, 1.3622, 1.5000)$, $f_{14}(x_0) = 9.5637$,
- $x_{27} = (1.0995, -0.5341, 1.5000, 1.5000)$, $f_{14}(x_{27}) = -4.2412$,
- $x_{55} = (0.7417, -1.2346, 1.5000, 1.5000)$, $f_{14}(x_{55}) = -11.6041$,
- $x_{110} = (-1.4718, -1.5000, 1.5000, 1.5000)$, $f_{14}(x_{110}) = -16.5000$.

4.2 Plastic Injection Molding

The parameters we consider are the melt temperature (T_{melt}), the packing time (t_{pack}), the packing pressure (P_{pack}) and the cooling time (t_{cool}). While the seven objectives are related to the quality and productivity of the PIM process. Cosmetic characteristics are measured by the *warpage* (f_1) in the product, *shrinkage* (f_2) and *sink marks* (f_3). Functional properties are represented by residual stresses such as *Von Mises* (f_4) and *shear* stresses (f_5). Productivity is measured by the *cycle time* (f_6) and *clamping force* (f_7). Commonly, only between two and four of these objectives are considered in other works (for more details see [1]).

As case study we use in this work the design of a particular plastic gear. The model (obtained by a surrogate model) is the following:

$$\begin{aligned} \min_{x\in\mathbb{R}^4} \; & F(x) = [f_1(x), \ldots, f_7(x)]^T, \\ \text{s.t } & 190 \le x_1 \le 230, \\ & 3 \le x_2 \le 5, \\ & 60 \le x_3 \le 100, \\ & 8 \le x_4 \le 14. \end{aligned} \tag{9}$$

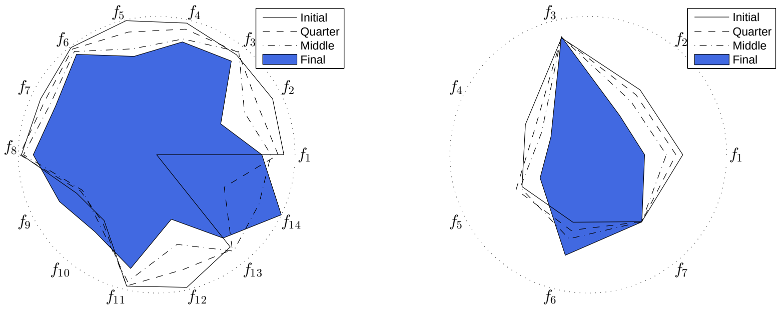
Here we consider the seven described objectives and as initial solution we chose $x_0 = (210.00, 4.00, 80.00, 11.00)^T$. Again, it is the middle point for each variable in the considered range. For the demonstration of Step 2 of the PE, we use the scenario in which we want to minimize the functions f_1, f_5, and f_6 at the same time, i.e., the direction is $d_y = (-1, 0, 0, 0, -1, -1, 0)^T$ with $\tau = 0.01$.

We can see in Figure 2b that the functions f_1 and f_6 are directly in conflict, while f_5 the value depends of both functions. At the end of the optimization process, we obtain the best value for f_6 and the worst value for f_1; for the case of f_5 the initial and the final values are similar, but along the steps such value has a lot of variation. Notice that, the result for this scenario is almost the same than the previous one.

As it can be seen, the movement has been performed according to the desired direction. We have presented here the entire path of solutions, however, in a real decision-making process, the DM can of course chose at any time either to accept a computed candidate solution, or to change the direction in which the steering has to be performed.

Table 1. Model values (F_M) against the simulated values F_S for the PIM.

Initial Configuration							
x_0	210.0000	4.0000	80.0000	11.0000			
$F_S(x_0)$	**0.2016**	5.6565	9.7470	0.0717	**0.8690**	20.1000	11.9460
$F_M(x_0)$	**0.2040**	5.7271	9.7329	0.0713	**0.8774**	20.1000	11.8221
Final Configuration							
x_{212}	213.3452	3.3421	60.0000	9.6950			
$F_S(x_{212})$	**0.2437**	6.5210	9.1729	0.0772	**1.0300**	**18.1371**	7.9492
$F_M(x_{212})$	**0.2419**	6.4289	9.6057	0.0770	**0.9199**	**18.1371**	11.7847



(a) Laundry Problem

(b) PIM problem

Fig. 2. Graphical results.

5 Conclusions and Future Work

In this paper, we present an overview of how to use the Pareto Explorer, a global/local exploration tool for the effective numerical treatment of many objective optimization problems, to solve real world applications in the context of the decision-making process. We use it because it is not possible to compute a suitable finite size approximation of the *entire* Pareto set/front for problems with many objectives. Instead, solutions are computed and presented to the DM in a two stage approach, where he/she express the preferences as a direction in objective space. We demonstrated the effectiveness and usefulness of this method with two real-world applications. The use of applications is essential in the context of interactive methods because making fair comparisons is not always possible, due to the fact that each process requires different pieces of information. Moreover, comparisons of the PE against other continuation methods is unfair, as they try to approximate all the set of optimal solutions.

However, the applicability of PE is restricted to continuous MaOPs. As future work, the adaptation of PE for problems with different smoothness assumptions can be explored.

Acknoweledgments. The authors acknowledge support from CONACyT project No. 285599 and SEP Cinvestav project No. 231.

References

1. Alvarado-Iniesta, A., Cuate, O., Schütze, O.: Multi-objective and many objective design of plastic injection molding process. The International Journal of Advanced Manufacturing Technology 102(9), 3165–3180 (Jun 2019)
2. Branke, J., Deb, K., Miettinen, K., Slowinski, R.: Multiobjective optimization: Interactive and evolutionary approaches, vol. 5252. Springer Science & Business Media (2008)
3. Buchanan, J.T.: A naive approach for solving mcdm problems: The guess method. Journal of the Operational Research Society 48(2), 202–206 (1997)
4. Deb, K., Sundar, J., Udaya Bhaskara Rao, N., Chaudhuri, S.: Reference point based multi-objective optimization using evolutionary algorithms. International Journal of Computational Intelligence Research 2(3), 273–286 (2006)
5. Eskelinen, P., Miettinen, K., Klamroth, K., Hakanen, J.: Pareto navigator for interactive nonlinear multiobjective optimization. OR spectrum 32(1), 211–227 (2010)
6. Fliege, J., Drummond, L.G., Svaiter, B.F.: Newton's method for multiobjective optimization. SIAM Journal on Optimization 20(2), 602–626 (2009)
7. Hillermeier, C.: Nonlinear multiobjective optimization: A generalized homotopy approach, vol. 135. Springer (2001)
8. Jaszkiewicz, A., Słowiński, R.: The 'light beam search'approach–an overview of methodology applications. European Journal of Operational Research 113(2), 300–314 (1999)
9. Martín, A., Schütze, O.: Pareto tracer: a predictor-corrector method for multi-objective optimization problems. Engineering Optimization 50(3), 516–536 (2018)
10. Miettinen, K.: Nonlinear multiobjective optimization, vol. 12. Springer (1999)
11. Miettinen, K., Eskelinen, P., Ruiz, F., Luque, M.: Nautilus method: An interactive technique in multiobjective optimization based on the nadir point. European Journal of Operational Research 206(2), 426–434 (2010)
12. Miettinen, K., Mäkelä, M.M.: Interactive multiobjective optimization system www-nimbus on the internet. Computers & Operations Research 27(7), 709–723 (2000)
13. Peitz, S., Dellnitz, M.: A survey of recent trends in multiobjective optimal control—surrogate models, feedback control and objective reduction. Mathematical and Computational Applications 23(2) (2018)
14. Schütze, O., Cuate, O., Martín, A., Peitz, S., Dellnitz, M.: Pareto explorer: a global/local exploration tool for many-objective optimization problems. Engineering Optimization pp. 1–24 (2019)
15. Tamime, A.Y.: Cleaning-in-place: dairy, food and beverage operations, vol. 13. John Wiley & Sons (2009)
16. Wang, H.: Zigzag search for continuous multiobjective optimization. INFORMS Journal on Computing 25(4), 654–665 (2012)
17. Wierzbicki, A.P.: A mathematical basis for satisficing decision making. Springer (1981)

ISSN 1870-4069

Elements of C-LSHADE Algorithm: An Empirical Study on Mechatronic Design Problems

Sebastián José de la Cruz Martínez[1], Efrén Mezura Montes[2]

[1] Laboratorio Nacional de Informática Avanzada,
Mexico

[2] Centro de Investigación en Inteligencia Artificial,
Mexico

emezura@uv.mx, sdelacruz.mca17@lania.edu.mx

Abstract. This paper presents an experimental study of the components of the Constrained Success History-Based Adaptive Differential Evolution with Linear Population Size Reduction (C-LSHADE) algorithm, to clarify its importance in generating good results by solving two instances of mechatronic optimal design. C-LSHADE has four main components: (1) a historical memory to adapt CR and F parameters, (2) a mutation strategy called *current-to-pbest*, (3) a constraint handling technique based on feasible rules; and (4) a function that linearly reduces the population size over generations. Based on the final results, the linear population size decreasing is the only component that, if omitted, affects the performance of the algorithm.

Keywords: evolutionary algorithms, differential evolution, dimensional synthesis, four-bar mechanism.

1 Introduction

A particular problem when designing mechatronic systems is finding the optimal dimensional synthesis of mechanisms to perform a prescribed task in the best possible way. The dimensional synthesis is responsible for specifying angular positions and lengths of each component to find solutions to problems of trajectory, function or movement generation to established specifications [4]. Such problem is solved by treating it as a numerical optimization problem. There are different optimization techniques, which could be classified as follows: traditional, stochastic, statistical and modern or nontraditional techniques [6]. According to the specialized literature, there is evidence of their usage to solve optimal design problems: prebil performed a study to find the optimal dimensional synthesis of a mechanism used as hydraulic support in the mining industry with the help of a gradient method generalization called Adaptive Grid Refinement algorithm

(AGR), where the distance between an arbitrary coupler point and a prescribed path is minimized.

saravanan employed Multi-objective Genetic Algorithm (MOGA), Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II) and Multi-objective Differential Evolution (MODE) to find geometric dimensions of three end effectors, optimal Pareto front and decrease the computational time involved in solving the problem. They also make a comparison between the algorithms through multi-objective performance measures and propose a software package for users who wish to solve a design problem in any field of study. In [2], the authors solved the synthesis of an Ackermann Steering Mechanism considering linkage lengths and distribution of precision points as optimization parameters using an algorithm inspired on the biological immune system of vertebrates. Zapata in [14] added a constraint-handling mechanism to the algorithm LSHADE, originally designed to solve unconstrained optimization problems, obtaining very competitive results when solving mechanical design problems. However, as C-LSHADE has different mechanisms within, it is unknown which ones are responsible of such good performance.

Motivated by the above, this paper proposes an experimental study of the C-LSHADE algorithm to clarify the importance of its components in obtaining good results when solving two optimal design problems.

The document is organized as follows: Section 2 presents the dimensional synthesis of a four-bar mechanism as well as case studies to be solved. Section 3 provides a description of the C-LSHADE algorithm. Section 4 shows the experimental results achieved as their discussion. Finally, Section 5 presents conclusions and future work lines.

2 Synthesis of Four-bar Linkage Mechanisms

Let be a four-bar mechanism type crank-rod-rocker shown in Figure 1, built by a reference bar (r_1), an input bar or crank (r_2), connecting rod or coupler (r_3) and an output bar or rocker (r_4). Two coordinate systems are established, the first fixed to the real world (O_1) and the second one for reference (O_2), where (x_0, y_0) is the distance between both systems, θ_0 corresponds to the mechanism's angle movement according to the horizontal axis, angles $\theta_1, \theta_2, \theta_3$ and θ_4 corresponding to the four bars angles and C(r_{cx}, r_{cy}) point that defines the coupler position [8].

In this work, it is desired to obtain the optimal design of a four-bar mechanism with the least possible error, that is, the coupler's point C must proceed as accurately as possible between the precision points C_d^i and the lowest distance of calculated points C_i. The suggested objective function is as follows (Eq. 1):

$$error = \sum_{i=1}^{n} \left[\left(C_{xd}^i - C_x^i \right)^2 + \left(C_{yd}^i - C_y^i \right)^2 \right]. \tag{1}$$

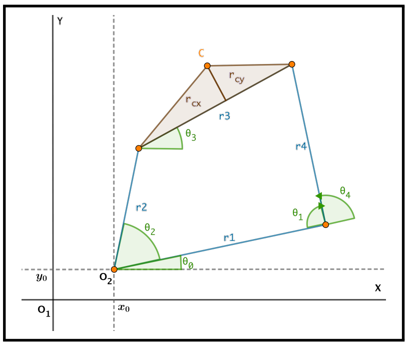


Fig. 1: Four-bar mechanism.

Subject to :

$$
\begin{aligned}
g_1\left(\overrightarrow{p}\right) &= p_1 + p_2 - p_3 - p_4 \leq 0,\\
g_2\left(\overrightarrow{p}\right) &= p_2 - p_3 \leq 0,\\
g_3\left(\overrightarrow{p}\right) &= p_3 - p_4 \leq 0,\\
g_4\left(\overrightarrow{p}\right) &= p_4 - p_1 \leq 0,
\end{aligned}
\tag{2}
$$

where $C_d^i = \left[C_{xd}^i, C_{yd}^i\right]^T$ is a precision point that defines the trajectory, a set of them as $\Omega = \{C_d^i | i \in N\}$ where N is the total number of points and $C^i = \left[C_x^i, C_y^i\right]$, each generated point expressed in accordance with the input bar and the set of bar lengths and their parameters x_0, y_0 and θ_0. For all case studies, 200 points C_i were considered. The kinematics of the mechanism can be found in [14,**?**].

Eq. 3 is a representation of the design variables vector established to four-bar mechanisms in this work:

$$
\begin{aligned}
\overrightarrow{p} &= \left[p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9\right],\\
&= \left[r_1, r_2, r_3, r_4, r_{cx}, r_{cy}, \theta_0, x_0, y_0\right],
\end{aligned}
\tag{3}
$$

where variables r_1, r_2, r_3, r_4 correspond to bar lengths, r_{cx}, r_{cy} correspond to coupler position, θ_0 movement angle of the mechanism concerning the horizontal axis of the second system and $O_2(x_0, y_0)$ starting point of the latter.

2.1 Numerical Optimization Problems

This section presents the optimization problems to be solved. To identify them, each problem was labeled with the capital letter M, associated with the word "mechanism"; and an integer, problem's index in the problem set enumeration.

(M01) Mechanism that follows a vertical linear path. Study case taken from [10], the dimensional synthesis of a mechanism that follows a vertical linear path defined by six points of precision with the least possible error is sought. The set of precision points is:

$$\Omega = \{(20,20),(20,25),(20,30),(20,35),(20,40),(20,45)\}. \tag{4}$$

Design variables vector has nine dimensions (Eq. 3). The boundaries defined for each one of them are:

$$\begin{aligned} & r_1, r_2, r_3, r_4 \in [0,60], \\ & r_{cx}, r_{cy}, x_0, y_0 \in [-60,60], \\ & \theta_0 \in [0,2\pi]. \end{aligned} \tag{5}$$

The objective function to this single-objective problem is presented in Eq. 1, subject to constraints shown in Eq. 2.

(M02) Mechanism that follows a path defined by five precision points. Problem recovered from [14], the coupler must crosses through five points of precision that form a curve. The precision points are:

$$\Omega = \{(3,3),(2.759,3.363),(2.372,3.663),(1.890,3.862),(1.355,3.943)\}. \tag{6}$$

The design vector has nine variables (Eq. 3). The suggested upper and lower values for each one of them are:

$$\begin{aligned} & r_1, r_2, r_3, r_4 \in [0,50], \\ & r_{cx}, r_{cy} \in [-50,50], \\ & x_0, y_0, \theta_0 = 0. \end{aligned} \tag{7}$$

The single-objective problem described in Eq. 1 is considered, subject to the constraints shown in Eq. 2.

3 Constrained Success History Based Adaptive DE with Linear Population Size Reduction

Proposed in [14], C-LSHADE is an algorithm focused on solving constrained optimization problems. Its components were borrowed from previous proposals: the mutation strategy was acquired from JADE [15], the historical memory 1.1 and the linear population size reduction function were inherited from L-SHADE [13]. In order to solve constrained problems, the Feasibility Rules constraint-handling technique was added [1]. Its components are briefly detailed, but a full explanation can be found in [14].

Parameter control based on historical memory. A historical memory is composed of M_{CR} and M_F structures of H dimensions for control parameters CR and F. Parameters CR_i and F_i of each individual are calculated by randomly selecting a memory space with index $r_i \in [1, H]$ as well as using Eqs. 8 and 9 corresponding to each one of them:

$$CR_i = \begin{cases} 0 & if\ M_{CR_{ri}} = \perp \\ randn_i(M_{CR_{ri}}, 0.1) & otherwise, \end{cases} \tag{8}$$

$$F_i = randc_i\left(M_{F_{ri}}, 0.1\right), \tag{9}$$

where $\perp = -1$ is a threshold, $randn_i$ a normal distribution and $randc_i$ a Cauchy distribution. If CR_i exceeds its limits, it is biased to the nearest. Similarly, when $F_i \geq 1$, is truncated to 1 and if $F_i \leq 0$ is regenerated. $CR_{i,g}$ and F_i that produced successful solutions are stored in S_{CR} and S_F structures. In the same way, the difference between objective functions values is stored in a similar structure. With the stored information, the memory content is updated as indicated in the Algorithm 1.

***Current-to-pbest* mutation strategy.** It includes information of the best individual with the aim to improve the convergence by varying the diversity of the population; and a p parameter that limits the selection space to control the convergence of the method during the search process. A representation of such operator is the expressed in Eq. 10:

$$v_{i,g} = x_{i,g} + F_i \cdot \left(x^p_{Best_g} - x_{i,g} \right) + F_i \cdot (x_{r1,g} - x_{r2,g}) \, , \tag{10}$$

where $x^p_{Best_g}$ is randomly selected from the $100p\%$ of the population and $p \in [0, 1]$.

Survivor Selection. C-LSHADE uses a constraint-handling technique called Feasiblility Rules, which is composed of the following three conditions:

- Between two infeasible individuals, the one with the smallest sum of constraint violation (SVR or ϕ_x) is selected. SVR is expressed in Eq. 11.
- A feasible individual is preferable over an infeasible one.
- Between two feasible individuals, the one with the best objective function value is preferred.

$$\phi_x = \sum_{j=1}^{m} max(0, g_j(x)). \tag{11}$$

Update of historical memory spaces. The updating of the averages contained in the memory is performed by Algorithm 1. In this, the index $k \in [1, H]$ is associated with the memory space to be updated. At the beginning k = 1, this is increased when update memory is performed and restored if $k > H$. Moreover, $mean_{WL}$ is remitted to the Lehmer's weighted average (Eq. 12) where wk refers to the difference between fitness functions values in order to provide information on the adaptation of parameters.

$$mean_{WL}(S_F) = \frac{\sum_{k=1}^{|S_F|} wk \cdot S^2_{F,k}}{\sum_{k=1}^{|S_F|} wk \cdot S_{F,k}}, \quad wk = \frac{\Delta fx}{\sum_{k=1}^{|S_F|} \Delta fx}, \quad \Delta fk = |f(u_{k,g}) - f(x_{k,g})|. \tag{12}$$

Linear Population Size Reduction (LPSR). It linearly reduces the population size with respect to the number of evaluations of the objective function, where its initial size is N_{init} and at the end is N_{min}. The population size for each generation is calculated according to Eq. 13:

$$NP_{G+1} = round\left[\left(\frac{N_{min} - N_{init}}{MAX\_NFE} \right) \right] * NFE + N_{init}, \tag{13}$$

Algorithm 1 Memory Update 1.1

```
if $S_{CR} \neq \emptyset$ y $S_F \neq \emptyset$ then
    if $M_{CR,k,g} = \perp$ or $max\left(S_{CR}\right) = 0$ then
        $M_{CR,k,g+1} = \perp$;
    else
        $M_{CR,k,g+1} = mean_{WL}\left(S_{CR}\right)$;
    end if
    $M_{F,k,g+1} = mean_{WL}\left(S_F\right)$;
    $k++$;
    if $k > H$ then
        $k = 1$;
    end if
else
    $M_{CR,k,g+1} = M_{CR,k,g}$;
    $M_{F,k,g+1} = M_{F,k,g}$;
end if
```

where MAX_NFE is the maximun number of evaluations, and NFE is the current number of evaluations of the objective function. This mechanism is activated when $NP_{G+1} < NP_G$, where NP_G corresponds to number of individuals in the current population.

Algorithm 2 is a general representation of C-LSHADE.

Algorithm 2 C-LSHADE

```
Require: $H, p, N_{init}, N_{min}$
Ensure: $P(x)$
Begin
$NP = D * N_{init}$
Create $P(x_{i,0})$ where i = 1,...,$NP$ and evaluate$f(x_{i,0})$
Set the content of $M_{CR,i}, M_{F,i}(i = 1, ..., H) = 0.5$
$g = 0, k = 1$
while stop criteria not met do
    Create $S_F = \emptyset, S_{CR} = \emptyset, S_{DIF} = \emptyset$
    Sort population indexes ascendingly
    for i=1 to NP do
        $r_i = randi\left(1, H\right)$
        Compute $CR_{i,g}$ y $F_{i,g}$ based on the Eqs. 8 and 9
        Create $u_{i,g}$ based on current-to-pbest/1/bin (Eq. 10)
        if $f(u_{i,g}) < f(x_{i,g})$ based on Factible Rules then
            $x_{i,g+1} = u_{i,g}$
            $S_{CR} = S_{CR} \cup CR_i$
            $S_F = S_F \cup F_i$
            $S_{DIF} = S_{DIF} \cup |f\left(u_{i,g}\right) - f\left(x_{i,g}\right)|$
        end if
    end for
    Update memory based on the Algorithm 1
    Compute $NP_{g+1}$ according to Eq. 13
    if $NP_{g+1} < NP$ then
        Sort population indexes ascendingly giving priority to the SVR and then to the fitness.
        Delete worse $NP - NP_{g+1}$.
    end if
    $g++$
end while
```

4 Results and Analysis

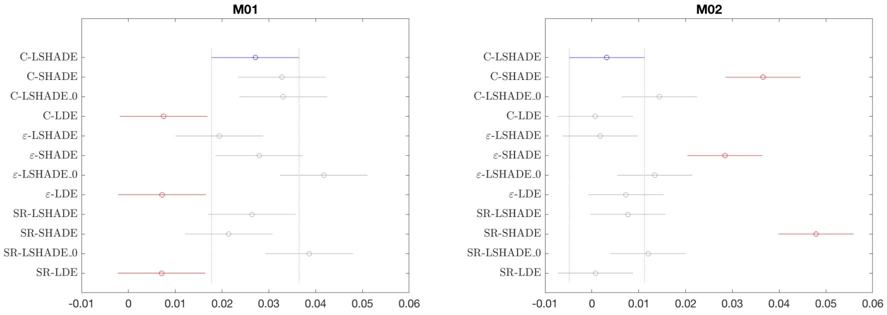
To study the C-LSHADE components, the following configurations were proposed:

- **Constraint handling technique:** Instead of the Feasibility Rules by ε-Constrained [11] and Stochastic Ranking [7] methods were adopted.
- **Linear population Size Reduction function:** Deactivate it.
- **Parameter adaptation scheme:** Replace historical memory update algorithm 1.1 by version 1.0 proposed in [12], and compute CR per individual and F at each generation.

Different variants of the algorithm were generated, grouped by the studied component and denoted as follows: (1) corresponds to versions that have different constraint-handling technique: LSHADE with ε-Constrained Method, ε-LSHADE, and LSHADE with Stochastic Ranking, SR-LSHADE; (2) variants without the population reduction mechanism: C-SHADE, ε-SHADE, and SR-SHADE; (3) variants with the historical memory version 1.0: C-LSHADE_0, ε-LSHADE_0, and SR-LSHADE_0; and (4) variants that compute CR and F dynamically: C-LDE, ε-LDE, and SR-LDE. A statistical comparison among the C-LSHADE variants was carried out to achieve the purpose of this study. The Kruskal-Wallis and the post-hoc Bonferroni tests were used. Each test was applied with 95% confidence. The experiments were performed on a computer with an Intel Core i7 - 2.5 GHz processor, 8 GB of RAM and 64-bit Windows 10 operating system. The algorithms and statistical analysis were developed in the M language using the MATLAB 2018a IDE. For all algorithms, 31 independent runs were performed to solve both optimization problems and the parameters recommended in [14] were used: $H = 6$, $p = 0.11$, $N_{init} = 18$ and $N_{min} = 4$. Likewise, the parameter values of the constraint handling techniques were taken from [7,**?**]: for Stochastic Ranking $Pf = 0.45$ and for ε-Constrained $cp = 0.5$, $\theta_0 = 0.2$, $Tc = 0.2MAX\_NFE$. The stop criterion was set at 400,000 maximun number of evaluations of the objective function (MAX_NFE) for M01 and 15,000 for M02. Case studies are treated as single-objective numerical optimization problems subject to constraints (Eq. 2) with the aim to minimize the trajectory tracking error. The complexity of the studies cases is high due to the amount of precision points that the coupler's point C must pass and the effort required to find a combination of design variables that allow a successful solution compared to the most known state-of-the-art algorithms. In general, all studied algorithms found feasible solutions in every independent run. Figure 2 shows the Bonferroni test results for the two test problems. There was significant differences in performance among the algorithm variants with different constraint-handling techniques (C-LSHADE, ε-LSHADE and SR-LSHADE) and those variants without the parameter adaptation scheme (C-LDE, ε-LDE and SR-LDE), of which the latter obtained better results for the M01 problem, see Figure 2a. Regarding this problem M01, all C-LSHADE mechanisms were removed and the good performance was still present by using any constraint-handling technique adopted in this paper. Concerning M02, those variants without a population reduction mechanism (C-SHADE, ε-SHADE and SR-SHADE) had a worse behavior, see Figure 2b. In contrast, the variants of group 4 (C-LDE, ε-LDE and SR-LDE) demonstrated better performance than the rest. Regarding problem MO2, the linear reduction is the only required mechanism by the algorithm to provide better results.

5 Conclusions and Future Work

This work proposed an empirical study of the C-LSHADE algorithm in order to determine the importance of its components in solving two mechatronic design optimization



(a) **(b)**

Fig. 2: Bonferroni post-hoc test based on final results. There are significant differences when the confidence intervals do not overlap. A variant is considered with a better performance when its confidence interval is closer to zero.

problems. The constraint-handling technique, the population size reduction and the historical memory version for parameter adaptation were the mechanisms under study. The overall results indicate that the only mechanism that must be present in the algorithm to provide competitive results, particularly for the second test problem, is the linear decreasing mechanism of the population size. In fact, a simplified version of the algorithm could successfully resolve the first test problem. As future work, the linear function for the population size reduction will be further analyzed and other case studies will be solved.

Acknowledgments. The first author acknowledges support from CONACyT through a scholarship to pursue graduate studies at LANIA.

References

1. Deb, K.: An efficient constraint handling method for genetic algorithms. Computer Methods in Applied Mechanics and Engineering 186(2-4), 311–338 (June 2000)
2. Hamidi, Y.E., Javash, M.S., Ettefagh, M.M., Doust, F.A.: Optimization of four-bar steering mechanism through Artificial Immune System (AIS) Algorithm. In: 12th International Conference on Control, Automation and Systems. pp. 913–917 (October 2012)
3. Hernández, B., del P. Pozos, M., Mezura, E., Portilla, E.A., Vega, E., Calva, M.B.: Two-Swim Operators in the Modified Bacterial Foraging Algorithm for the Optimal Synthesis of Four-Bar Mechanisms. Hindawi Publishing Corporation Computational Intelligence and Neuroscience p. 18 (January 2016)
4. Myszka, D.H.: Machines and mechanisms. Prentice Hall, 4th edn. (2004)
5. Prebil, I., Krašna, S., Ciglarič, I.: Synthesis of four-bar mechanism in a hydraulic support using a global optimization algorithm. Structural and Multidisciplinary Optimization 24(3), 246–251 (September 2002), https://doi.org/10.1007/s00158-002-0234-y
6. Rao, S.S.: Engineering Optimization: Theory and Practice. John Wiley and Sons, Inc., River Street,Hoboken, New Jersey, 4th edn. (June 2009)

7. Runarsson, T.P., Yao, X.: Stochastic ranking for constrained evolutionary optimization. IEEE Transactions on Evolutionary Computation 4(3), 284–294 (September 2000)
8. Sánchez-Márquez, Á., Vega-Alvarado, E., Portilla-Flores, E.A., Mezura-Montes, E.: Synthesis of a planar four-bar mechanism for position control using the harmony search algorithm. In: 11th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE). pp. 1–6 (September 2014)
9. Saravanan, R., Ramabalan, S., Godwin Raja Ebenezer, N., Dharmaraja, C.: Evolutionary multi criteria design optimization of robot grippers. Applied Soft Computing 9(1), 159 – 172 (2009)
10. Sleesongsom, S., Bureerat, S.: Four-bar linkage path generation through self-adaptive population size teaching-learning based optimization. Knowledge-Based Systems (August 2017)
11. Takahama, T., Sakai, S.: Constrained optimization by the ε constrained differential evolution with an archive and gradient-based mutation. In: IEEE Congress on Evolutionary Computation. pp. 1–9 (July 2010)
12. Tanabe, R., Fukunaga, A.: Success-history based parameter adaptation for Differential Evolution. In: 2013 IEEE Congress on Evolutionary Computation. pp. 71–78 (June 2013)
13. Tanabe, R., Fukunaga, A.: Improving the Search Performance of SHADE Using Linear Population Size Reduction. IEEE Congress on Evolutionary Computation pp. 1658–1665 (July 2014)
14. Zapata, M.F.: Control de Parámetros del Algoritmo Evolución Diferencial con Variantes Combinadas para la Solución de Problemas de Optimización en Mecatrónica. Master's thesis, Laboratorio Nacional de Informática Avanzada, Xalapa, Veracruz, México (Agosto 2017)
15. Zhang, J., Sanderson, A.C.: JADE: Adaptive Differential Evolution With Optional External Archive. IEEE Transactions on Evolutionary Computation 13(5), 945–958 (October 2009)

ISSN 1870-4069

A Model for Disaggregated Data Using Gini Index

Adriana Laura López Lobato[1], Martha Lorena Avendaño Garrido[2],
Johan Van Horebeek[2]

[1] Universidad Veracruzana, Facultad de Matemáticas,
Mexico

[2] Centro de Investigación en Matemáticas A.C.,
Mexico

adrilau17@gmail.com

Abstract. Corrado Gini developed in 1914 a methodology to measure the difference between two probability distributions, the Gini Index. In this paper, we propose the Bimodal Gini Index. We based this model on the definition of the Gini Coefficient, a model of independence between two distributions, so we set a model that approximates the Gini Index with the supposition that the searched distribution is a linear combination of independent distributions, without adding a lot of computational cost. We show some applications in political sciences concerning voting problems to illustrate the performance of the Bimodal Gini Index.

Keywords: Gini index, Gini coefficient, probability estimation.

1 Introduction

The Gini Index is a measure of the level of inequality between two probability distributions. It is applied in several fields of study like engineering, ecology, transport and economics, see [8].

The Gini Index problem is a particular case of Monge's mass transfer problem, as we will see in the following section. This problem always has a solution that is a distance between the involved probability distributions, but it can be very expensive to find it, computationally speaking, see [9] and [12]. To handle these expensive calculations, the Gini Coefficient was introduced as a natural upper bound of the Gini Index. The Gini Coefficient has several applications, many of them in economics and sociology, [2] and [8]. However, it differs a lot from the value of the Gini Index.

In this work we present the Bimodal Gini Index, a model that is a better approximation to the Gini Index than the Gini Coefficient with a low computational cost, by taking the Gini Index problem and doing the supposition that the searched probability is a linear combination of independent probability distributions. With this model, we reduced the number of variables and the restrictions of the Gini Index problem and can be solved by using numerical optimization.

Also, it has several interesting properties, among these we highlight that it can be split in two linear programming problems, both easily solved by the simplex method.

2 Gini Index and Gini Coefficient

Let X be a discrete random variable with n elements and two probability distributions p and q on X. The Gini Index problem (GI) can be stated as:

$$\begin{aligned}
\text{Minimize: } & \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}\pi_{ij}, && && (1)\\
\text{subject to: } & \pi_{ij} \geq 0, && \text{for all } i,j && (2)\\
& \sum_{j=1}^{n} \pi_{ij} = p_i, && i = 1,2,...,n && (3)\\
& \sum_{i=1}^{n} \pi_{ij} = q_j, && j = 1,2,...,n && (4)\\
& \sum_{i=1}^{n}\sum_{j=1}^{n} \pi_{ij} = 1, && && (5)
\end{aligned}$$

where $p_i = p(x_i)$ y $q_i = q(x_i)$ for $i = 1, ..., n$, the cost function is a distance function $d_{ij} = d(x_i, x_j)$ on $X \times X$, for all i and j, and $\pi_{ij} = \pi(x_i, x_j)$ denotes the variables. The solution is a probability distribution $\pi^* = \{\pi^*_{ij} : i = 1, ..., n, j = 1, ..., n\}$. We define the Gini Index for the distributions p and q, denoted by $GI(p, q)$, as the optimal value of the GI problem. Note that there are n^2 no negative variables, then the solution of the problem can be expensive to find for large n, even with the use of computational tools. For more information about the Gini index and its problem in both forms, continuos and discrete, see [7,12,13].

On the other hand, we have a "measure of uncertainty" of a random variable, the Gini Coefficient for a discrete random variable X with n elements and two probability distributions p and q on X, see [1]:

$$GC(p, q) = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij} p_i q_j.$$

With these definitions we can establish the following inequality

$$GI(p, q) \leq GC(p, q).$$

The Gini Index and the Gini Coefficient are used as indicators of social and economic inequality, as we can see in the articles [3,10,11].

3 Proposed Model: Bimodal Gini Index

To set the Bimodal Gini Index we consider the Gini Index problem and we add the additional assumption that the searched probability distribution π is a linear combination of independent distributions, that means, it has the form:

$$\pi_{ij} = \alpha f_i^{(1)} f_j^{(2)} + (1-\alpha) g_i^{(1)} g_j^{(2)}, \tag{6}$$

where $\alpha \in (0,1)$ and $f^{(1)}, f^{(2)}, g^{(1)}$ and $g^{(2)}$ are independient probability distributions pairwise on X, this is that $f^{(1)}$ and $f^{(2)}$ are independent and $g^{(1)}$ and $g^{(2)}$ are independent.

As $f^{(1)}, f^{(2)}, g^{(1)}$ and $g^{(2)}$ are probability distributions, by replacing (6) in the expressions (3) and (4) we obtain:

$$p_i = \alpha f_i^{(1)} + (1-\alpha) g_i^{(1)}, \text{ for all } i \quad \text{and} \quad q_j = \alpha f_j^{(2)} + (1-\alpha) g_j^{(2)}, \text{ for all } j,$$

and expressing the variables $g_i^{(1)}$ and $g_j^{(2)}$ in terms of $f_i^{(1)}$ and $f_j^{(2)}$, respectively, as:

$$g_i^{(1)} = \frac{p_i - \alpha f_i^{(1)}}{1-\alpha} \quad \text{and} \quad g_j^{(2)} = \frac{q_j - \alpha f_j^{(2)}}{1-\alpha}, \text{ for all } i, j \text{ and } \alpha \in (0,1),$$

we can express the variables π_{ij} only in terms of $f_i^{(1)}$ and $f_j^{(2)}$ as:

$$\pi_{ij} = \frac{\alpha}{1-\alpha}\left(f_i^{(1)} f_j^{(2)} - p_i f_j^{(2)} - q_j f_i^{(1)} + \frac{1}{\alpha} p_i q_j \right). \tag{7}$$

Also, we can express the values of $f_n^{(k)}$, with $k = 1, 2$, by $f_n^{(k)} = 1 - \sum_{i=1}^{n-1} f_i^{(k)}$, when we use this expressions in (7) we can define the following functions:

$$h_1(f_i^{(1)}, f_j^{(2)}) = f_i^{(1)} f_j^{(2)} - p_i f_j^{(2)} - q_j f_i^{(1)} + \frac{1}{\alpha} p_i q_j, \text{ for } i = 1, ..., n-1, \ j = 1, ..., n-1,$$

$$h_2(f^{(1)}, f_j^{(2)}) = f_j^{(2)} \left(1 - \sum_{i=1}^{n-1} f_i^{(1)} - p_n \right) + q_j \left(\sum_{i=1}^{n-1} f_i^{(1)} - 1 + \frac{1}{\alpha} p_n \right), \text{ for } j = 1, ..., n-1,$$

$$h_3(f_i^{(1)}, f^{(2)}) = f_i^{(1)} \left(1 - \sum_{j=i}^{n-1} f_j^{(2)} - q_n \right) + p_i \left(\sum_{j=1}^{n-1} f_j^{(2)} - 1 + \frac{1}{\alpha} q_n \right), \text{ for } i = 1, ..., n-1,$$

where $f^{(1)} = (f_1^{(1)}, ..., f_{n-1}^{(1)})$ and $f^{(2)} = (f_1^{(2)}, ..., f_{n-1}^{(2)})$, then we define

$$H(f^{(1)}, f^{(2)}) = \frac{\alpha}{1-\alpha} \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} d_{ij} h_1(f_i^{(1)}, f_j^{(2)}) + \sum_{j=1}^{n-1} d_{ij} h_2(f^{(1)}, f_j^{(2)}) + \sum_{i=1}^{n-1} d_{ij} h_3(f_i^{(1)}, f^{(2)}).$$

This function only depends on the first $n-1$ variables of the distributions $f^{(1)}$ and $f^{(2)}$. Also we have:

$$\frac{p_i - (1-\alpha)}{\alpha} \leq f_i^{(1)} \leq \frac{p_i}{\alpha}, \quad \frac{q_j - (1-\alpha)}{\alpha} \leq f_j^{(2)} \leq \frac{q_j}{\alpha}, \quad \text{for all } i, j \text{ and } \alpha \in (0,1).$$

Thus, we define the Bimodal Gini Index (BGI) as:

$$\begin{aligned}
&\text{Minimize: } H(f^{(1)}, f^{(2)}),\\
&\text{subject to: } \max\left\{0, \frac{p_i-(1-\alpha)}{\alpha}\right\} \le f_i^{(1)} \le \min\left\{1, \frac{p_i}{\alpha}\right\},\ i=1,...,n-1,\\
&\qquad \max\left\{0, \frac{q_j-(1-\alpha)}{\alpha}\right\} \le f_j^{(2)} \le \min\left\{1, \frac{q_j}{\alpha}\right\},\ j=1,...,n-1,\\
&\qquad \max\left\{0, \frac{p_n-(1-\alpha)}{\alpha}\right\} \le 1-\sum_{i=1}^{n-1} f_i^{(1)} \le \min\left\{1, \frac{p_n}{\alpha}\right\},\\
&\qquad \max\left\{0, \frac{q_n-(1-\alpha)}{\alpha}\right\} \le 1-\sum_{j=1}^{n-1} f_j^{(2)} \le \min\left\{1, \frac{q_n}{\alpha}\right\}.
\end{aligned}$$

If $f^* = (f^{(1)*}, f^{(2)*})$ is the optimal solution of the previous problem, then we define the Bimodal Gini Index as:

$$BGI(p,q) = H(f^{(1)*}, f^{(2)*}).$$

Note that the Gini Index problem has n^2 no negative variables and $2n+1$ equality restrictions. With the proposed model we can reduce this amount to $2(n-1)$ variables, $2(n-1)$ box restrictions and 2 linear box restrictions.

Moreover, the Bimodal Gini Index is a better bound for the Gini Index than the Gini Coefficient, that is, the following inequality is fulfilled:

$$GI(p,q) \le BGI(p,q) \le GC(p,q).$$

So, we add the additional assumption that the searched probability distribution π is a linear combination of independent distributions, as in 6, based on the idea of independence given by the Gini Coefficient, to make it more complex without adding a lot of computational cost:

- If α takes the value 0 or 1 in (6), then the optimal value of the BGI problem and the value of the Gini Coefficient will be the same.
- The objective function H of the BGI problem is a convex and symmetric function of α and reaches its minimum value in 1/2 (or in α close to 1/2).
- We can separate the BGI problem in two linear programming problems, both solved by the simplex method, as we will see in the following section.

3.1 Approximation to the Bimodal Gini Index

We can express the function $H(f^{(1)}, f^{(2)})$ as:

$$H(f^{(1)}, f^{(2)}) = H_L(f^{(1)}) + H_L(f^{(2)}) + H_C(f^{(1)}, f^{(2)}) + C,$$

where:

$$H_L(f^{(1)}) = \frac{\alpha}{1-\alpha}\left[\sum_{i=1}^{n-1}\sum_{j=1}^{n} d_{ij}(-q_j f_i^{(1)}) + \sum_{i=1}^{n-1}\sum_{j=1}^{n-1} d_{nj} q_j f_i^{(1)} + \sum_{i=1}^{n-1} d_{in} f_i^{(1)}\right],$$

$$H_L(f^{(2)}) = \frac{\alpha}{1-\alpha}\left[\sum_{i=1}^{n}\sum_{j=1}^{n-1} d_{ij}(-p_i f_j^{(2)}) + \sum_{i=1}^{n-1}\sum_{j=1}^{n-1} d_{in} p_i f_j^{(2)} + \sum_{j=1}^{n-1} d_{nj} f_j^{(2)}\right],$$

$$H_C(f^{(1)}, f^{(2)}) = \frac{\alpha}{1-\alpha}\left[\sum_{i=1}^{n-1}\sum_{j=1}^{n-1} d_{ij} f_i^{(1)} f_j^{(2)} - \sum_{i=1}^{n-1}\sum_{j=1}^{n-1} d_{nj} f_i^{(1)} f_j^{(2)} - \sum_{i=1}^{n-1}\sum_{j=1}^{n-1} d_{in} f_i^{(1)} f_j^{(2)}\right],$$

$$C = \frac{1}{1-\alpha}\left[\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij} p_i q_j\right] + \frac{\alpha}{1-\alpha}\left[\sum_{j=1}^{n-1} d_{nj} q_j + \sum_{i=1}^{n-1} d_{in} p_i\right].$$

The linear functions $H_L(f^{(1)})$ and $H_L(f^{(2)})$ depends on $f^{(1)}$ and $f^{(2)}$, respectively. The value of C is known. The quadratic function $H_C(f^{(1)}, f^{(2)})$ only have negative values bounded by $-2d$, where d is the maximum distance between the elements of X. We can move the elements of X to a specific range, so d is as small as we desired. Then, we only consider the linear functions, leaving the following separate problems.

Linear problem with respect to $f^{(1)}$:

Minimize: $H_L(f^{(1)})$

$$\text{subject to: } \max\left\{0, \frac{p_i - (1-\alpha)}{\alpha}\right\} \le f_i^{(1)} \le \min\left\{1, \frac{p_i}{\alpha}\right\},\ i = 1, ..., n-1,$$

$$\max\left\{0, \frac{p_n - (1-\alpha)}{\alpha}\right\} \le 1 - \sum_{i=1}^{n-1} f_i^{(1)} \le \min\left\{1, \frac{p_n}{\alpha}\right\}.$$

Linear problem with respect to $f^{(2)}$

Minimize: $H_L(f^{(2)})$

$$\text{subject to: } \max\left\{0, \frac{q_j - (1-\alpha)}{\alpha}\right\} \le f_j^{(2)} \le \min\left\{1, \frac{q_j}{\alpha}\right\},\ j = 1, ..., n-1$$

$$\max\left\{0, \frac{q_n - (1-\alpha)}{\alpha}\right\} \le 1 - \sum_{j=1}^{n-1} f_j^{(2)} \le \min\left\{1, \frac{q_n}{\alpha}\right\}.$$

Then we define the Separated Bimodal Gini Index as:

$$BGIs(p, q) = H(f^{(1)*}, f^{(2)*}),$$

where $f^{(1)*} = (f_1^{(1)*}, ..., f_{n-1}^{(1)*})$ y $f_1^{(2)*} = (f_1^{(2)*}, ..., f_{n-1}^{(2)*})$ are the points where the optimal results are reached in the linear problems with respect to distributions $f^{(1)}$ y $f^{(2)}$, respectively, and H is the objective function previously expressed.

We can obtain the Separated Bimodal Gini Index solving two linear problems by the simplex method, each of one with $n-1$ variables, $n-1$ box restrictions and a linear box restriction. Solving these two problems is much less expensive, computationally speaking, than solving the original one.

– The Bimodal Gini Index and the Separated Bimodal Gini Index take the same value in $\alpha = 1/2$ in computational experiments.

– If we obtain the values of the distributions $f^{(1)}$ and $f^{(2)}$ we can obtain the values of the distribution π of the form (6). The distribution π is of great importance for the application in the following section.

4 Aplication in Political Science

4.1 Voting Data Ohio, 1990

We can see in the Table (1) the data of race of voting-age person and the voting decision for the 1990 election in the Ohio State House, District 42, see [4]. The unobservable values in the empty cells must be found from the observed values in the marginals.

Table 1. Aggregate data for the 1990 election in the Ohio State House, District 42.

Race	Voting decision			
	Democrat	Republican	No vote	
African american				221 (0.313)
White				484 (0.687)
	130 (0.184)	92 (0.131)	483 (0.685)	705 (1.000)

We want to fill this table using the problems raised in the previous section by taking the value of $\alpha = 1/2$, the random variable X ={African american, White, Democrat, Republican, No vote} and the probability distributions $p = \{0.313, 0.687, 0, 0, 0\}$ and $q = \{0, 0, 0.184, 0.131, 0.685\}$. Since the values of the random variable X are categorical, we will use the discrete metric. So, the problems are:

$$\begin{aligned} \text{Minimize:} \quad & 0.315 f_1^{(1)} + 0.315 f_2^{(1)} \\ \text{subject to:} \quad & 0 \leq f_1^{(1)} \leq 0.626, \\ & 0.374 \leq f_2^{(1)} \leq 1, \\ & f_1^{(1)} + f_2^{(1)} = 1. \end{aligned} \qquad \begin{aligned} \text{Minimize:} \quad & f_3^{(2)} + f_4^{(2)} \\ \text{subject to:} \quad & 0 \leq f_3^{(2)} \leq 0.368, \\ & 0 \leq f_4^{(2)} \leq 0.262, \\ & 0 \leq f_3^{(2)} + f_4^{(2)} \leq 0.63. \end{aligned}$$

We found the searching value $BGIs$ in points of the form

$$(f_1^{(1)*}, f_2^{(1)*}, f_3^{(1)*}, f_4^{(1)*}, f_1^{(2)*}, f_2^{(2)*}, f_3^{(2)*}, f_4^{(2)*}) = (f_1^{(1)}, 1 - f_1^{(1)}, 0, 0, 0, 0, 0, 0),$$

with $f_1^{(1)} \in [0, 0.626]$. We analize the solution in the extreme point with $f_1^{(1)} = 0$. So the Table 2 shows the data of interest, the probability distribution π.

Table 2. Results given by the Separated Binomial Gini Index for the 1990 election in the Ohio State House, District 42.

	Democrat	Republican	No vote
African american	0.115184	0.082006	0.11581
White	0.068816	0.048994	0.56919

Note that with the problems of the Separate Gini Index we can obtain the wanted probabilities and a scenario of how the votes in Ohio could have been distributed with respect to the race of the voters.

In [5] three types of results obtained for this same problem given by King are shown, with the particularity that this solutions are interest intervals. Thus, when making a comparison of the puntual results obtained by the IGAs problems, we can notice that these are within the corresponding intervals.

4.2 Elections in the republic of Weimar, 1932

One of the most studied questions in the history is "who voted by Hitler?". In [6] identify some factors that could explain why certain groups of voters joined the Nazi party, concluding that a determining factor was the economic great depression, so the occupations of voters are studied. In the Table (3) we observe the marginals obtained for this problem, the left column of the table denotes each occupational group while the upper row indicates the different political parties.

Table 3. Aggregate data for elections in 1932 in the republic of Weimar.

	Far Left	Left/Center	Far Right	Nazi	Liberal	No vote/ Other	
Self-employed							0.164
Blue collar							0.314
White collar							0.144
Domestic							0.197
Unemployed							0.181
	0.120	0.311	0.049	0.311	0.018	0.191	

The objetive of this problem is filling the Table to answer questions like "what fraction of independent people voted for the Nazi party?". Analyzing historically this type of questions, it is expected that the results related to the working class (blue collar) will be those that favor the Nazi party, since they feared losing their jobs if the centralist party remained in power, see [6]. There are no statistical references for the solution to this problem, our results would be a way to confirm the hypothesis made by researchers in Social Sciences.

We solved the Separated Bimodal Gini Index problems with $\alpha = 1/2$ and the random variable $X = \{$Self-employed, Blue collar, White collar, Domestic, Unemployed, Far Left, Left/Center, Far Right, Nazi, Liberal, No vote/Other$\}$ and the probability distributions $p = \{0.164, 0.314, 0.144, 0.197, 0.181, 0, 0, 0, 0, 0, 0\}$ and $q = \{0, 0, 0, 0, 0, 0.120, 0.311, 0.049, 0.311, 0.018, 0.191\}$. So, we have the following problems:

$$\begin{aligned}\text{Minimize: }\ & 0.809f_1^{(1)} + 0.809f_2^{(1)} + 0.809f_3^{(1)} + 0.809f_4^{(1)} + 0.809f_5^{(1)}\\ \text{subject to: }\ & 0 \le f_1^{(1)} \le 0.328, \quad 0 \le f_2^{(1)} \le 0.628,\\ & 0 \le f_3^{(1)} \le 0.288, \quad 0 \le f_4^{(1)} \le 0.394,\\ & 0 \le f_5^{(1)} \le 0.362, \quad f_1^{(1)} + f_2^{(1)} + f_3^{(1)} + f_4^{(1)} + f_5^{(1)} = 1.\end{aligned}$$

$$\begin{aligned}\text{Minimize: }\ & f_6^{(2)} + f_7^{(2)} + f_8^{(2)} + f_9^{(2)} + f_{10}^{(2)}\\ \text{subject to: }\ & 0 \le f_6^{(2)} \le 0.240, \quad 0 \le f_7^{(2)} \le 0.622,\\ & 0 \le f_8^{(2)} \le 0.098, \quad 0 \le f_9^{(2)} \le 0.622,\\ & 0 \le f_{10}^{(2)} \le 0.036, \quad 0.618 \le f_6^{(2)} + f_7^{(2)} + f_8^{(2)} + f_9^{(2)} + f_{10}^{(2)} \le 1.\end{aligned}$$

The minimum value is reached in the points of the form

$$(f_1^{(1)}, f_2^{(1)}, f_3^{(1)}, f_4^{(1)}, f_5^{(1)}, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, f_6^{(2)}, f_7^{(2)}, f_8^{(2)}, f_9^{(2)}, f_{10}^{(2)})$$

where the values of this variables meet the constraints of the previous problems.

We calculated the values in the Table 4 for the point

$$(0.198, 0.12, 0.258, 0.254, 0.17, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.18, 0.074, 0.098, 0.23, 0.036).$$

This point meets the aforementioned restrictions.

Table 4. Results given by the Separated Binomial Gini Index for the 1932 elections in the republic of Weimar.

	Far Left	Left/Center	Far Right	Nazi	Liberal	No vote/ Other
Self-employed	0.022	0.043	0.009	0.048	0.004	0.038
Blue collar	0.026	0.144	0.006	0.113	0.002	0.023
White collar	0.024	0.018	0.013	0.035	0.005	0.049
Domestic	0.027	0.047	0.013	0.057	0.004	0.049
Unemployed	0.021	0.059	0.008	0.058	0.003	0.032

As we can see, it is true that the working class, blue collar, is the most likely to belong to the Nazi party or the centralist party, as expected.

5 Conclusions and Future Work

The Bimodal Gini Index is a better bound for the Gini Index than the Gini Coefficient. The Bimodal Gini Index has many favorable properties like the Separated Bimodal Gini Index problems. This is possible because of the specific form given to the searched distribution, that reduces the feasible set of the problem. Also, because this model is based in the Gini Coefficient, the computational cost does not increase as much. In this way we reduced the problem in terms of the number of variables and we found a simpler way to solve it by means of two linear problems with box constraints using the simplex method.

We can also observe in the given examples that the problems of the separated Bimodal Gini Index are very useful to solve problems where we have grouped information and we want to obtain data at a disaggregated level. The solved examples are current problems pertinent to political science and history, and their solutions are of great importance for these fields of science.

As future work, we want to use this model in other data bases in different areas of science and in any type of problems that involved disaggregated data or lack of information.

References

1. Bassetti, F., Bodini, A., Regazzini, E.: On Minimum Kantorovich Distance Estimators. Statistics and probability letters 76(12), 1298–1302 (2006)
2. Chakravarty, S.: Ethical Social Index Numbers. Springer Berlin Heidelberg (2012)
3. Han, J., Zhao, Q., Zhang, M.: China's income inequality in the global context. Perspectives in Science 7, 24–29 (2016)
4. King, G.: A Solution to the Ecological Inference Problem: Reconstructing Individual Behavior from Aggregate Data. Princeton University Press (1997)
5. King, G.: A solution to the ecological inference problem: Reconstructing individual behavior from aggregate data. Princeton University Press (2013)
6. King, G., Rosen, O., Tanner, M., Wagner, A.F.: Ordinary Economic Voting Behavior in the Extraordinary Election of Adolf Hitler. The Journal of Economic History 68(4), 951–996 (2008)
7. Peyré, G., Cuturi, M., et al.: Computational optimal transport. Foundations and Trends® in Machine Learning 11(5-6), 355–607 (2019)
8. Rachev, S., Klebanov, L., Stoyanov, S., Fabozzi, F.: The Methods of Distances in the Theory of Probability and Statistics. SpringerLink : Bucher, Springer New York (2013)
9. Rubner, Y., Tomasi, C., Guibas, L.J.: The Earth Mover's Distance as a Metric for Image Retrieval. International journal of computer vision 40(2) (2000)
10. Schneider, M., et al.: Measuring Inequality: The Origins of the Lorenz Curve and the Gini Coefficient. La Trobe University, School of Business (2004)
11. Sturm, J.E., De Haan, J.: Income inequality, capitalism, and ethno-linguistic fractionalization. American Economic Review 105(5), 593–97 (2015)
12. Villani, C.: Topics in Optimal Transportation. American Mathematical Society (2003)
13. Villani, C.: Optimal transport: old and new, vol. 338. Springer Science & Business Media (2008)

ISSN 1870-4069

Towards Windowing as a Sub-Sampling Method for Distributed Data Mining.

David Martínez Galicia[1], Alejandro Guerra Hernández[1],
Nicandro Cruz Ramírez[1], Xavier Limón[2], Francisco Grimaldo[3]

[1]Universidad Veracruzana, Centro de Investigación en Inteligencia Artificial,
México

[2]Universidad Veracruzana, Facultad de Estadística e Informática,
México

[3]Universitat de València, Departament d'Informàtica, Avinguda de la Universitat,
España

davidgalicia@outlook.es, {aguerra, ncruz, hlimon}@uv.mx,
francisco.grimaldo@uv.es

Abstract. Windowing is a sub-sampling method that enables the induction of decision trees with large datasets. Using a small sample of the available training examples, the method can achieve levels of accuracy comparable or better than those obtained using the full available dataset. More relevant is the fact that Windowing-based strategies for Distributed Data Mining (DDM) have shown a correlation between the accuracy of the learned decision tree and the number of examples used to learn it, i.e., the higher the accuracy, the fewer examples used to induce the model. This paper corroborates that this behavior is also observed when adopting inductive algorithms of a different nature than C4.5 or ID3, the algorithms usually adopted when windowing, contributing to the use of Windowing as a general sub-sampling method for DDM. The paper also contributes exploring some metrics to the validation of the obtained sub-samples of examples.

Keywords: sub-sampling, windowing, distributed data mining.

1 Introduction

Windowing is a sub-sampling method that enabled the decision tree inductive algorithms ID3 [9–11] and C4.5 [12, 13] to cope with large datasets, i.e., those whose size precludes loading them in memory. Algorithm 1 defines the method: First, a window is created by extracting a small random sample of the available examples in the full dataset. The main step consists of inducing a model with the window and testing it on the remaining examples, such that all misclassified examples are moved to the window.

Algorithm 1 Windowing.

```
function WINDOWING($Examples$)
    $Window \leftarrow sample(Examples)$
    $Examples \leftarrow Examples - Window$
    repeat
        $stopCond \leftarrow true$
        $model \leftarrow induce(Window)$
        for $example \in Examples$ do
            if $classify(model, example) \neq class(example)$ then
                $Window \leftarrow Window \cup \{example\}$
                $Examples \leftarrow Examples - \{example\}$
                $stopCond \leftarrow false$
    until $stopCond$
    return $model$
```

This step iterates until a stop condition is reached, e.g., all the available examples are correctly classified or a desired level of accuracy is reached.

It has been argued [3] that the method offers three advantages: It copes well with memory limitations, reducing considerably the number of examples required to induce a model of acceptable accuracy. It offers an efficiency gain by reducing the time of convergence, specially when using a separate-and-conquer inductive algorithm, as FOIL [8], instead of the divide-and-conquer algorithms such as ID3 and C4.5. It offers an accuracy gain, specially in noiseless datasets, possibly explained by the fact that learning from a subset of examples may often result in a less over-fitting theory.

Although the lack of memory does not use to be an issue nowadays, similar concerns arise when mining big and/or distributed data. Windowing has been used as the core of a set of strategies for Distributed Data Mining (DDM) [6], obtaining consistent results with respect to the achievable accuracy and the number of examples required by the method. On the contrary, efficiency suffered for large datasets as the cost of testing the models in the remaining examples is not negligible. However, this is alleviated by using GPUs [5]. More relevant for this paper is the fact that the Windowing-based strategies shows a strong correlation (-0.8175845) between the accuracy of the learned decision trees and the number of examples used to induce them, i.e., the higher the accuracy obtained, the fewer the number of examples used to induce the model. Reductions are as big as the 90% of the available training data.

The objective of this work is to corroborate if such a correlation is observed when using inductive algorithms of different nature, so that the advantages of windowing as a sub-sampling method could be generalized beyond decision trees. For this, the paper is organized as follows: Section 2 introduces the adopted methodology; Section 3 presents the obtained results; and Section 4 discusses conclusions and future work. A preliminary contribution of the paper is the study of some metrics to try to validate the obtained windows and to understand the way such sub-sampling works so efficiently in some cases.

2 Methodology

Because of our interest in distributed settings, JaCa-DDM [1] was adopted to run experiments. This tool [6] defines a set of Windowing-based strategies using J48, the Weka [14] implementation of C4.5, as inductive algorithm. Among them, Counter is the most similar to the original formulation of Windowing, excepting that: i) the dataset can be distributed in different sites, and ii) an auto-adjustable stop criteria with a established maximum number of iterations (10) is adopted. The parameters of the strategy, e.g., the maximum number of rounds, are adopted from the literature. The same configuration is used for all the experiments. The Counter strategy is tested on the datasets shown in Table 1, selected from the UCI [2] and MOA [1] repositories. They vary in the number of instances, attributes, and class' values; as well as in the type of the attributes. Some of them are affected by missing values.

Table 1. Datasets, adopted from UCI and MOA.

Dataset	**Instances**	**Attribs**	**Types**	**Missing**	**Class**
Adult	48842	15	Mixed	Yes	2
Australian	690	15	Mixed	No	2
Breast	683	10	Numeric	No	2
Credit-g	1000	21	Mixed	No	2
Diabetes	768	9	Mixed	No	2
Ecoli	336	8	Numeric	No	8
German	1000	21	Mixed	No	2
Hypothyroid	3772	30	Mixed	Yes	4
Kr-vs-kp	3196	37	Numeric	No	2
Letter	20000	17	Mixed	No	26
Mushroom	8124	23	Nominal	Yes	2
Poker-lsn	829201	11	Mixed	No	10
Segment	2310	20	Numeric	No	7
Sick	3772	30	Mixed	Yes	2
Splice	3190	61	Nominal	No	3
Waveform5000	5000	41	Numeric	No	3

Apart from J48, the Counter strategy will be tested using the Weka implementations of Naive Bayes, jRip, Multi-Perceptron, and SMO as inductive algorithms. A 10-fold stratified cross-validation is run on each dataset, observing the average accuracy of the obtained models and the average percentage of original dataset used to induce the model, i.e., 100% means the full original dataset was used. All experiments were executed on a Intel Core i5-8300H at 2.3GHz, up to 3.9GHz with 8Gb DDR4. 8 distributed sites were simulated on this machine.

[1] https://github.com/xl666/jaca-ddm

In order to understand the performed sub-sampling, the following measures were used to compare the obtained window and the original dataset:

- The Kullback-Leibler divergence (D_{KL}) [4] is defined as:

$$D_{KL}(P||Q) = \sum_{x \in X} P(x) log_2 (\frac{P(x)}{Q(x)}),$$

  where $P(x)$ is the full dataset class distribution and $Q(x)$ the window class distribution. Instead of using a model to represent a conditional distribution of variables, as usual, we focus on the class distribution, computed as the marginal probability. Values closer to zero reflect higher similarity.
- Sim_1 [15] is a similarity measure between datasets defined as:

$$sim_1(D_i, D_j) = \frac{|Item(D_i) \cap Item(D_j)|}{|Item(D_i) \cup Item(D_j)|},$$

  where D_i is the window and D_j is the full dataset; and $Item(D)$ denotes the set of pairs attribute-value occurring in D. Values closer to one reflect higher similarity.
- Red [7] measures redundancy in a dataset in terms of conditional population entropy (CPE), defined as:

$$CPE = -\sum_{i=1}^{n_c} p(c_i) \sum_{a=1}^{n_a} \sum_{v=1}^{n_{v_a}} p(x_{a,v}|c_i) log_2 p(x_{a,v}|c_i),$$

  where n_c is the number of classes, n_a is the number of attributes, and n_{v_a} is the number of values for the attribute a. c_i stands for the $i-th$ class and $x_{a,v}$ represents the $v-th$ value of attribute a. CPE can be normalized [3] in such a way that values closer to zero reflect lower redundancy:

$$Red = 1 - \frac{CPE}{\sum_{a=1}^{n_a} log_2 n_{v_a}}.$$

3 Results

Figure 1 shows a strong negative correlation between the percentage of training instances used to induce the models and their accuracy, independently of the adopted inductive algorithm. This reproduces the results for J48 reported in literature [6] and corroborates that under Windowing, in general, the models with higher accuracy require less examples to be induced. However, accuracy is affected by the adopted inductive algorithm, e.g., Poker-lsn is approached very well by J48 (99.75 ± 0.07 of accuracy) requiring few examples (5% of the full dataset); while Naive Bayes is not quite successful in this case (60.02 ± 0.42 of accuracy) requiring more examples (59%). This behavior is also observed between jRip and MultiPerceptron for Hypothyroid; and between SMO and jRip for Waveform5000.

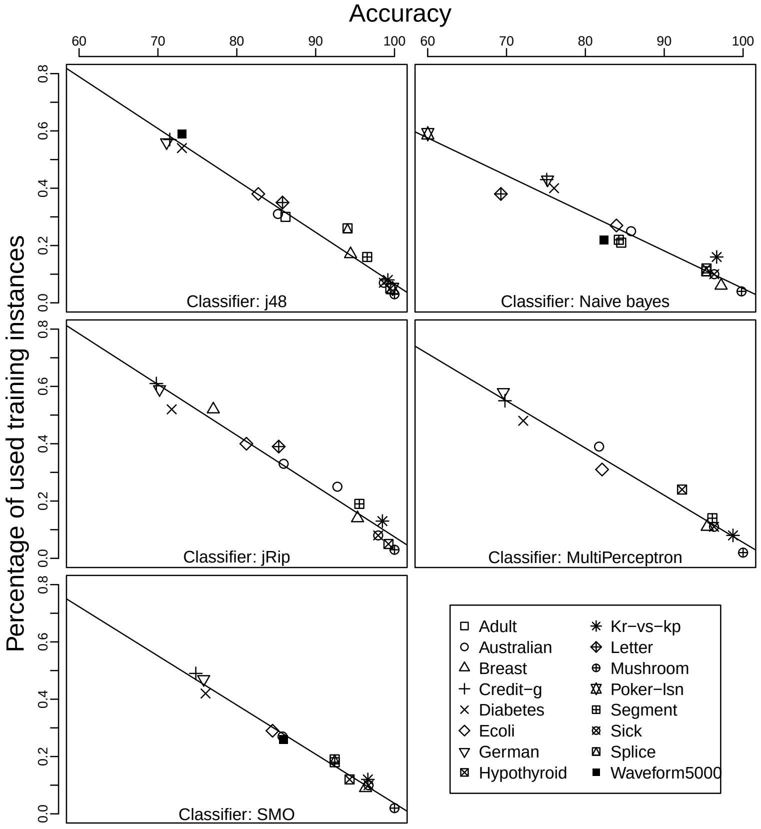


Fig. 1. Correlation between accuracy and percentage of used training examples. J48 = -0.98, NB = -0.96, jRip = -0.98, MP = -0.98 and SMO = -0.99.

Table 2 shows the accuracy results in detail while Table 3 show the number of used examples results, in terms of the percentage of the full dataset used for each inductive algorithm. Although not shown because of the available space, accuracies are comparable to those obtained without using Windowing, i.e., using the 100% of the available data to induce the models. Big datasets, as Adult, Letter, Poker-Isn, Splice, and Waveform5000 did not finish on reasonable time when using jRip, MultiPerceptron and SMO, with and without Windowing. In such cases, results are reported as not available (na). This might be solved by running the experiments in a real cluster of 8 nodes, instead of simulating the sites in a single machine, as done here, but it is not relevant for the purposes of this work.

The Kullback-Leibler divergence coefficient between the windows and the full datasets was close to zero in all cases ($D_{KL} < 0.25$), evidencing that the class distribution of the windows is very similar to that observed in the full datasets.

Table 2. Accuracies obtained from 10-fold cross validation (na = not available).

	J48	**NB**	**jRip**	**MP**	**SMO**
Adult	86.17 ± 0.55	84.54 ± 0.62	na	na	na
Australian	85.21 ± 4.77	85.79 ± 4.25	85.94 ± 3.93	81.74 ± 6.31	85.80 ± 4.77
Breast	94.42 ± 3.97	97.21 ± 2.34	95.31 ± 2.75	95.45 ± 3.14	96.33 ± 3.12
Credit-g	71.50 ± 5.81	75.10 ± 2.60	69.80 ± 3.71	69.80 ± 5.63	74.80 ± 5.98
Diabetes	73.03 ± 3.99	76.03 ± 4.33	71.74 ± 7.67	72.12 ± 4.00	76.04 ± 3.51
Ecoli	82.72 ± 6.81	83.93 ± 7.00	81.22 ± 6.63	82.12 ± 7.49	84.53 ± 4.11
German	71.10 ± 5.40	75.20 ± 2.82	70.20 ± 3.85	69.60 ± 4.84	75.80 ± 3.12
Hypothyroid	99.46 ± 0.17	95.36 ± 0.99	99.23 ± 0.48	92.26 ± 2.75	94.30 ± 0.53
Kr-vs-kp	99.15 ± 0.66	96.65 ± 0.84	98.46 ± 0.95	98.72 ± 0.54	96.62 ± 0.75
Letter	85.79 ± 1.24	69.28 ± 1.26	85.31 ± 1.06	na	na
Mushroom	100.00 ± 0.00	99.80 ± 0.16	100.00 ± 0.00	100.00 ± 0.00	100.0 ± 0.00
Poker-lsn	99.75 ± 0.07	60.02 ± 0.42	na	na	na
Segment	96.53 ± 1.47	84.24 ± 1.91	95.54 ± 1.55	96.10 ± 1.15	92.42 ± 1.87
Sick	98.64 ± 0.53	96.34 ± 1.44	97.93 ± 0.95	96.32 ± 1.04	96.71 ± 0.77
Splice	94.04 ± 0.79	95.32 ± 1.07	92.75 ± 2.11	na	92.41 ± 1.34
Waveform5000	73.06 ± 2.55	82.36 ± 1.64	77.02 ± 1.59	na	85.94 ± 1.32

Table 3. Percentage of the full dataset used for induction (na = not available).

	J48	**NB**	**jRip**	**MP**	**SMO**
Adult	0.30 ± 0.01	0.21 ± 0.00	na	na	na
Australian	0.31 ± 0.02	0.25 ± 0.01	0.33 ± 0.02	0.39 ± 0.04	0.27 ± 0.01
Breast	0.17 ± 0.01	0.06 ± 0.00	0.14 ± 0.01	0.11 ± 0.01	0.09 ± 0.01
Credit-g	0.57 ± 0.03	0.43 ± 0.01	0.61 ± 0.01	0.55 ± 0.04	0.49 ± 0.01
Diabetes	0.54 ± 0.05	0.40 ± 0.02	0.52 ± 0.04	0.48 ± 0.03	0.42 ± 0.02
Ecoli	0.38 ± 0.03	0.27 ± 0.01	0.40 ± 0.03	0.31 ± 0.03	0.29 ± 0.02
German	0.56 ± 0.04	0.43 ± 0.01	0.59 ± 0.02	0.58 ± 0.02	0.47 ± 0.02
Hypothyroid	0.05 ± 0.00	0.12 ± 0.01	0.05 ± 0.00	0.24 ± 0.01	0.12 ± 0.01
Kr-vs-kp	0.08 ± 0.01	0.16 ± 0.01	0.13 ± 0.00	0.08 ± 0.00	0.12 ± 0.00
Letter	0.35 ± 0.02	0.38 ± 0.00	0.39 ± 0.01	na	na
Mushroom	0.03 ± 0.00	0.04 ± 0.00	0.03 ± 0.00	0.02 ± 0.00	0.02 ± 0.00
Poker-lsn	0.05 ± 0.00	0.59 ± 0.00	na	na	na
Segment	0.16 ± 0.01	0.22 ± 0.01	0.19 ± 0.01	0.14 ± 0.01	0.18 ± 0.00
Sick	0.07 ± 0.00	0.10 ± 0.01	0.08 ± 0.00	0.11 ± 0.01	0.10 ± 0.00
Splice	0.26 ± 0.01	0.11 ± 0.00	0.25 ± 0.01	na	0.19 ± 0.00
Waveform5000	0.59 ± 0.02	0.22 ± 0.01	0.52 ± 0.00	na	0.26 ± 0.01

However it does not seem to be a correlation between this coefficient and the obtained accuracy, e.g., Mushroom has zero as divergence coefficient and 100% of accuracy, but Waveform5000 has similar divergence but considerable lower accuracy.

Table 4 shows the results for sim_1, suggesting that the windows for Australian, Breast, German, Letter, Kr-vs-Kp, and Poker-lsn conserve all the values

Table 4. Table of similarity measure sim_1 using the 10-folds cross-validation windows.

	j48	**NB**	**jRip**	**MP**	**SMO**
Adult	0.39±0.01	0.29±0.00	na	na	na
Australian	1.00±0.00	1.00±0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Breast	1.00±0.00	1.00±0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Credit-g	0.63±0.03	0.51±0.01	0.69 ± 0.01	0.63 ± 0.04	0.58 ± 0.01
Diabetes	0.73±0.04	0.63±0.02	0.72 ± 0.03	0.69 ± 0.02	0.64 ± 0.01
Ecoli	0.77±0.03	0.65±0.02	0.78 ± 0.02	0.69 ± 0.04	0.65 ± 0.03
German	1.00±0.00	1.00±0.00	1.00 ± 0.00	1.00 ± 0.00	0.99 ± 0.00
Hypothyroid	0.45±0.01	1.00±0.01	0.48 ± 0.01	0.68 ± 0.01	0.59 ± 0.01
Kr-vs-kp	1.00±0.01	0.97±0.01	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.00
Letter	0.99±0.01	0.99±0.01	0.98 ± 0.00	na	na
Mushroom	0.97±0.02	0.99±0.01	0.98 ± 0.00	0.97 ± 0.01	0.97 ± 0.01
Poker-lsn	1.00±0.00	1.00±0.00	na	na	na
Segment	0.28±0.01	0.32±0.01	0.31 ± 0.01	0.25 ± 0.01	0.28 ± 0.00
Sick	0.57±0.02	0.58±0.01	0.59 ± 0.01	0.60 ± 0.02	0.60 ± 0.01
Splice	0.97±0.04	0.96±0.05	0.97 ± 0.03	na	0.96 ± 0.04
Waveform5000	0.93±0.01	0.71±0.01	0.90 ± 0.00	na	0.76 ± 0.01

for their attributes observed in the full datasets; while Adult and Segment have problems achieving this. As in the previous case, this notion of similarity neither seems to correlate with the observed accuracy, e.g., Segment:

Red shows consistently the same values for the windows and the full datasets, meaning that both of them have very similar levels of redundancy. Given the nature of Windowing this can be a little bit surprising, since the window is expected to be less redundant than the full dataset because it does not include examples already covered by the induced models. But *Red* measures the information value given the information about the class values, an intrinsic property of the data set; while the redundancy reduction expected by Windowing is a property of a dataset given a classifier. This behavior of *Red*, reported in literature [3], suggests that a different measure for redundancy should be adopted.

4 Conclusions and Future Work

The correlation between the accuracy of the models obtained by Windowing and the number of examples used for this task was corroborated, independently of the adopted inductive algorithm, i.e., high accurate models require fewer examples to be learned. The metrics suggest that the windows have a class distribution very similar to the full datasets, as well as the same items (attribute-value pairs). They also have very similar intrinsic redundancy.

Unfortunately, such similarities are not enough to explain the success of the technique since they do not correlate with the obtained accuracy of the models.

Up to our knowledge, this is the first comparative study of Windowing in this respect. Future work requires finding a metric reflecting the notion of redundancy in terms of the set of covered examples to quantify the efficiency of Windowing as a sub-sampling method. Also, observing the evolution of the windows through the whole process seems pertinent to enhance our understanding of Windowing.

References

1. Bifet, A., Holmes, G., Kirkby, R., Pfahringer, B.: MOA: massive online analysis. J. Mach. Learn. Res. 11, 1601–1604 (2010), http://portal.acm.org/citation.cfm?id=1859903
2. Dua, D., Graff, C.: UCI machine learning repository (2017), http://archive.ics.uci.edu/ml
3. Fürnkranz, J.: Integrative windowing. Journal of Artificial Intelligence Research 8, 129–164 (1998)
4. Kullback, S., Leibler, R.A.: On information and sufficiency. The annals of mathematical statistics 22(1), 79–86 (1951)
5. Limón, X., Guerra-Hernández, A., Cruz-Ramírez, N., Acosta-Mesa, H.G., Grimaldo, F.: A windowing strategy for distributed data mining optimized through GPUs. Pattern Recognition Letters 93(Suplement C), 23–30 (July 2017)
6. Limón, X., Guerra-Hernández, A., Cruz-Ramírez, N., Grimaldo, F.: Modeling and implementing distributed data mining strategies in JaCa-DDM. Knowledge and Information Systems 60(1), 99–143 (2019)
7. Møller, M.: Supervised learning on large redundant training sets. International Journal of Neural Systems 4(1), 15–25 (1993)
8. Quinlan, J.R.: Learning logical definitions from relations. Machine Learning 5, 239–266 (1990)
9. Quinlan, J.R.: Induction over large data bases. Tech. Rep. STAN-CS-79-739, Computer Science Department, School of Humanities and Sciences, Stanford University, Stanford, CA, USA (May 1979)
10. Quinlan, J.R.: Learning efficient classification procedures and their application to chess en games. In: Michalski, R.S., Carbonell, J.G., Mitchell, T.M. (eds.) Machine Learning, vol. I, chap. 15, pp. 463 – 482. Morgan Kaufmann, San Francisco (CA) (1983), http://www.sciencedirect.com/science/article/pii/B9780080510545500194
11. Quinlan, J.R.: Induction of decision trees. Machine Learning 1, 81–106 (1986)
12. Quinlan, J.R.: C4. 5: programs for machine learning, vol. 1. Morgan kaufmann, San Mateo, CA., USA (1993)
13. Quinlan, J.R.: Improved use of continuous attributes in c4.5. Journal of Artificial Intelligence Research 4, 77–90 (1996)
14. Witten, I.H., Frank, E., Hall, M.A.: Data Mining: Practical Machine Learning Toools and Techniques. Morgan Kaufmann Publishers, Burlington, MA., USA (2011)
15. Zhang, S., Zhang, C., Wu, X.: Knowledge Discovery in Multiple Databases. Advanced Information and Knowledge Processing, Springer-Verlag London, Limited, London, UK (2004)

ISSN 1870-4069

Towards a Transfer Learning Strategy in Full Model Selection Algorithm for Temporal Data Mining

Nancy Pérez Castro[1], Héctor Gabriel Acosta Mesa[2]

[1]University of Papaloapan, Engineering and Technology,
México

[2]University of Veracruz, Artificial Intelligence Research Center,
México

nperez@unpa.edu.mx, heacosta@uv.mx

Abstract. Traditional machine learning techniques were designed for training for scratch depend on the current feature-space distribution. In many real applications, the fact to obtain new data for training and rebuilds models could become expensive or impossible. Therefore, from a lifelong machine learning conceptualization, transfer learning can be indeed beneficial to speed up the time it takes to develop and train a model by reusing an isolated pre-training setting as a starting point for another target domain, especially when multiple tasks and hyper-parameter optimization are considered, such as a full model selection approach. This document presents an early transfer learning strategy based on a decision tree powered by full models for temporal databases trained in an isolated way with different search methods. The proposed transfer learning strategy is capable to suggesting the starting point and the search method adopted by the full model selection approach.

Keywords: transfer learning, full model selection, temporal databases.

1 Introduction

Humans have the innate capacity to transfer knowledge across tasks. In this regard, the acquired experience can be utilized in the same way to solve related tasks. Therefore, the more connected tasks, the easier it is to transfer or cross-utilize the knowledge. Concerning Computer Science, particularly, Data Mining (DM) and Machine Learning (ML) fields, have been inspired by how human beings learn and transfer knowledge to simulate those behaviors through algorithms.

However, although significant progress in knowledge engineering in both DM and ML algorithms has achieved, most of them have been traditionally designed to work in isolation. The isolated training means that the built models are focused on specific tasks and depend on the current feature-space distribution.

pp. 65–73; rec. 2019-09-15; acc. 2019-09-30

Therefore, if the distribution changes, the models need to be rebuilt from scratch using newly collected training data.

In many real applications, the fact to obtain new data for training and rebuilds models could become expensive or impossible. In order to overcome these issues, researchers and scientists turn the gaze toward knowledge transfer or transfer learning, whose purpose is centered on the need for lifelong machine-learning methods that retain and reuse previously learned knowledge. According to Goodfellow et al. [5], transfer learning is described as *the situation where what has been learned in one setting is exploited to improve generalization in another environment.*

Research on transfer learning has attracted more attention in the last decades in different ways such as life-long learning, multi-task learning, knowledge transfer, inductive transfer, knowledge consolidation, context-sensitive learning, knowledge-based inductive bias, meta-learning, incremental/cumulative learning, and recently Auto-machine learning (AutoML) or Full Model Selection.

The specialized literature on learning transfer highlights three important research issues:

- What to transfer?
- How to transfer?
- When to transfer?

Regarding these issues, Pan and Yang [8], suggest a classification of transfer learning approaches according to three sub-settings: a) *Inductive transfer learning*, the target task is different from the source task, while the source and target domains can be the same or not. b) *Transductive transfer learning*, the source and target tasks are the same, while the source and target domains could be the same. c) *Unsupervised transfer learning*, the target task is different but related to the source task focus on unsupervised learning tasks in the target domain.

In related literature, it has been observed that in transfer learning approaches, especially inductive transfer learning, it is possible to transfer instances, feature representation, parameters, and relational-knowledge. Regarding related works of transfer learning, most of the research has been developed within the framework of artificial neural networks where multi-task can be involved [10].

In the context of the Full Model Selection (FMS) problem, where multiple task and hyper-parameters optimization are involved, the transfer learning has not been explored. Since one of the benefits of transfer learning is that it can speed up the time it takes to develop and train a model by reusing these settings as a starting point for another scene. The transfer learning strategy turns out to be an attractive option to accelerate the search for complete models.

In this regard, an early proposal of transfer learning in the frame of an FMS algorithm for temporal data mining tasks is presented. The remaining sections of this document are organized as follows. In Section 2, a brief theoretical background of FMS in temporal data is given. In Section 3, the employed methodology is described. Then, in Section 4, the preliminary results are outlined. Finally, Section 5 presents conclusions and future work.

2 Background

2.1 Full Model Selection Problem

Full Model Selection problem (FMS) refers to all aspects of automating the machine learning process, including model selection and hyper-parameter optimization for carrying on different tasks in an incremental way. In order to produce a suitable combination of methods which help to classify or predict a new data within a fixed computational cost, FMS can involve two remarkable concerns: (1) no single method performs well on each dataset, and (2) some methods work appropriately base on its hyper-parameter optimization. Both issues are known as Algorithm Selection and Model Selection problems [3, 11].

This work tackles the FMS problem in temporal databases, mainly in time-series, as a single-objective optimization problem through an evolutionary wrapper approach, where population-based metaheuristics or single point-search metaheuristics can be used [7, 2, 9]. An instance of FMS problem for temporal data consists of finding a suitable combination of smoothing, time-series representation, instances reduction, and classification methods with the setting of their related hyper-parameters. The FMS problem is expressed in Equation 1, where a set of algorithms $\mathcal{A} = A^1, ..., A^n$ with their related hyper-parameters $\theta = \{\theta_1, ..., \theta_m\}$ and labeled training data $\mathcal{D} = \{(x_1, y_1), ..., (x_n, y_n)\}$ are used to find the optimal generalized performance, for which, the training data is split up into disjoint training $\mathcal{D}^i_{train}$ and validation $\mathcal{D}^i_{valid}$ datasets which are evaluated by loss function $\mathcal{L}$ in an isolated training through k-cross-validation method:

$$\mathcal{A}^*, \boldsymbol{\theta}_* \in \operatorname*{arg\,min}_{A^J \in A, \theta \in A^j} \sum_{i=1}^{K} \mathcal{L}(\mathcal{A}^j_\theta, D^i_{train}, D^i_{valid}). \tag{1}$$

One of the advantages of solving the FMS problem by evolutionary wrapper is the capacity of manipulate multiple task and the hyper-parameter optimization at the same time. However, the main disadvantages of this approach are the high computational cost during the isolate training and the absence of reusing trained models for other data domains. Therefore, to treat those drawbacks, and inspired by lifelong machine learning (LML) paradigm [12], a set of experiments to find a strategy of transfer learning within the framework of the FMS algorithm is carried out.

3 Methodology

In this section, the adopted general methodology is described and presented in Figure 1. The considered instance space $\mathcal{X}$ is a set of time-series databases, taken of a well-known benchmark [6]. The considered FMS approach is widely described in [9], and general behavior is presented in Algorithm 1. This approach can be works under two different metaheuristics structures, based-population or a single point optimizer [1, 13].

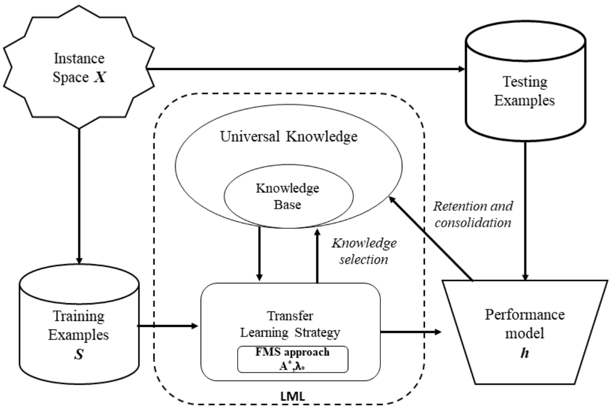


Fig. 1. Graphical representation of interaction between specific instance space, FMS approach and transfer learning strategy under LML paradigm.

The based-population structure is guided by μ-Differential Evolution algorithm (four variants are evaluated) while the single point search operates under local search (two different solution encoding are considered). Candidate solutions are composed of a combination of methods for smoothing, time-series representation, instance selection and classification with associated hyper-parameters. The original FMS approach was designed to train in an isolated way where all candidate solutions are evaluated according to the cross-validated miss classification rate, depending on the available database.

At the end of the evolutionary process, the best solution is obtained for each database. In order to build a strategy to transfer learning under LML paradigm, a knowledge base is needed, that a long term will be interacting with universal knowledge. In this work, the knowledge base is powered by a decision tree building from the best full models obtained during the isolated training of the FMS algorithm in its different versions per each database. Concerning the LML framework, the FMS algorithm must use the decision tree to determine the starting point for the training stage of a new database, as well as the recommended metaheuristic. Therefore, with the transfer learning strategy based on a decision tree, the cost to generate full models for new instances of temporal databases is expected to be lower than training from scratch. So far, the retention and consolidation are not considered in this early proposal.

4 Experiment and Results

This section presents a set of experiments realized to build and evaluate the early transfer learning strategy for FMS algorithm. The experimentation is presented in two stages: (1) A comparison of the final statistical results of variants of FMS

Algorithm 1 General behavior of FMS algorithm

```
Require: A (Pool of available methods), $\theta$ (Set of involved hyper-parameters), $M$ (Metaheuristic
    as optimizer), $fitness$ (Fitness Function), $D_{train}$ (Train dataset), $D_{valid}$ (Validation dataset),
    $D_{test}$ (Test dataset),
Set $maxItera$ % Maximum number of iterations
Set $i = 0$
Set $M$ % which can be population-based or single point optimizer
Randomly generate initial solution(s) % if $M$ is population-based a set of solutions are generated,
    other side one initial solution is generated
while $i < maxItera$ do
    Starts optimization process through population-based or single point search
    A fitness function is used for evaluating
    Special operators are involved (crossover, mutation, selection or neighborhood generator)
end while
Get best final solution $\vec{s}$ % involves a suitable combination of methods and their hyper-
    parameters
Evaluate $\vec{s}$ on $D_{test}$
```

algorithm to build a based decision tree knowledge base and (2) preliminary results of the transfer learning strategy.

4.1 Stage 1: Knowledge Base Building

Six versions of FMS algorithm were trained in an isolated environment where eight temporal databases (Table 1) were used. The four firsts versions of FMS algorithm correspond to population-based option while the two rests are compatible with the single-point search option. In all cases, the termination condition was 3000 evaluations, and five independent runs were carried out. The configuration used by each involved metaheuristics is described as following, based on [2, 14]:

P-DEMS versions: $iter = 500$, $NP = 6$, $CR = 0.1$, $F = 0.9$, $N = 2$ and $R = 10$.
S-LSMS versions: $iter = 500$ and $N_k = 6$.

Table 3 shows the final numerical results obtained by the six FMS version training in a isolated way. The population-based versions were known as P-DEMS1 to P-DEMS4 and the single point search versions were named as S-LSMS1 and S-LSM2. The reported values correspond to the average of five trials evaluated in the testing set of each database. A non-parametric Friedman test was used [4] for multiple comparison among FMS versions. Friedman test converts numerical values to ranks. Thus, it ranks the FMS versions for each problem separately, the best performing algorithm version should have rank 1, the second-best rank 2, etc. When ties are presented, like this case, average ranks are computed.

According to the average ranks, it is observed that SLSMS1 and PDEMS1 were the two best versions, followed by P-DEMS2, S-LSMS2, P-DEMS4 and P-DEMS3, respectively. From this, information a simple log-archive was created. The log-archive contains information related to each database's information (time-series length, number of classes, domain), the full best model obtained by isolated training of the six versions of FMS algorithm for each database, the

Table 1. Time-series databases description.

No.	Name	No. of classes	Training set size	Testing set size	Time-series length	Domain
1.	Beef	5	30	30	470	Spectro
2.	CBF	3	30	900	128	Simulated
3.	Coffee	2	28	28	286	Spectro
4.	ECG200	2	100	100	96	ECG
5.	FaceFour	4	24	88	350	Image
6.	Gun_Point	2	50	150	150	Motion
7.	Lightning-2	2	60	61	637	Sensor
8.	Lightning-7	7	70	73	319	Sensor
9.	OliveOil	4	30	30	570	Spectro
10.	Trace	4	100	100	275	Sensor

Table 2. Description of knowledge base attributes.

Attribute	**Description**	Type
length	Time-series length.	Numeric
classes	Number of classes.	Numeric
smooth	Type of selected smoothing method.	Nominal
representation	Type of selected time-series representation method.	Nominal
insReduc	Type of selected instance selection method.	Nominal
error	Misclassification rate of the tested model.	Numeric
Meta	Version of FMS algorithm.	Nominal

average runtime during isolated training for each database and FMS algorithm, the test misclassification rate of each full model and the name of FMS algorithm.

From this, information a simple log-archive was created. The log-archive contains information related to each database's information (time-series length, number of classes, domain), the full best model obtained by isolated training of the six versions of FMS algorithm for each database, the average runtime during isolated training for each database and FMS algorithm, the test misclassification rate of each full model and the name of FMS algorithm. A total of 300 models were stored that gave rise to form a supervised knowledge database, where the name of the FMS algorithm was considered as the class attribute.

Because only two versions of FMS algorithm reported competitive results, the knowledge database was limited to only store the models of these versions. Then, the knowledge base was composed of seven attributes, detailed in Table 2, with 100 different models. A decision tree of Weka was selected to generate a practical and visual way that supports the rules generation that can be incorporated as part of the learning transfer strategy. The accuracy of the decision tree was of 83.10%, and it is presented in Figure 2.

4.2 Stage 2: Adoption and Testing of the Learning Strategy

According to the taxonomy of learning transfer approaches, the proposed strategy, in this work, is classified as *Transductive transfer learning*, because of the

Table 3. Comparison of averaging performance among the six metaheuristics for each database. Values to the right of ± represent the standard deviation and the values in parentheses represent the ranks computed by the Friedman test. Values in **boldface** mean the lowest values found or the best ranking.

Database	P-DEMS1	P-DEMS2	P-DEMS3	P-DEMS4	S-LSMS1	S-LSMS2
Beef	0.053±0.102 (3)	0.087±0.038 (4)	**0.000±0.000 (1.5)**	0.160±0.060 (5)	**0.000±0.000 (1.5)**	0.367±0.227 (6)
CBF	**0.000±0.000 (3)**	**0.000±0.000 (3)**	**0.000±0.000 (3)**	**0.000±0.000 (3)**	**0.000±0.000 (3)**	0.030±0.027 (6)
Coffee	**0.000±0.000 (3)**	**0.000±0.000 (3)**	**0.000±0.000 (3)**	**0.000±0.000 (3)**	0.268±0.157 (6)	0.000+0.000 (3)
ECG200	**0.000±0.000 (2)**	0.800±0.447 (4)	1.000±0.000 (5.5)	1.000±0.000 (5.5)	**0.000±0.000 (2)**	**0.000±0.000 (2)**
FaceFour	**0.000±0.000 (3.5)**	**0.000±0.000 (3.5)**	**0.000±0.000 (3.5)**	**0.000±0.000 (3.5)**	**0.000±0.000 (3.5)**	**0.000±0.000 (3.5)**
Gun_Point	**0.000±0.000 (1)**	0.395±0.221 (4)	0.493±0.000 (5.5)	0.493±0.000 (5.5)	0.388±0.217 (3)	0.212±0.253 (2)
Lightning-2	**0.000±0.000 (3)**	**0.000±0.000 (3)**	**0.000±0.000 (3)**	**0.000±0.000 (3)**	**0.000±0.000 (3)**	0.069±0.154 (6)
Lightning-7	0.766±0.035 (5)	0.762±0.028 (4)	0.786±0.019 (6)	0.761±0.012 (3)	**0.019±0.019 (1)**	0.082±0.046 (2)
OliveOil	0.013±0.030 (2.5)	0.033±0.047 (5)	0.027±0.043 (4)	0.013±0.018 (2.5)	**0.000±0.000 (1)**	0.133±0.122 (6)
Trace	0.800±0.447 (3.5)	0.800±0.447 (3.5)	1.000±0.000 (5.5)	1.000±0.000 (5.5)	**0.000±0.000 (1.5)**	**0.000±0.000 (1.5)**
Average rank	2.950	3.700	4.050	3.950	**2.550**	3.800

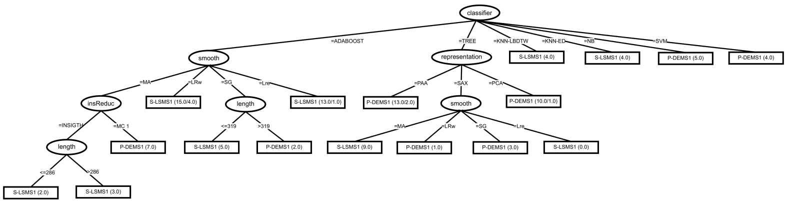


Fig. 2. Decision tree of knowledge base.

source and target tasks are the same, while the source and target domains could be the same or not. Regarding the three principal questions on *what*(Q1), *how*(Q2)and *when*(Q3) to transfer, these will be described below:
Q1: Setting of full models that includes selected methods and their hyper-parameters optimized. Besides the suggested search engine for continue the training process.
Q2: The pre-trained models that will be the starting point for another dataset will be randomly selected within the knowledge base, as long as the pre-trained models have been used in smaller time series or of the same length as the new database. A set of six different models can be selected as randomly, which will be evaluated by the decision tree. The tree will suggest a class tag for each instance, which corresponds to the type of search engine. Considering the majority vote, If the population-based option (P-DEMS1) is suggested, all models are transferred. Otherside, if the single point optimizer (S-LSMS1) is suggested, only one of the six can be assigned.
Q3: At the beginning of the training process of a new data set that not exists in the knowledge base.

The proposed transfer strategy for FMS algorithm was tested on four databases, the preliminary results are shown in Table 4. Similar behavior was obtained by LML-FMS in two of the four databases against isolated training. The suggested search strategy for theses cases was S-LMS1. Otherside, the significantly worse cases were produced by the P-DEMS1 search engine. An improvement speed up on training was observed when P-DEMS1 was suggested as a search engine.

Table 4. Comparison between the isolated training and proposed approach by transfer learning. IT means isolated training, while LML-FMS means lifelong machine learning for the full model selection. Values in **boldface** mean the significant lowest values.

No.	Name	Classes	TS-Length	.6 Domain	.6 IT	.6 LML-FMS
1.	ECGFiveDays	2	136	ECG	**0.0000**	0.0011
2.	SonyAIBORobotSurface	2	70	SENSOR	0.0032	0.0080
3.	SonyAIBORobotSurfaceII	2	65	SENSOR	**0.0031**	0.0183
4.	ItalyPowerDemand	2	24	SENSOR	0.0264	0.0255

5 Conclusions and Future Work

In this paper, a transfer learning strategy for the FMS algorithm for temporal data mining was presented. The initial knowledge base was built from isolated pre-trained full models, and the transfer learning is based on a decision tree powered by that base. Although isolated training provides better solutions in two of the databases, preliminary results through transfer learning show competitive results, encouraging to extend experiments in other database domains. Therefore, as future work, data complexity measures, test data distribution, or model complexity could be considered into the knowledge base. Besides, to explore other ways to transfer the learning between different temporal domains data.

References

1. Caraffini, F., Neri, F., Poikolainen, I.: Micro-differential evolution with extra moves along the axes. In: Differential Evolution (SDE), 2013 IEEE Symposium on. pp. 46–53 (April 2013)
2. Escalante, H.J., Montes, M., Sucar, L.E.: Particle swarm model selection. Journal of Machine Learning Research 10(Feb), 405–440 (2009)
3. Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., Hutter, F.: Efficient and robust automated machine learning. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems 28, pp. 2962–2970. Curran Associates, Inc. (2015)
4. García, S., Fernández, A., Luengo, J., Herrera, F.: Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Information Sciences 180(10), 2044 – 2064 (2010), special Issue on Intelligent Distributed Information Systems
5. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016), http://www.deeplearningbook.org
6. Keogh, E., Zhu, Q., Hu, B., Hao, Y., Xi, X., Wei, L., Ratanamahatana, C.A.: The UCR Time Series Classification/Clustering Homepage (2011), www.cs.ucr.edu/ẽamonn/time_series_data/
7. Momma, M., Bennett, K.P.: A pattern search method for model selection of support vector regression. In: Proceedings of the 2002 SIAM International Conference on Data Mining. pp. 261–274. SIAM (2002)
8. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. on Knowl. and Data Eng. 22(10), 1345–1359 (Oct 2010)

9. Pérez-Castro, N., Acosta-Mesa, H.G., Mezura-Montes, E., Cruz-Ramírez, N.: Towards the full model selection in temporal databases by using micro-differential evolution. an empirical study. In: 2015 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC). pp. 1–6 (Nov 2015)
10. Reddy, T.K., Arora, V., Kumar, S., Behera, L., Wang, Y., Lin, C.: Electroencephalogram based reaction time prediction with differential phase synchrony representations using co-operative multi-task deep neural networks. IEEE Transactions on Emerging Topics in Computational Intelligence 3(5), 369–379 (Oct 2019)
11. Rice, J.R.: The algorithm selection problem. In: Rubinoff, M., Yovits, M.C. (eds.) Advances in computers, vol. 15, pp. 65 – 118. Elsevier (1976)
12. Silver, D.L., Yang, Q., Li, L.: Lifelong machine learning systems: Beyond learning algorithms. In: AAAI Spring Symposium: Lifelong Machine Learning. AAAI Technical Report, vol. SS-13-05. AAAI (2013)
13. Talbi, E.: Metaheuristics: From Design to Implementation. Wiley Series on Parallel and Distributed Computing, Wiley (2009)
14. Viveros-Jiménez, F., Mezura-Montes, E., et al.: Empirical analysis of a micro-evolutionary algorithm for numerical optimization. International Journal of Physical Sciences 7(8), 1235–1258 (2012)

ISSN 1870-4069

Discriminative Parameter Learning of Bayesian Networks Using Differential Evolution: A Preliminary Analysis

Alejandro Platas López, Nicandro Cruz Ramírez, Efrén Mezura Montes,
Alejandro Guerra Hernández

Universidad Veracruzana, Centro de Investigación en Inteligencia Artificial,
México

alejandroplatasl@gmail.com,{ncruz,emezura,aguerra}@uv.mx

Abstract. This work proposes Differential Evolution (DE) to train parameters of Bayesian Networks (BN) for optimizing the Conditional Log-Likelihood (Discriminative Learning) instead of the log-likelihood (Generative Learning). Although Discriminative Parameter Learning algorithms have been proposed, to the best of the authors' knowledge, a metaheuristic approach has not been devised yet. Thus, the objective of this research is to come up with this kind of solution and evaluate its behavior so that its feasibility in this domain can be determined. According to the theory such a solution tends to generate low-bias classifiers that minimize classification error but this is not reflected in results, regarding proposed method, bias in search for best solutions improves DEs performance.

Keywords: Bayesian networks, differential evolution, discriminative parameter learning.

1 Introduction

Two paradigms are distinguished for parameter learning of Bayesian networks. One of them, called Generative Learning (GL), optimizes Log-Likelihood in order to obtain the parameters that characterize the joint distribution in the form of local conditional distributions, and subsequently estimates class conditional probabilities using the Bayes rule. Even though this paradigm is computationally efficient, it is likely to generate biased classifiers [12].

The other paradigm optimizes Conditional Log-Likelihood (CLL) to directly estimate the parameters associated with conditional class distribution. Such paradigm is known as Discriminative Learning (DL) and generates low-bias classifiers that typically tend to minimize the classification error. In addition, the effect caused by the assumption of conditional independence among attributes in the network structure, but which may be violated in the data, is reduced.

pp. 75–82; rec. 2019-09-17; acc. 2019-09-30

However, the huge search space defined by the parameters that optimize the CLL function motivates this work to find efficient and effective search algorithms for discriminative parameter learning in BN classifiers [1,9].

Based on the above, different algorithms have been developed with the purpose of generating unbiased classifiers that mitigate the assumption of conditional independence among attributes. To the best of the authors' knowledge, there are nor proposals that apply evolutionary algorithms for DL of parameters. In this paper, we propose the use of the Differential Evolution (DE) algorithm for learning parameters in BN optimizing CLL. The aim is to understand the behavior of this evolutionary algorithm in this particular optimization task in both optimized structures for classification purposes, learned with a Bi-Objective PSO [2] and structures that are not optimized, learned by Tree-Augmented Network [3]. A comparison is made against some parameter learning algorithms for Log-Likelihood optimization.

The rest of the paper is organized as follows. Section 2 describes the optimization problem and introduces notations and terminologies. Section 3 gives details about the implementation of algorithms and experimental settings. The obtained results are presented in section 4. Finally, some conclusions and possible paths of future work are given in section 5.

2 Parameter Learning

GL is based on two steps, the first involves the maximization of $P(y, \mathbf{x})$, where y is the class and $\mathbf{x}$ is the set of attributes; and the second step is the application of the Bayes rule to obtain $P(y|\mathbf{x})$. In DL, it is possible to directly optimize $P(y|\mathbf{x})$, maximizing CLL.

Although there are approaches for parameter learning (not structures) with a discriminative approach [4]-[12], no meta-heuristic algorithms for DL of parameters in BNs have been adopted. A related work was proposed by [13], where they optimize LL (Generative approach) with a Genetic Algorithm combined with Expectation Maximization (GAEM). This proposal, according to the authors, combines the global search and local search properties of the respective algorithms. Part of notation and definitions used throughout this paper are taken from that work.

The proposed methods in this paper is based on Differential Evolution, which has been used for optimization problems in real-world applications[14]. DE was introduced in 1996 [15], and improved with some mechanisms to decrease the dependence to its parameter values such as the mutation factor F and the crossover rate CR [14], so as to increase its search performance[16].

To determine which search strategies provide a better performance, four DE variants will be used in this study: DE/rand/1/bin [15], JADE without archive, JADE with archive [14] and L-SHADE [16]. Such variant selection was made to include the most popular DE variant (DE/rand/1/bin), a variant with a novel differential mutation operator (JADE) and a recent one with a memory-based parameter adaptation mechanism (L-SHADE).

Let $\mathbf{X} = \{X_1, X_2, \ldots, X_R\}$ denote the set of random variables in a BN. Each random variable X_k is associated with a Conditional Probability Table (CPT). An individual ρ^t consists of a random variables vector of CPTs in a BN. The estimated CPT of an individual i at generation t is denoted by Θ_i^t. An individual is defined as a vector consisting of CPTs: $\rho_i^t = (\Theta_1^t, \Theta_2^t, \ldots, \Theta_R^t)$.

A CPT is generated based on the constraint that the sum of probabilities for different states of the random variable should be equal to 1 for a parent instantiation. A CPT is given by: $\Theta_i^t = (\theta_{1,1}^t, \ldots, \theta_{1,b}^t, \ldots, \theta_{a,1}^t, \ldots, \theta_{a,b}^t)$, where $\theta_{ab}^t \in [0,1]$ denotes a probability value for a particular state given a parent instantiation.

3 Implementation

The obtained performance by the DE variants was compared based on both CLL optimization and predictive accuracy. Such results were further contrasted against those obtained by three GL algorithms: (1) Bayesian estimation, (2) maximum-likelihood and (3) Attribute-Weighted Naive Bayes. The parameter learning was applied to (1) BN structures optimized for classification with a bi-objective PSO algorithm that seeks trade-offs between predictive power and compression of data with the MDL metric [2]; the solution found in the "knee" of the Pareto front was selected as the best BN structure, and (2) BN structures learned with TAN-CL[1]. The datasets shown in Table 1 were used for comparison purposes and predictive accuracy was tested with 15 rounds of 2-fold stratified cross validation. 2-fold cross validation is used in order to maximize the variation in the training data from trial to trial [12].

Table 1. Details of datasets. Abrev: Abbreviation. Class = Number of classes. Att: Number of attributes. Case: Number of cases. θs: Number of parameters to be optimzed.

Data	Abbrev	Class	Att	Case	θs	Data	Abbrev	Class	Att	Case	θs
australian	aust	2	15	690	130	hepatitis	hepa	2	20	80	162
chess	ches	2	37	3296	290	lymphography	lymp	4	19	148	1220
cleve	clev	2	12	296	1005	Mofn-3-10	mofn	2	11	1324	78
corral	corr	2	7	128	46	pima	pima	2	9	768	102
crx	crx	2	16	653	848	segment	segm	7	20	2310	2548
diabetes	diab	2	9	768	102	Soybean-large	soyb	19	36	316	5265
flare	flar	8	11	1389	276	Tic-tac-toe	tic-	2	10	958	152
german	germ	2	21	1000	866	vehicle	vehi	4	19	958	1152
glass2	glas	2	10	163	1038	vote	vote	2	18	436	278
heart	hear	2	14	270	118	Waveform-21	wave	3	22	301	3186

Two repairs were applied to satisfy the constraints for $\theta_{ab}^t \in [0,1]$:

$$\theta_{ab}^{'} = \begin{cases} |\theta_{ab}| \mod 1 & \text{if} \quad \theta_{ab} < 0 \\ 1 - (\theta_{ab} \mod 1) & \text{if} \quad \theta_{ab} > 1, \end{cases}$$

and to keep the sum of row vectors equal to 1:

$$\theta^{'}_{ab} = \theta_{ab} \Big/ \sum_{b=1}^{n} \theta_{ab}.$$

For the DE variants, 31 independent runs were performed on each dataset. The parameter values used in each DE variant are detailed in Table 2. Such values were adopted from the specialized literature [14] and by further experimentation.

Table 2. Parameter values of DE variants

DE algorithm	**NP**	**G**	**F**	**CR**	**c**	**p**	**|A|**
rand/1/bin	200	25 × Att	0.5	0.7			
JADE without A	200	10 × Att			0.05	0.05	∅
JADE with A	200	10 × Att			0.05	0.05	NP
L-SHADE	200	10 × Att			0.05	0.05	NP_g

4 Results

Based on the results summarized in Fig. 1, in datasets with a few number of parameters θ, the DE variants provided better results than those of the GL algorithms. Such behaviour was less marked in complex BN. Graphically there is no difference among structure types. As expected, those algorithms that had CLL as objective function, gave better results (Fig. 2). Regarding predictive accuracy, there is no clear evidence in favor of any approach, although DE variants are not the best, as shown in Figs. 3 and 4.

5 Conclusion and Future Work

A comparison of representative DE variants in an open problem about discriminative learning of parameters in BNs was presented. This would lead to the generation of classifiers with low bias that minimize the classification error. Based on the results obtained, difficulties were noted for DE variants when the number of parameters θ to be optimized increased. On the other hand, it was also observed that bias in search for high-quality solutions as well as the reduction in population size improved the DE variants performance. Future work contemplates the application of strategies that are capable of contending with big networks. Although it was not the purpose of this research, it is important to evaluate the performance of the proposed DE variants against state-of-art discriminative learning algorithms.

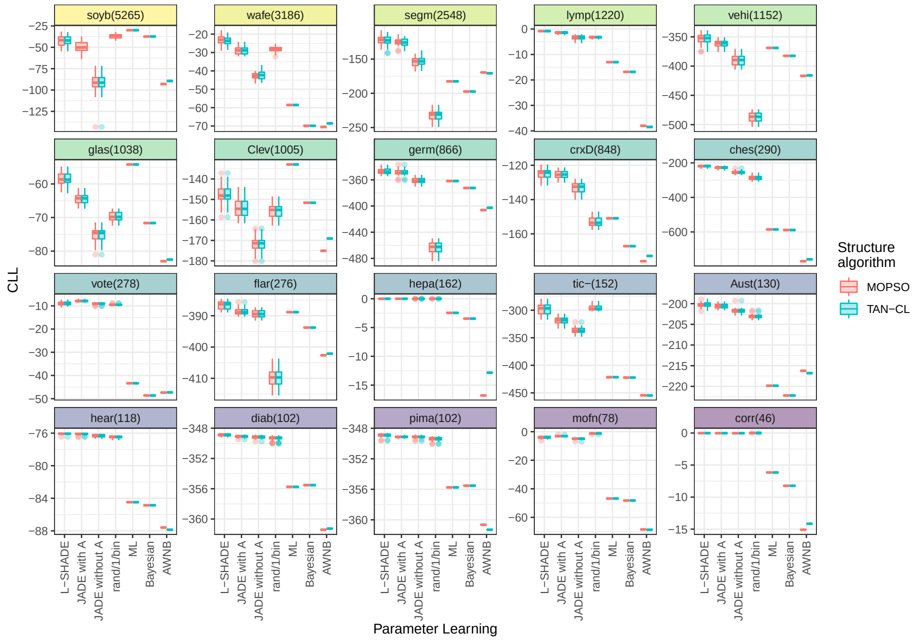


Fig. 1. Best CLL obtained in 31 independent runs by the DE variants and CLL obtained by the GL algorithms. Number of parameters θ are shown in parentheses.

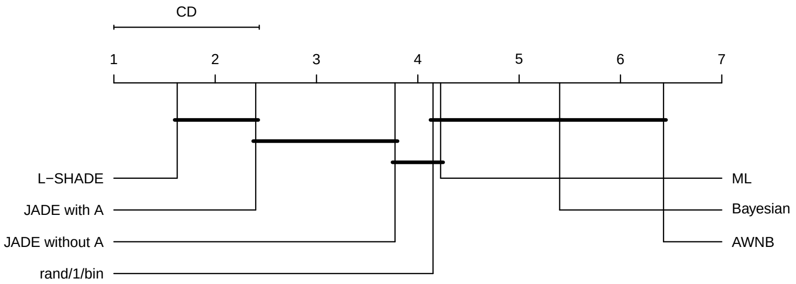


Fig. 2. Critical Differences diagram for the median CLL value of 31 independent runs (DE variants) and CLL value (GL algorithms). Horizontal line segments group together algorithms with CLL that are not significantly different (at $\alpha = 0.05$). Top line axis ranks methods from best (left) to worst (right).

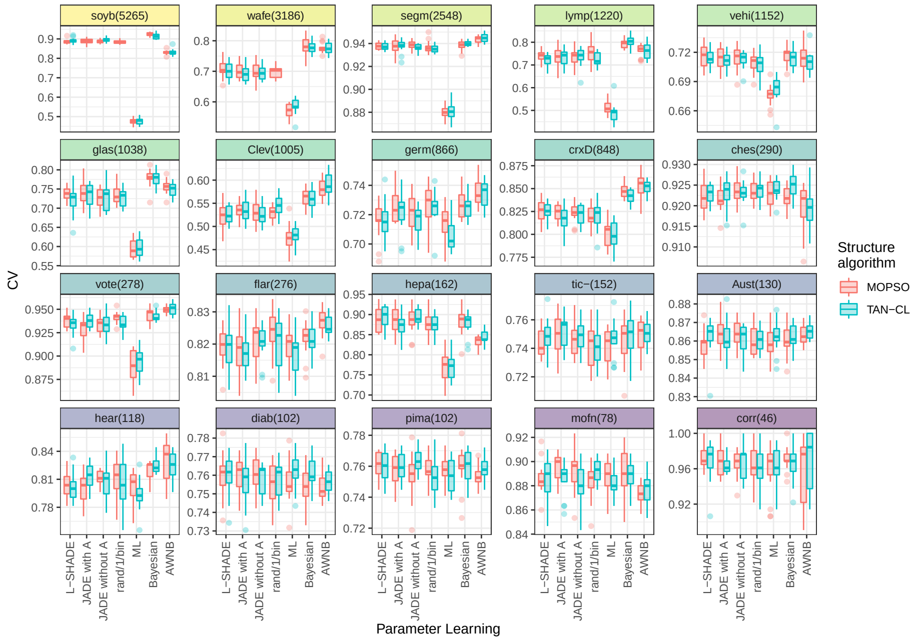


Fig. 3. Predictive accuracy of 15 rounds of 2-fold CV with the parameters learned by median of best solutions among 31 independent runs by DE variants and solution of GL algorithms. Datasets are sorted by number of parameters θ (shown in parentheses).

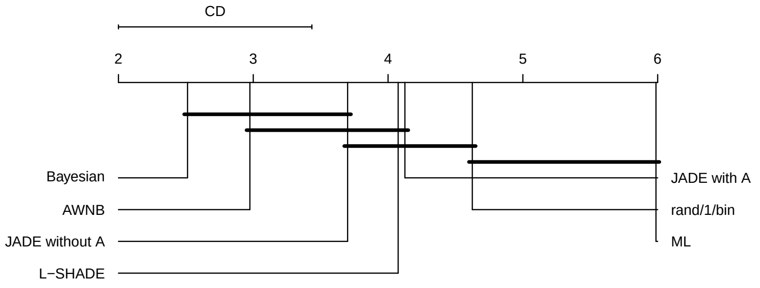


Fig. 4. Critical Differences diagram for median predictive accuracy of 15 rounds of 2-fold CV among algorithms. Horizontal line segments group together algorithms with predictive accuracy that are not significantly different (at $\alpha = 0.05$). Top line axis ranks methods from best (left) to worst (right).

References

1. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Machine Learning, 29:2,131–163. (1997)
2. Aguilera-Rueda, V.J., Cruz-Ramírez, N., Mezura-Montes, E., Vilalta, R.: Learning bi-objective Bayesian Networks Structure from data using Particle Swarm Optimization. Elsevier (forthcoming)
3. Chow, C.K., Liu, C.N.: Approximating discrete probability distributions with dependence trees. IEEE Transactions on Information Theory, 14(3), pp. 462–467. (1968)
4. Shen, B., Su, X., Greiner, R., Musilek, P., Cheng, C.: Discriminative parameter learning of general Bayesian network classifiers. In: 15th IEEE International Conference on Tools with Artificial Intelligence, pp. 296–305. (2003)
5. Grossman, D., Domingos, P.: Learning Bayesian Network classifiers by maximizing conditional likelihood. In R. Greiner, and D. Schuurmans (Eds.), 21st International Conference on Machine Learning, ICML. pp. 361–368. (2004)
6. Guo, Y., Greiner, R.: Discriminative Model Selection for Belief Net Structures. In: Proceedings of the National Conference on Artificial Intelligence. 2. pp. 770–776. (2005)
7. Jing, Y., Pavlović, V., Rehg, J.M.: Efficient discriminative learning of Bayesian network classifier via boosted augmented naive Bayes. In: Proceedings of the 22nd international conference on Machine learning. ACM, New York, NY, USA, pp. 369–376. (2005)
8. Greiner, R., Su, X., Shen, B., Zhou, W.: Structural Extension to Logistic Regression: Discriminative Parameter Learning of Belief Net Classifiers. Maching Learning 59, 297–322 (2005).
9. Su, J., Zhang, H., Ling, C.X., Matwin, S.: Discriminative parameter learning for Bayesian networks. In: Proceedings of the 25th international conference on Machine learning. ACM, New York, NY, USA, pp. 1016–1023. (2008)
10. Pernkopf, F., Wohlmayr, M.: On Discriminative Parameter Learning of Bayesian Network Classifiers. In: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part II (ECML PKDD '09), Wray Buntine, Marko Grobelnik, Dunja Mladenić, and John Shawe-Taylor (Eds.). Springer-Verlag, Berlin, Heidelberg, pp. 221–237. (2009)
11. Carvalho, A.M., Roos, T., Oliveira, A.L., Myllymäki, P.: Discriminative Learning of Bayesian Networks via Factorized Conditional Log-Likelihood. J. Mach. Learn. Res. 12, 2181–2210. (2011)
12. Zaidi, N.A., Webb, G.I., Carman, M.J., Petitjean, F., Buntine, W., Hynes, M., Sterck, H.: Efficient parameter learning of Bayesian network classifiers. Maching Learning. 106, 1289–1329. (2017)
13. Sundararajan, P.K., Mengshoel, O.J.: A Genetic Algorithm for Learning Parameters in Bayesian Networks using Expectation Maximization. In: Proceedings of the 8th International Conference on Probabilistic Graphical Models, PMLR 52, pp. 511–522. (2016)
14. Zhang, J., Sanderson, A.: JADE: adaptive differential evolution with optional external archive. IEEE Transactions on evolutionary computation, 13(5), pp. 945–958. (2009).
15. Price, K., Storn, R.: Minimizing the Real Functions of the ICEC'96 contest by Differential Evolution, IEEE International Conference on Evolutionary Computation (ICEC'96), pp. 842–844. (1996)

16. Tanabe, R., Fukunaga, A.S.: Improving the Search Performance of SHADE Using Linear Population Size Reduction. IEEE Congress on Evolutionary Computation (CEC), Beijing, pp. 1658–1665. (2014)

ISSN 1870-4069

Optimization of Energy Production in Oklahoma: A Variant of an Evolutionary Algorithm of Multi-Objective Optimization

Gustavo Adolfo Vargas Hakim, Efrén Mezura Montes

University of Veracruz,
Artificial Intelligence Research Center,
Mexico

vargashakimg@gmail.com, emezura@uv.mx

Abstract. A preliminary performance assessment of the well-known Non-Dominated Sorting Genetic Algorithm II and one of its variants to optimize the electrical power production in the state of Oklahoma is presented in this paper. Such variant has a chaotic model to generate the initial population. Solar, wind and natural gas power systems, the first two renewable energies, are considered in the problem of interest. Three conflicting objectives are optimized: (1) power production, (2) production costs and (3) CO_2 emissions. The spacing metric is computed to compare the performance of both variants. The obtained results suggest that the chaotic model for the initial population does not improve the performance of the original algorithm in this particular multi-objective optimization problem.

Keywords: evolutionary multi-objective optimization, energy production, renewable energies.

1 Introduction

Climate change is a very relevant problem that human kind is facing. The United Nations has warned that approximately a dozen years are left to limit climate change at 1.5 °C in order to prevent a world crisis [1]. Power production is one of the key elements to consider. Fossil fuels have been and continue to be widely used to produce electricity. These energy sources, when burned to generate heat to produce power, spreads nitrogen oxides and other contaminants that contribute to the smog and acid rain [2].

Several countries, including the United States, have started to adopt renewable energy sources, such as solar and wind energy, to produce power. In 2017, around the 11% of the total consumed energy in the US came from renewable energies [3]. With respect to the state of Oklahoma, according to the US department of energy, 50% of the annual energy production in 2016 comes from natural gas, 38% from coal, 10% comes from solar and wind power and 1% from hydro-energy [4]. Despite the US government position about climate change, as mentioned in [5], the renewable energy industry is expected to grow and play an important role in the energy production in some of the most populated states of the country.

Beyond the merely direct utilization of these power sources, an optimal distribution of the generation power between them is highly desirable. To optimize the power production problem different multi-objective evolutionary algorithms (MOEAs) have been adopted. The algorithm of interest in this paper is the Non-Dominated Sorting Genetic Algorithm II (NSGA-II) [6], which has been a popular choice to solve multi-objective power optimization problems. Wahlroos *et al.* [7] optimized a generation system in terms of CO_2 emissions, production costs and production adequacy, using NSGA-II. Wang and Zhou [8] utilized the same algorithm to optimize the emissions and energy-savings of a wind power system. Liu and Dongdong [9] optimized a multiple source power system considering its production cost and the amount of emissions it produces. Zhou and Sun [10] utilized this MOEA to optimize a hybrid energy system consisting of solar power and wind power. In this study, the results were compared with a modified version of this algorithm, called L-NSGA-II [11].

Motivated by the above mentioned, and particularly by [10], a multi-objective optimization problem is stated for the power production of Oklahoma and solved using NSGA-II and one of its variants (L-NSGA-II) with the aim to assess their performance in this new real-world instance.

The paper is organized as follows: Section 2 states the problem of interest, Section 3 details both, NSGA-II and the variant adopted in this work. After that, Section 4 includes the experiments and results and Section 5 presents the conclusions and future work.

2 Problem Statement

A multi-objective optimization problem can be defined, without loss of generality, as to: find a solution vector $\vec{x} = [x_1, x_2, \dots, x_n]^T$, which minimizes $\vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \dots, f_m(\vec{x})]^T$, where each $x_j \in [lo_j, up_j]$.

Pareto dominance is used as a criterion to solve multi-objective optimization problems, and it is defined as follows: a solution vector $\vec{x} = [x_1, x_2, \dots, x_n]^T$ is said to dominate $\vec{y} = [y_1, y_2, \dots, y_n]^T$, denoted as $\vec{x} \prec \vec{y}$ if and only if $f_i(\vec{x}) \le f_i(\vec{y})$ for all $i \in [1, \dots, m]$ and $f_i(\vec{x}) < f_i(\vec{y})$ for at least one $i \in [1, \dots, m]$.

A solution vector $\overline{x^*}$ is part of the Pareto Optimal Set P^* if there does not exist other solution $\vec{x}$ such that $\vec{x} \prec \overrightarrow{x^*}$. The Pareto Optimal front is then $PF^* = \{\vec{f}(\vec{x}) | \vec{x} \in P^*\}$.

The analyzed power production system, as previously mentioned, is based on solar, wind and natural gas energy. The three objectives to be considered are: (1) the overall power production, (2) the overall production cost, and (3) the overall CO_2 emission. The decision variables are the operation hours of the solar, wind and natural gas production systems, *hs*, *hw* and *hg*, respectively, and the natural gas power produced *Pg*, which is assumed without considering the ways to produce it.

2.1 Power Production Function

This objective function represents the entire, monthly amount of power produced by the three sources. This objective function is constructed as shown in Eq. 1:

$$P = f_s \boldsymbol{h_s} \overline{P_s} + f_w \boldsymbol{h_w} \overline{P_w} + f_g \boldsymbol{h_g} \boldsymbol{P_g}. \tag{1}$$

Decision variables are bolded. $\overline{P_s}$ and $\overline{P_w}$ are the estimation of the generated solar and wind power for any arbitrary month (explained later). P_g is the generated natural gas power, considered as decision variable (assuming it could be controlled). Constants f_s, f_w and f_g are the capacity factors of solar, wind and natural gas systems, respectively (see Table 1).

Table 1. Capacity factors (percentage of input power that is effectively transformed into electrical power) of the three studied energy sources.

Power Source	**Capacity Factor (%)**
Solar Power (f_s)	33
Wind Power (f_w)	43
Natural Gas Power (f_g)	87

To get all three objective functions to be minimized, this objective function, originally to be maximized, is transformed by using the concept of power relation as in Eq. 2:

$$PR_T = \frac{N * P_{SWG}}{f_s \boldsymbol{h_s} \overline{P_s} + f_w \boldsymbol{h_w} \overline{P_w} + f_g \boldsymbol{h_g} \boldsymbol{P_g}}, \tag{2}$$

where P_{SWG} is the current amount of energy produced by the three sources together. This value is approximately 7.07 MWh and it was measured for the month of August, 2018 [12]. N is an increasing factor. In an ideal context, where renewable power is reinforced, the increasing factor is greater than one, so that the total energy from solar, wind and natural gas power is increased. For this study, this total energy production is encouraged to be doubled, then $N = 2$.

2.2 Production Cost Function

The second objective function is the total, monthly production cost. Eq. 3 shows this objective to be minimized:

$$C = c_s \boldsymbol{h_s} \overline{P_s} + c_w \boldsymbol{h_w} \overline{P_w} + c_g \boldsymbol{h_g} \boldsymbol{P_g}, \tag{3}$$

where constants c_s, c_w and c_g are the production costs of solar, wind and natural gas systems, respectively (see Table 2).

Table 2. Production cost of solar, wind, and natural gas power systems.

Power Source	**Production Cost ($/MWh)**
Solar Power (c_s)	48.2
Wind Power (c_w)	33
Natural Gas Power (c_g)	15.5

Data from Tables 1 and 2 are approximations obtained from [13].

2.3 CO_2 Emissions Function

The third objective function is the monthly CO_2 emissions caused by the three energy sources. This objective function to be minimized is introduced in Eq. 4:

$$E = e_s \boldsymbol{h_s} \overline{P_s} + e_w \boldsymbol{h_w} \overline{P_w} + e_g \boldsymbol{h_g} \boldsymbol{P_g}, \tag{4}$$

where the CO_2 emissions rates for the solar, wind and natural gas systems are e_s, e_w and e_g, respectively (see Table 3).

Table 3. CO_2 emissions rates (related with global warming [14]) for solar, wind, and natural gas power systems.

Power Source	**CO_2 Emissions Rate (gr/KWh)**
Solar Power (e_s)	48.2
Wind Power (e_w)	33
Natural Gas Power (e_g)	15.5

The estimations of solar and wind power $\overline{P_s}$ and $\overline{P_w}$, depend on the solar radiation and wind speed, so climate predictions are required. Climate is hard to predict. Moura and de Almeida proposed a climate model prediction for Portugal, based on previous data [15]. A similar model is used for the state of Oklahoma but using applicable data. A dataset was built in Microsoft Excel with measurements of solar radiation and wind speeds for each month of each year from 2003 to 2017, provided by MESONET, an environmental monitoring station available in the state of Oklahoma [16].

The solar radiation measured by MESONET is in MJ/m^2. It is multiplied by the total solar panels area and divided by the total number of seconds in each month. Months with 30, 31 and 28 days were considered. The result unit is the Watt (W). The wind speed is measured in miles per hour and only needs to be converted to m/s.

3 NSGA-II and L-NSGA-II

NSGA-II is a genetic algorithm (GA) adapted to solve multi-objective optimization problems. Besides those GA elements (tournament selection and crossover and mutation operators), NSGA-II uses the so-called non-dominated sorting process to rank solutions based on Pareto dominance from the union of parents and offspring.

Those non-dominated solutions get rank 1 and they are separated from the aforementioned union.

```
Begin
    Generate an initial population Pop with POPSIZE solutions
    Evaluate each solution in the objective functions
    Apply non-dominated sorting to Pop
    While termination condition not reached
        Generate offspring population Offs
        Evaluate each offspring in the objective functions
        Apply non-dominated sorting to Pop U Offs
        Select, based on ranking and crowding distance, the next population from Pop U Offs
    End While
End
```

Fig. 1. NSGA-II general pseudocode.

From the remaining solutions those non-dominated are assigned rank 2 and so on. The next population is chosen based on ranking. Furthermore, a crowding-distance measure in the objective space is used to choose among solutions with the same ranking to get a population with the same size to start the next generation. A general pseudocode is presented in Figure 1.

L-NSGA-II keeps most of the original NSGA-II structure, except for adding a different population initialization method. Here, a hybrid chaotic model is defined for the initialization part. The usual initialization method is shown in Eq. 5:

$$x_j = lo_j + (up_j - lo_j)u, \tag{5}$$

where u is a random number with uniform distribution; lo_j and up_j are the variable boundaries. In L-NSGA-II, the random number u is substituted. First, for the current value of a counter k, there are two randomly found numbers (between 0 and 1) defined as part of the hybrid chaotic mapping model, u_k and r_k. The value of u for the next count $k + 1$ is defined in Eq. 6:

$$u_{k+1} = \mu u_k(1 - u_k), \tag{6}$$

where μ is a control variable, set as 0.5 for this study. The value of r for the step $k + 1$ is calculated depending on the value of r_k. This is displayed in Eq. 7:

$$r_{k+1} = \begin{cases} \dfrac{1}{1.001}(2r_k + 0.001u_k), & 0 \leq r_k \leq 0.5, \\ \dfrac{1}{1.001}[2(1 - r_k) + 0.001u_k], & 0.5 < r_k \leq 1. \end{cases} \tag{7}$$

Then, the initialization of a single variable of the next step $k + 1$ is as in Eq. 8:

$$x_j = lo_j + \left(up_j - lo_j\right)r_{k+1}. \tag{8}$$

According to the authors of this initialization proposal, this initialization model should contribute to the diversity of solutions in the Pareto front. Diversity is highly desired as it provides for more options to choose between advantages and disadvantages of each possible solution. It remains to be seen if this applies for the problem of interest in this paper.

4 Experiments and Results

The optimization of the power production was conducted following the most accurate conditions that were possible, i.e., actual wind farms (412 wind turbines) and solar panels (20,000) that are currently in use in Oklahoma were considered. The decision variables were constrained due to the real limitations of the power production systems and the problem requirements. The hours of operation could not exceed the number of hours in a month. The maximum number of days considered was 28 (as February is the shortest month), equivalent to 672 hours. The minimum number of hours allowed was 240 hours. The maximum produced natural gas power was 7.07 MW and its minimum produced power was 4.88 MW. The boundaries of the decision variables are summarized in Table 4.

Table 4. Boundaries of the decision variables.

	h_s	h_w	h_g	P_g
Maximum	672 hr	672 hr	672 hr	7.07 MW
Minimum	240 hr	240 hr	240 hr	4.88 MW

The simulations of power production were executed using MATLAB using an Intel Core i7 processor. A previously constructed NSGA-II framework by Seshadri was utilized and modified to run the problem objectives and also to code L-NSGA-II [17].

Both algorithms were tested using 100 generations and an initial population of 20 individuals. Crossover and mutation were performed with 90% and 10% of probabilities, respectively, in both cases.

Yen and He defined several metrics to test the performance of MOEAs [18]. In this work the Spacing metric was chosen as it measures how diverse or well distributed are the solutions in a Pareto front. Eq. 9 describes the spacing metric:

$$S = \sqrt{\frac{1}{\bar{n}}\sum_{i=1}^{\bar{n}}(d_i - \bar{d})^2}, \tag{9}$$

where d_i is the Euclidean distance between a solution x_i and its nearest solution, $\bar{n}$ is the number of solutions in the Pareto front and $\bar{d}$ is the average Euclidean distance between solutions. A lower value indicates a better solution distribution.

For this paper, the experiments covered the month of May, 2022 and each algorithm was run 25 times. The spacing value was calculated for each one of the 25 fronts obtained per variant and the averages per variant are shown in Table 5.

Table 5. Average spacing for the Pareto Fronts of each variant for May, 2022.

Algorithm	Average Spacing
NSGA-II	6.87×10^8
L-NSGA-II	6.77×10^8

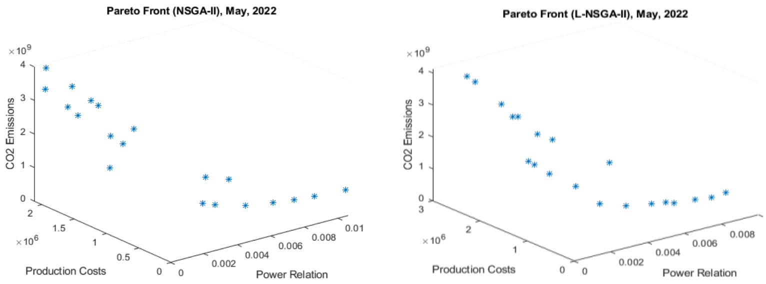


Fig. 2. Pareto front using NSGA-II (left) and using L-NSGA-II (right).

Considering the fact that none of the results samples fit the normal distribution (based on the Kolmogorov-Smirnov test), the Wilcoxon Signed-Rank test was computed, and its result ($p - value = 0.672$) indicated no significant differences between the two compared algorithms. Fig. 2 shows two Pareto fronts from both algorithms and a slightly better distribution provided by L-NSGA-II can be observed.

The decision-making of the best solution from the obtained Pareto front was based on the third objective function, i.e., CO_2 emissions. Such decision was based on the idea of getting the most environmentally friendly power production option. From the Pareto front in Fig. 2 (right), the individual that produced the lowest emissions was chosen (h_s = *361.06 h*, h_w = *419.79 h*, h_g = *251.6 h* and P_g = *6.837 MW*), which corresponds to the L-NSGA-II front. It is worth noticing that the operation hours number (h_g = *251.6 h*) of the natural gas system was the lowest of the three sources, reducing the CO_2 emissions.

5 Conclusions and Future Work

A comparison of a variant of the popular NSGA-II algorithm with the original to solve one instance of the power production in Oklahoma was presented in this paper. The variant was L-NSGA-II, which had a chaotic model to generate the initial population. The spacing metric was used for comparison purposes between the two algorithms when solving one instance of the problem related with one month (May 2022). The statistical results obtained suggested that the way L-NSGA-II generates the initial population does not produce any significantly distribution improvement in the Pareto front. The preference handling also showed that, when preferring the objective related to CO_2 emissions, an environmentally friendly solution can be obtained.

Future work consists on testing both variants in more problem instances and using other metrics like hyper-volume. Moreover, preference handling can be used within the search to focus only on well-distributed solutions favoring low CO_2 emissions.

Appendix

The codes used and the climate database constructed for this paper can be found and downloaded following the link below: http://drive.google.com/drive/folders/1eNfMthw7v-i-gK_UkQx_zW6Wk3zFgt6J?usp=sharing

References

1. The Guardian: https://www.theguardian.com/environment/2018/oct/08/global-warming-must-not-exceed-15c-warns-landmark-un-report. (2018)
2. US EPA: https://www.epa.gov/nutrientpollution/sources-and-solutions-fossil-fuels. (2019)
3. US EIA: https://www.eia.gov/energyexplained/?page=renewable_home. (2019)
4. US Department of Energy: https://www.energy.gov/sites/prod/files/2016/09/f33/OK_Energy%20Sector%20Risk%20Profile.pdf. (2019)
5. Rödl & Partner: https://www.roedl.com/insights/erneuerbare-energien/2017-05/renewable-energy-trump-administration. (2019)
6. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multi-objective Genetic Algorithm; NSGA-II. IEEE Transactions on Evolutionary Computation. 6(2), pp. 182–197. (2002)
7. Wahlroos, M., Jääskeläinen, J., Hirvonen, J.: Optimisation of an Energy System in Finland using NSGA-II Evolutionary Algorithm. 2018 15th International Conference on the European Energy Market (EEM). (2018)
8. Wang, J., Zhou, Y.: Multi-objective Dynamic Unit Commitment Optimization for Energy-saving and Emission Reduction with Wind Power. 2015 5th International Conference on Electric Utility Deregulation and Restructuring and Power Technologies (DRPT). (2015)
9. Liu, T., Dongdong, Z..: Multi-Objective Optimal Calculation for Integrated Local Area Energy System Based on NSGA-II Algorithm. 2019 IEEE International Conference on Energy Internet (ICEI). (2019)
10. Zhou, T., Sun, W.: Optimization of Wind-PV Hybrid Power System based on Interactive Multi-objective Optimization Algorithm. 2012 International Conference on Measurement, Information and Control. (2012)
11. Liu, T., Gao, X., Wang, L.: Multi-objective Optimization Method using and improved NSGA-II algorithm for oil-gas production process. Journal of the Taiwan Institute of Chemical Engineers, 57, pp. 42–53. (2015)
12. US EIA: https://www.eia.gov/state/?sid=OK#tabs-4. (2019)
13. IRENA: https://www.irena.org/-/media/Files/IRENA/Agency/Publication/2018/Jan/IRENA_2017_Power_Costs_2018.pdf. (2019)
14. NASA: https://climate.nasa.gov/vital-signs/carbon-dioxide/. (2019)

15. Moura, P., De Almeida, A.: Multi-Objective Optimization of a Mixed Renewable System with Demand-Side Management. Renewable and Sustainable Energy Reviews. 14(6), pp. 1461–1468. (2010)
16. MESONET: https://www.mesonet.org/index.php/weather/mesonet_averages_maps #y=average&m=12&p=wspd_mx&d=false. (2019)
17. Seshadri, A.: https://www.mathworks.com/matlabcentral/fileexchange/10429-nsga-ii-a-multi-objective-optimization-algorithm. (2019)
18. Yen, G., He, Z.: Performance Metrics Ensemble for Multi-Objective Evolutionary Algorithms. IEEE Transactions on Evolutionary Computation, 18(1), pp. 131–144. (2013)

Thematic Section
“Causal Reasoning”

Luis Enrique Sucar
Nicandro Cruz-Ramírez
Felipe Orihuela-Espina
Samuel Montero
Jenny Betsabé Vázquez-Aguirre (eds.)

ISSN 1870-4069

Causal Based Q-Learning

Arquímides Méndez Molina, Ivan Feliciano Avelino,
Eduardo F.Morales, L. Enrique Sucar

Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE),
Coordinación de Ciencias Computacionales,
Mexico

arquimides.mendez@gmail.com, ivan.felavel@gmail.com

Abstract. Reinforcement learning and Causal Inference are indispensable part of machine learning. However, they are usually treated separately, although that both are directly relevant to problem solving methods. One of the challenges that emerge in Reinforcement Learning, is the trade-off between exploration and exploitation. In this work we propose to use causal models to attend the learning process of an agent. The causal models helps to restrict the search space by reducing the actions that an agent can take through interventional queries like: *Would I have achieved my goal if I had drop the passenger off here?.* This simulates common sense that lightens the time it takes the trial and error approach. We attack the classic taxi problem and we show that using causal models in the Q-learning action selection step leads to higher and faster jump-start reward and convergence, respectively.

Keywords: reinforcement learning, causal models, taxi domain.

1 Introduction

Reinforcement learning (RL) is the study of how an agent can learn to choose actions that maximize its future rewards through interactions with an environment [18]. RL is a technique to solve complex sequential decision making problems in several domains as healthcare, economics, robotics, among others. Existing studies apply RL algorithms in discovering optimal policies for a targeted problem, but ignores the abundant causal relationships present in the target domain.

Causal inference (CI) is another learning paradigm concerned at uncovering the cause-effect relationships between different variables [16,15]. CI addresses questions like: If I desire this outcome, what action do I need to take? So it may provide the information for an intelligent system to predict what may happen next so that it can better plan for the future. Given a causal structure of a system it is possible to predict what would happen if some variables are intervened, estimate the effect of confounding factors that affect both an intervention and its outcome, but also, predict the outcomes of cases that are never observed before.

Both reinforcement learning (RL) and causal inference have evolved independently and practically with no interaction between them, despite the fact that

pp. 95–104; rec. 2019-09-16; acc. 2019-09-30

both are directly relevant to problem solving processes. Nonetheless, recent work has focused on connecting these fields [8,9,20,5]. The goal of these works is to show how RL can be made more robust and general through causal mechanisms or vice versa. Also, a growth in what some are beginning to call (CausalRL) [12] is expected to become an indispensable part of General Artificial Intelligence. What CausalRL does seems to mimic human behaviors, learning causal effects from an agent communicating with the environment and then optimizing its policy based on the learned causal relations.

One of the challenges that emerge in Reinforcement Learning, is the trade-off between try new actions (exploration) and select the best action based on previous experience (exploitation) in a given state. Traditional exploration and exploitation strategies are undirected and do not explicitly chase interesting transitions. Using predictive models is a promising way to cope with this problem. In particular, these models may hold causal knowledge, that is, causal relationships.

In the present investigation we propose a method to guide the action selection in an RL algorithm using one or more causal models as oracles. The agent can consult those oracles to not perform actions that lead to unwanted states or choose the best option. This helps the agent learn faster since it will not move blindly. Through interventions in the causal model, we can make queries of the type *What if I do ...?*, e.g., If I drop the passenger off here, will my goal be achieved? This type of interventions can help to reduce the search space. An important distinction is that, in order to use a causal model as in favor of a reinforcement learning algorithm, we do not need it to be complete. In other words, we can think of one or several partial models that express relationships between variables of one or several subtasks of the general task we are trying to solve.

The remainder of this paper is organized as follows. Section 2 reviews related works. Section 3 describes in a very general way some concepts used in the proposal. Section 4 describes the proposed method. In Section 5 the experimental set-up is described and the main results presented. Finally, in Section 6, conclusions and future research directions are given.

2 Related Work

RL and CI have been widely explored separately [16,18]. Nevertheless, there are recent studies that are looking to connect the concepts of these two areas to set something they call *Causal Reinforcement Learning*, a paradigm that unites both approaches to solve problems that cannot be solved individually in each discipline [1,11]. The authors in [8], from a psychological approach, establish that the model used in model-based reinforcement learning algorithms it is causal. Taking an action in a state causes both a reward and a transition to a new state. However, the manipulationist mechanism is not addressed or explained.

Some other works have focus on handling confounders (those variables that affect action and output) in classic RL problems [2,12,7]. Besides that, it has been show that causal reasoning can arise from RL [4,13].

The idea of using knowledge from causal models to avoid or reduce trial-and-error learning in RL has not been explored, as far as we know. Authors in [14] propose a new method to speed up RL training through the use of a property that they define as *state-action permissibility*.

The main idea is to have a predictor that guides the action selection step. The predictor classifies whether an action leads to an optimal solution given the action and the current state. What distinguishes our work from this one is the use of causal model composed of state variables, actions and goals. Instead of consulting the model for predictions we propose to make intervention type queries so the agent is in the second rung of the ladder of causation.

3 Background

The definition of causality is that X causes Y, $X \to Y$, if and only if an intervention or manipulation in X has an effect on Y, keeping everything else constant [17].

A *graphical causal model* is a pair $M = \langle D, \Theta_D \rangle$ consisting of a *causal structure* D and a set of parameters compatible with D. A causal structure of a set of variables V is a directed acyclic graph (DAG) in which each node corresponds to a different variable, and each arc represents a direct relationship among the corresponding variables [16]. The parameters Θ_D assign a function $x_i = f_i(pa_i, u_i)$ to each $X_i \in V$ and a probability measure $P(u_i)$ to each u_i, where PA_i are the parents of X_i in D and where each U_i is a random disturbance distributed according to $P(u_i)$, independently of all other u.

To better illustrate the above, consider the following example. Travis is a taxi driver whose main goal is to pick up a passenger at a certain point (passenger position) and take him to his destination (destination position) and drop him off there. For Travis, meeting his goal is based on his common sense. He doesn't try to pick a passenger when there is no passenger, drop him off there when he doesn't has arrived to the goal position, etc. We can create a causal model from the rules that guide Travis.

The parameters of our causal model can be defined as Boolean variables like in the set of equations 1, where $u_1, u_2 \in \{True, False\}$, u_3, c_4, c_5 can take some value that characterizes some position in the environment, e.g., coordinates in a map (c_4 and c_5 can be constant values). The rest of $u_i, u'_i \in \{True, False\}$ variables can be seen as unusual behaviors.

Let's suppose the case when $onDestinationLocation = False$, even when the taxi is on the same position as the passenger, maybe the passenger position has been updated without notifying the taxi driver, in this scenario $u'_6 = True$.

The counterpart happens when $u_6 = True$, then the taxi is on the passenger position, see eq. 1 (the corresponding causal structure is shown in Figure 1):

$$\begin{aligned}
pickup &= u_1,\\
dropoff &= u_2,\\
cabPosition &= u_3,\\
destinationPosition &= c_4,\\
passengerPosition &= c_5,\\
onDestinationPosition &= [(destinationPosition = cabPosition) \vee u_6] \wedge \neg u_6',\\
onPassengerPosition &= [(passengerPosition = cabPosition) \vee u_7] \wedge \neg u_7',\\
inTheCab &= [(pickup = True \wedge onPassengerLocation = True)\\
&\quad \vee u_8)] \wedge \neg u_8',\\
goal &= [(dropoff = True \wedge inTheCab = True \wedge\\
&\quad onDestinationLocation = True) \vee u_9] \wedge \neg u_9'.
\end{aligned} \tag{1}$$

Causal models, unlike probabilistic models, can serve to predict the effect of *interventions*. Interventions allow us to make queries of the type: Would the passenger be inside the taxicab if we make sure that the passenger is picked up here?. An *intervention*, which we denote by $do(X_i = x_i)$, means removing the equation $x_i = f_i(pa_i, u_i)$ from the model and substituting $X_i = x_i$ in the remaining equations [16]. The new model represents the system's behavior under the intervention $do(X_i = x_i)$ and, when solved for the distribution of X_j, produces the *causal effect* of X_i on X_j, which is denoted $P(x_j|do(X_i = x_i))$.

For example, to intervene on the variable $inTheCab$ in our example would be to set to one despite of whether the passenger was picked up. We would represent this by replacing the equation $inTheCab = pickup \times onPassengerLocation$ with $inTheCab = True$. Graphically, we can think of the intervention as "breaking the arrows" pointing into $inTheCab$.

4 Proposed Method

Our hypothesis is that causal inference can assist RL in learning value functions or policies more efficiently through the use of causal relations between state variables or between actions and state variables and therefore reducing the state or action space significantly.

To that end we proposed a method which consists of applying Algorithm 3 as a modification of the exploitation stage of Q-learning [19]. In general the method operates as follows. The agent observes a state, and through queries to one or more causal models, selects the action likely to allow the agent to meet a goal. The parameters of each causal model are given by a probabilistic SEM. The variables of the model are divided in three sets: state variables X, actions A and targets Z. The variables are defined as follows: $x = f_x(Pa_x), x \in X$,

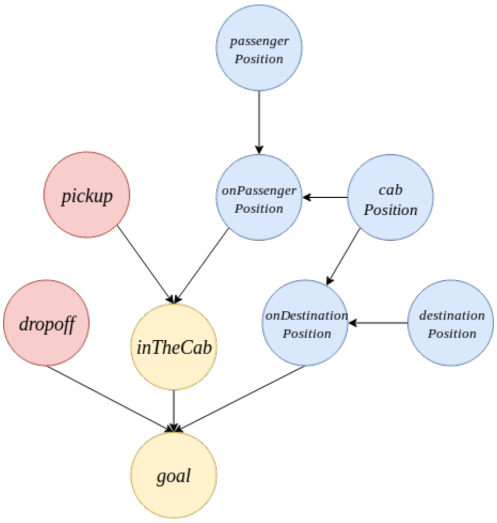


Fig. 1. Causal structure D for set of equations 1. The color of the nodes indicates to which set of variables corresponds. Red for actions (A), Yellow for target variables (Z) and blue for state variables (X). (Best seen in color).

$z = f_z(Pa_z), z \in Z$ where $Pa_x \subseteq X \cup A$ and $Pa_z \subseteq X \cup Z \cup A$. From the taxi example, the corresponding variables from Equation 1 for X, A, Z can be set as follows:

$$\begin{aligned} X = \{&passengerPosition, onPassengerPosition, cabPosition,\\ &onDestinationPosition, destinationPosition\},\\ A = \{&pickUp, dropOff\},\\ Z = \{&inTheCab, goal\}. \end{aligned}$$

In Algorithm 3, B is a set of observable instantiated variables, i.e., given the agent's observation we assign values to state variables from X. We assume that interventionist and observation distributions are already given so simply ask for $P(z|do(a), B)$ to obtain the causal effect in Algorithm 3 step 4. For our proposed method to work, the following assumptions must be meet:

- Non-empty set Z of target variables, can be ordered by a priority function.
- Non-empty set A of actions variables, containts only boolean variables.
- The agent can select only one action in a given state.
- All parameters of each Causal Model are defined.

5 Experimental Set-Up and Results

To show that our approach promises to be a way to improve RL we integrate it into the classical Q-learning algorithm. We replace the exploration step in

Algorithm 1: Q-Learning

```
input  : $< S, A, R >$
output: Table $Q$
Initialize $Q(s, a)$ arbitrarily
Repeat (for each episode):
    Initialize $s$
    Repeat (for each step of episode):
        Choose $a$ from $s$ using policy derived from $Q$(e.g., $\epsilon$ - greedy)
        Take action $a$, observe $r$, $s'$
        $Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma max_{a'} Q(s', a')) - Q(s, a))]$
        $s \leftarrow s'$
    until $s$ is terminal or invalid
return $Q$
```

Algorithm 2: Causal Q-Learning

```
input  : $< S, A, R >, G$
output: Table $Q$
Initialize $Q(s, a)$ arbitrarily
Repeat (for each episode):
    Initialize $s$
    Repeat (for each step of episode):
         $a \leftarrow$ interventional based selection using (s, G)
         If $(a = None)$:
            Choose $a$ from $s$ using policy derived from $Q$(e.g., $\epsilon$ - greedy)
        Take action $a$, observe $r$, $s'$
        $Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma max_{a'} Q(s', a')) - Q(s, a))]$
        $s \leftarrow s'$
    until $s$ is terminal or invalid
return $Q$
```

ϵ-greedy method to choose the actions by our method that queries the model. The problem to solve is the classical taxi task [6]. Figure 11 graphically shows the problem. A 5×5 grid world dwelled by a taxi agent. There are four locations in this world, marked as R, B, G, and Y.

The taxi problem is episodic. In each episode, the taxi starts in a randomly-chosen square. There is a passenger at one of the four locations (chosen randomly), and that passenger wishes to be transported to one of the four locations (also chosen randomly). The taxi must go to the passenger's location, pick up the passenger, go to the destination location , and drop the passenger off there. The episode ends when the passenger is deposited at the destination location.

There are six primitive actions in this domain: (a) four navigation actions that move the taxi one square North, South, East, or West; (b) a Pickup action; and (c) a Drop off action. The six actions are deterministic. There is a reward of -1 for each action and an additional reward of +20 for successfully delivering

Algorithm 3: Action selection based on interventional queries.

```
Input  : A state $s$ sense by the agent, A set of causal models $G$, A set $Z$ of
         target variables of every $g \in G$ ordered by a priority function
Output: An action $a$.
$B \leftarrow get\_state\_observable\_values(s)$
foreach $z \in Z$ do
    foreach $a \in parents(z)$ where $a$ is an action variable do
        $p \leftarrow P(z = True|do(a = True), B)$
            ▷ Here we get the causal effect on the target variable $z$ through an
          intervention in the action variable $a$ using the causal model $g$
          containing $z$.
        if $p > 0.5$ then
            return $a$
        end
    end
end
return $None$
```

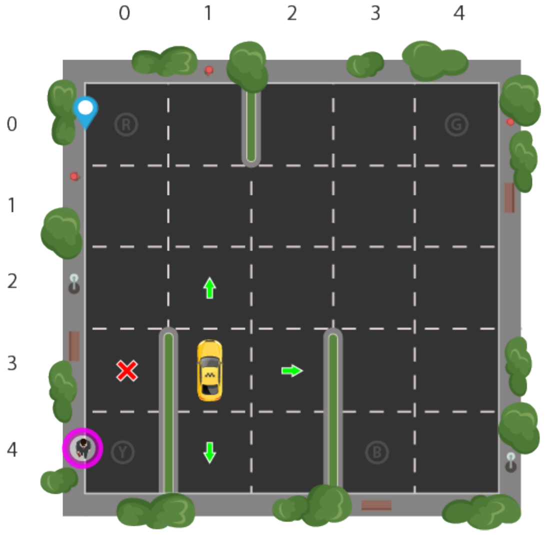


Fig. 2. Sketch of the taxi enviroment [10].

the passenger. There is also a 10 point penalty for illegal pick-up and drop-off actions [6]. There are 500 possible states: 25 squares, 5 locations for the passenger (including when he's inside the cab), and 4 destinations.

The causal model that is consulted to choose the actions is the presented in Section 3, extending it to queries on movement actions, so that the agent does not try move to positions where there are obstacles. For ease, we got rid of the u_i variables.

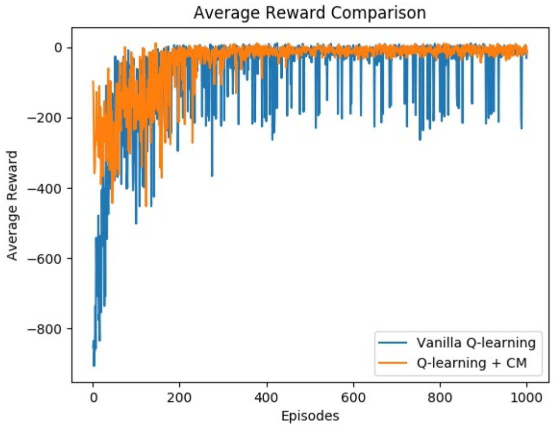


Fig. 3. Average reward of Vanilla Q-learning and Q-learning guided by a causal model.

From the model in Figure 1 the color of the nodes indicates to which set of variables corresponds. Red for actions (A), yellow for target variables (Z) and blue for state variables (X). Since the environment is deterministic, there is no need to compute a probability for the value of a target variable. Instead, we evaluate whether the value of the target variable is $True$ given the action and B.

As our baseline we implement a vanilla version of the Q-learning algorithm and we compare it with our version to which we denominate Q-learning + Causal Model (CM). We run 50 times each version of the algorithm and in each execution we compute the average reward per episode. Also, we set a qualifying mark based on the one established by Open AI Gym [1]. For this, we consider that the algorithm had reached an optimal reward once the average reward is equal to 9. So we assume that the algorithm that achieve it a smaller number of episodes is faster. On average, vanilla Q-learning reaches that reward in 95 episodes and Q-learning + CM in 65 episodes. In order to validate the results that the guided Q-learning version of the algorithm performs better than the vanilla version, we use the Wilcoxon Mann-Whitney rank sum test[3] with $p < 0.001$ to find statistical significant differences.

Figure 3 show the average reward per episode in both version of the algorithm for (average over 10 experiments). From the plot we can observe that our guided version starts with a higher reward. This is to be expected because, the agent doesn't start blindly. For a range of episodes there is no difference between the methods. However after a couple of hundred episodes, the Q-learnig guided by a causal model seems to converge and keeps more stable.

[1] `https://gym.openai.com/envs/Taxi-v1/`

6 Conclusions

Reinforcement Learning has proved to be successful in decision making problems. On the other hand, causal inference is clearly a novel but relevant and related area with untapped potential for any learning task. The use of causal models to provide auxiliary knowledge to an RL algorithm is a barely explored area. However, from the results obtained, we can see that this type of knowledge has the potential to accelerate RL. Although the problem attacked is simple because all the causes we have are direct and observable, the experimental results show that using causal models in the Q-learning action selection step leads to higher and faster jump-start reward and convergence, respectively. As future work we would like to try this action selection framework in Deep RL algorithms to solve more complex problems. Coping with more complex problems involves tasks not covered in this work, for example, undefined model parameters, incomplete causal structure or an unreliable causal model. In addition, we would like to explore the possibility that the causal model could also be learned during the training of the RL algorithm.

References

1. Bareinboim, E.: Causal reinforcement learning. `https://crl.causalai.net/` (2019)
2. Bareinboim, E., Forney, A., Pearl, J.: Bandits with unobserved confounders: A causal approach. In: Advances in Neural Information Processing Systems. pp. 1342–1350 (2015)
3. Colas, C., Sigaud, O., Oudeyer, P.Y.: A hitchhiker's guide to statistical comparisons of reinforcement learning algorithms (2019)
4. Dasgupta, I., Wang, J., Chiappa, S., Mitrovic, J., Ortega, P., Raposo, D., Hughes, E., Battaglia, P., Botvinick, M., Kurth-Nelson, Z.: Causal reasoning from meta-reinforcement learning. arXiv preprint arXiv:1901.08162 (2019)
5. Dasgupta, I., Wang, J.X., Chiappa, S., Mitrovic, J., Ortega, P.A., Raposo, D., Hughes, E., Battaglia, P., Botvinick, M., Kurth-Nelson, Z.: Causal reasoning from meta-reinforcement learning. CoRR abs/1901.08162 (2019), `http://arxiv.org/abs/1901.08162`
6. Dietterich, T.G.: Hierarchical reinforcement learning with the maxq value function decomposition. J. Artif. Int. Res. 13(1), 227–303 (Nov 2000), `http://dl.acm.org/citation.cfm?id=1622262.1622268`
7. Forney, A., Pearl, J., Bareinboim, E.: Counterfactual data-fusion for online reinforcement learners. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 1156–1164. PMLR, International Convention Centre, Sydney, Australia (06–11 Aug 2017), `http://proceedings.mlr.press/v70/forney17a.html`
8. Gershman, S.J.: Reinforcement learning and causal models. The Oxford handbook of causal reasoning p. 295 (2017)
9. Ho, S.: Causal learning versus reinforcement learning for knowledge learning and problem solving. In: The Workshops of the The Thirty-First AAAI Conference on Artificial Intelligence, Saturday, February 4-9, 2017, San Francisco, California,

USA. AAAI Workshops, vol. WS-17. AAAI Press (2017), http://aaai.org/ocs/index.php/WS/AAAIW17/paper/view/15182

10. Kansal, S.: Reinforcement q-learning from scratch in python with openai gym (2018), https://www.learndatasci.com/tutorials/reinforcement-q-learning-scratch-python-openai-gym/
11. Lu, C.: Introduction to causalrl (Jan 2019), https://causallu.com/2018/12/31/introduction-to-causalrl/
12. Lu, C., Schölkopf, B., Hernández-Lobato, J.M.: Deconfounding reinforcement learning in observational settings. CoRR abs/1812.10576 (2018), http://arxiv.org/abs/1812.10576
13. Madumal, P., Miller, T., Sonenberg, L., Vetere, F.: Explainable reinforcement learning through a causal lens. CoRR abs/1905.10958 (2019), http://arxiv.org/abs/1905.10958
14. Mazumder, S., Liu, B., Wang, S., Zhu, Y., Yin, X., Liu, L., Li, J., Huang, Y.: Guided exploration in deep reinforcement learning (2019), https://openreview.net/forum?id=SJMeTo09YQ
15. Pearl, J., Mackenzie, D.: The Book of Why: The New Science of Cause and Effect. Penguin Books Limited (2018)
16. Pearl, J.: Causality: models, reasoning, and interference. Cambridge University Press (2009)
17. Runge, J.: Causal network reconstruction from time series: From theoretical assumptions to practical estimation. Chaos: An Interdisciplinary Journal of Nonlinear Science 28(7), 075310 (2018), https://doi.org/10.1063/1.5025050
18. Sutton, R.S., Barto, A.G.: Reinforcement learning: an introduction. The MIT Press. (2018)
19. Watkins, C.J., Dayan, P.: Q-learning. Machine learning 8(3-4), 279–292 (1992)
20. Yu, C., Dong, Y., Liu, J., Ren, G.: Incorporating causal factors into reinforcement learning for dynamic treatment regimes in HIV. BMC Med. Inf. & Decision Making 19-S(2), 19–29 (2019), https://doi.org/10.1186/s12911-019-0755-6

ISSN 1870-4069

Indirect Spatiotemporal Short-Range Vehicle Communication Approach Inspired on Capillary Waves

Grigory Evropeytsev[1], Saul E. Pomares Hernandez[1,2],
Jose Roberto Perez Cruz[1], Lil María Rodríguez Henríquez[1,3]

[1] Instituto Nacional de Astrofísica, Óptica y Electrónica,
Mexico

[2] Centre National de la Recherche Scientifique,
France

[3] Consejo Nacional de Ciencia y Tecnologia,
Mexico

{grigory,spomares,jrpc}@inaoep.mx, lmrodriguez@conacyt.mx

Abstract. Short-range communication protocols allow for vehicles to exchange messages with the aim to provide new services for road security, emergency and intelligent transportation systems. For these services, each message has particular importance depending on where and when the event has happened. The message's importance degrades as time elapses and distance between the source and recipient increases. The similar effect can be observed on the water surface when an object falls on it. The resulting wave loses its strength with distance from the drop point and time. However, for vehicle communication, it is hard to establish this degradation due to: its distributed and asynchronous nature and the absence of permanent connections. This work presents an approach to establish the message's importance degradation while it is disseminated throughout a road network. To determine this effect, a fuzzy-causal closeness relation is used to combine information about traffic flow and location with temporal restrictions, expressed as causal dependencies. To face the lack of perdurable transmission links, the fixed communication elements embedded into the transportation infrastructure are used as communicant entities while vehicles are used as messages' carriers. In this way, the proposed solution operates with constant processing and communication overhead while the system scalability does not depend on the number of vehicles.

Keywords: spatiotemporal dependencies, fuzzy-causal closeness, capillary waves, indirect communication, short-range vehicle communication.

pp. 105–111; rec. 2019-09-16; acc. 2019-09-30

1 Introduction

Recent advances in wireless communication have allowed emerging standards for vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication, such as Dedicated Short-Range Communication (DSRC) [4]. This communication is mainly oriented to offer new services or enhance the current Intelligent Transportation Systems (ITS) such as lane change assistance, intersection coordination, emergency response time reduction, among others [7].

For these services, it is crucial to determine how important is a message depending on how far away and how long ago it was generated. For example, in an emergency response time reduction, traffic lights need to be coordinated to facilitate the circulation of emergency vehicles. In this scenario, an emergency vehicle dynamically adapts its trajectory according to the decisions of its driver and as a consequence, the nearest traffic lights must react to those changes.

As a message from the emergency vehicle is propagated through the road network, i.e. the farther it travels and the more time passes, the message loses its importance.

The same phenomenon can be observed on the water surface when an object falls. It produces ripples originating at the drop point of an object that propagates through the water surface. The resulting wave loses its strength with distance travelled and as time passes.

This scenario is depicted in Figure 1, where both the accident and the emergency vehicle generates capillary waves.

Even though this phenomenon is well studied by physics, it is hard to capture it in a V2V communication due to multiple factors. On the one hand, there are no perdurable transmission links among vehicles. On the other hand, interaction among vehicles is performed in a distributed and asynchronous fashion. As a result, it is hard to establish how far away and how long ago events have happened.

This work presents an approach for indirect spatiotemporal short-range vehicle communication inspired on capillary waves. To determine the messages' importance degradation, this approach is based on two main components: an indirect causal flooding protocol [3] and a fuzzy inference system to establish a degree of spatiotemporal closeness among events, that work together.

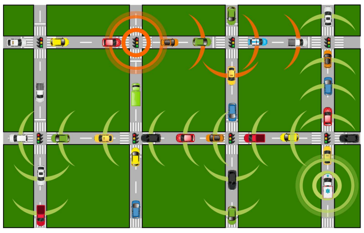


Fig. 1. Vehicular "capillary waves"

The causal flooding protocol was designed to face the lack of perdurable transmission links. However, this protocol establishes only the event order, but it cannot determine how long ago an event has happened.

This paper extends the flooding protocol by integrating a fuzzy inference system designed to support the constraints imposed by the asynchronous nature of interactions. In this sense, the spatiotemporal dependencies are estimated by combining heterogeneous data about traffic flow and location with temporal restrictions, expressed as causal dependencies.

By using these components, the solution does not require extra overhead. As a result, the system's scalability does not depend on the number of vehicles present in the system.

This paper is organized as follows. Section 2 presents a short overview of related works. The background and definitions are defined in section 3. The system model is presented in section 4. Section 5 describes the indirect spatiotemporal approach. The conclusions are presented in section 6.

2 Related Work

Several approaches have been proposed to determine whether the message in vehicular communication is important or not.

A Context-Aware Class Based Broadcast [2] proposes four different algorithms for message importance estimation. These algorithms consider one or two of the following parameters, without combining them: the number of retransmissions, geographical area and the message expiration time. Independently such algorithms, by considering these parameters, bring an exclusive binary result: important or not.

The algorithm proposed in [8] uses the linear combination of physical distance and expiration time, to determine a binary result, establishing whether the message is important or not.

Another solution is presented in the Floating Content approach [1]. In this work, the message is maintained among several surrounding devices located in a specific geographical area without requiring a dedicated communication node. As soon as the node leaves the area, the message is removed. Thereby, this solution is also a binary one.

Even though multiple solutions for a message's importance estimation exist, they are all based on binary decisions. If conditions are met, the message is considered important. Otherwise, the message is discarded. In addition, these solutions consider only parameters separately or using linear combinations. Thus, if one parameter falls outside of the threshold, the message is considered to be non-important.

3 Background

Definition 1. *Happened before relation. The HBR, denoted by "$\rightarrow$", is the least strict partial order on a set of events E, such that:*

1. *$\forall a, b \in E$, if a and b occur in the same process, and a was executed before b, then $a \rightarrow b$.*
2. *$\forall a, b \in E$ if a is the sending of a message m, and b is the reception of m sent through a, then $a \rightarrow b$.*
3. *$\forall a, b, c \in E$, if $a \rightarrow b$ and $b \rightarrow c$, then $a \rightarrow c$.*
4. *$\forall a \in E$, $a \not\rightarrow a$.*
5. *$\forall a, b \in E$, with $a \neq b$, if $a \rightarrow b$ then $b \not\rightarrow a$.*

4 System Model

For this work, a V2I DSRC-based communication [4] is modelled as a distributed system based on a loosely coupled ad hoc asynchronous message-passing scheme. Within the system, all entities (fixed and mobile) are represented as processes. To be able to communicate both entities should be in the communication range one with respect to another.

Any process performs instantaneous executions, referred to as events: the sending and the reception of messages. This proposal assumes that for any pair of events, a causal order can be established according to the *happened before relation* (HBR) [5].

5 Indirect Spatiotemporal Communication Approach

5.1 Communication Protocol

The message dissemination between fixed and mobile entities is performed using the indirect communication protocol introduced in [3]. Concurrently, the protocol establishes the causal dependencies between each pair of exchanged messages.

The communication protocol is summarized in the following cases.

Case 1. Messages generated by fixed entity: Messages generated by communication entities embedded into the transportation infrastructure are stored in a buffer. When a vehicle approaches the fixed entity, the later sends the message to the vehicle considering its movement direction. This entity sends the message to the vehicles that are following a specific direction (one message's copy per direction). After the transmission, the message is removed from the fixed entity's buffer.

Case 2. Message received from fixed entities by vehicle: A vehicle stores into its buffer messages received from fixed communication entities. The vehicle holds the message until it can be sent to the next fixed communication entity encountered.

Case 3. Messages received by fixed entities from vehicles: These messages are handled in the following manner.

1. The message's causal conditions imposed by HBR are verified. If HBR is satisfied, the message is delivered.
2. Otherwise, the message is buffered for a Δt time.

   (a) If during this time, the causal conditions become satisfied (due to other messages being received), the message is delivered immediately.
   (b) If the causal conditions are not satisfied after Δt, the message is delivered, marking the previous non-delivered messages as lost.
3. After the message is delivered, it is transmitted using the same mechanism as described in Case 1.

Case 4. Messages generated by a vehicle: Messages generated by vehicles are stored separately from messages received from fixed entities until they can be sent to the first encountered fixed entity. Delivered messages are handled in the same way as other messages generated by fixed entities (Case 1).

5.2 Fuzzy-Inference System to Determine the Message's Importance

The communication protocol described above disseminates messages and ensuring that HBR dependencies are not violated. In this way, the temporal coherence of the exchanged information is guaranteed.

The message's dissemination consumes time which implies an induced delay in its delivery. How long ago an event has happened can be estimated considering the distance between the sender and receiver as well as the followed transmission path and the traffic density. This estimation can be done in a similar form as humans estimate the trip delays by considering the transportation mode and the trip length.

The less traffic density (<90 veh/km/lane), the fewer vehicles are available to transmit messages, resulting in a greater delay for message transmission. The medium traffic density (90 - 150 veh/km/lane) offers the ideal message propagation conditions as vehicles are eagerly available and their movement is not restricted by other vehicles. When the traffic density is high (>150 veh/km/lane), the vehicles move slowly. An intuitive assumption is that the vehicle's slow speed will increase the transmission delays, however, this does not happen. This effect is not produced due to the possibility of passing messages from vehicle to vehicle, thus the slow vehicle's speed has no repercussions.

In vehicular communication, a message is retransmitted, due to propagation, multiple times between fixed entities and vehicles. Each retransmission induces a transmission delay. The number of retransmission represents the dissemination path length that the message has followed.

Due to the communication protocol only considers HBR-based causal dependencies to establish a timeline to order the received messages, it is not possible to determine how long ago an event a has happened before an event b. In this paper, a fuzzy inference system (FIS) was designed to relate heterogeneous data about traffic flow and location with temporal restrictions, expressed as causal dependencies to determine the causal closeness between two events. The FIS is based on the following inference:

"How far away, how long ago and how dense the traffic is implying how close or how important the message is".

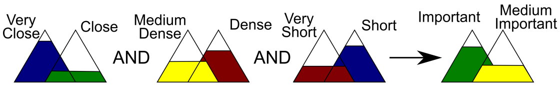


Fig. 2. Fuzzy inference system.

To estimate the message importance, we consider the following linguistic variables:

- *Path length*, whose universe of discourse is temporal data, expressed as causal dependencies.
- *Physical distance*, whose universe of discourse is the physical location.
- *Traffic density*, whose universe of discourse is the number of vehicles in a road segment.
- and *Fuzzy-causal closeness*, whose universe of discourse is the importance of the message.

The fuzzy sets, related to the four linguistic variables, are bounded as follows. The path length is bounded between 0 and 16 retransmissions. For the physical distance, the minimum and maximum values are set to 0 and 4 blocks. The boundaries for the traffic density values are 0 and 240 veh/km/lane. Finally, the fuzzy causal closeness or message importance is represented as a value between 0 and 1, where 0 means not important and 1 means very important.

By considering triangular membership functions for the fuzzy sets, the degree of fuzzy-causal closeness is determined through a Mamdani FIS [6]. The generalized version of this system is presented in Figure 2. In this way, the message's importance for a receiver can be estimated by defuzzification the outputs through a weighted average method.

6 Conclusions

This paper presents an indirect spatiotemporal short-range vehicle communication approach inspired on capillary waves. This approach extends the communication protocol [3] with the fuzzy inference system, that is designed to estimate the message's importance.

The proposed solution, determines the message's importance at each entity individually by combining heterogeneous data about traffic flow and location with temporal restrictions, expressed as causal dependencies.

An advantage of the approach is that it operates with constant communication and processing overhead and the amount of control information does not depend on the number of vehicles in the system.

References

1. Ali, S., Rizzo, G., Mancuso, V., Ajmone Marsan, M.: Persistence and availability of floating content in a campus environment. In: The 34th IEEE International Conference on Computer Communications (IEEE INFOCOM 2015) (04 2015)

2. Dressler, F., Klingler, F., Sommer, C., Cohen, R.: Not all VANET broadcasts are the same: Context-aware class based broadcast. IEEE/ACM Transactions on Networking 26(1), 17–30 (February 2018)
3. Evropeytsev, G., Pomares Hernández, S.E., Pérez Cruz, J.R., Rodríguez Henríquez, L.M., López Domínguez, E.: A scalable indirect position-based causal diffusion protocol for vehicular networks. IEEE Access 7, 14767–14778 (2019)
4. Kenney, J.B.: Dedicated short-range communications (DSRC) standards in the united states. Proceedings of the IEEE 99(7), 1162–1182 (July 2011)
5. Lamport, L.: Time, clocks, and the ordering of events in a distributed system. Commun. ACM 21(7), 558–565 (1978)
6. Mamdani, E.H., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. International Journal of Man-Machine Studies 7(1), 1–13 (1975)
7. Martinez, F.J., Toh, C., Cano, J., Calafate, C.T., Manzoni, P.: Emergency services in future intelligent transportation systems based on vehicular communication networks. IEEE Intelligent Transportation Systems Magazine 2(2), 6–20 (2010)
8. Xu, B., Ouksel, A., Wolfson, O.: Opportunistic resource exchange in inter-vehicle ad-hoc networks. In: IEEE International Conference on Mobile Data Management, 2004. Proceedings. 2004. pp. 4–12 (2004)

ISSN 1870-4069

Toward Knowledge Transfer for Learning Markov Equivalence Classes

Verónica Rodríguez López, Luis Enrique Sucar, Felipe Orihuela Espina

Instituto Nacional de Astrofísica, Óptica y Electrónica,
Mexico

{verorl,esucar,f.orihuela-espina}@inaoep.mx

Abstract. Most algorithms for causal discovery require large sample sizes for finding Markov equivalence classes that include the structure of the true causal probabilistic graphical models. In some situation collecting data could be difficult, especially for learning models that encode the specific causal relations of a particular subject of a population. Although transfer learning techniques have shown to be useful for improving predictive associative models learned with limited datasets, their application in the field of causal discovery has not been sufficiently explored. In this paper, we explore transferring weighted instances of auxiliary datasets for improving Markov equivalence classes learned with otherwise limited datasets. A knowledge transfer algorithm extended from the Greedy equivalence search algorithm that locally selects the instances of the best auxiliary datasets is proposed. Preliminary results using synthetic datasets suggest that our knowledge transfer algorithm outperforms the base algorithm, increasing the adjacency recall from 0.58 ± 0.28 to 0.94 ± 0.13.

Keywords: causal discovery, transfer learning, causal probabilistic graphical models.

1 Introduction

Causal probabilistic graphical models (causal PGMs) are useful tools for encoding causal relations between variables of closed systems and provide information to make predictions under manipulations. From observational data, it is possible discovering Markov Equivalence Classes (MECs) that represent the structure of a set of equivalent causal PGMs with the same joint probability distribution [2].

Learning MECs that include the true causal structure from a limited sample size could be challenging using many existing algorithms, since they find these MECs in the large sample limit [18, 5]. In some situations, it can be difficult collecting data, especially for learning casual PGMs that encode the specific causal relations for a particular member of a population. Transfer learning has shown to be useful for improving models learned with limited datasets, allowing the use of auxiliary data that come from different models with different probability distributions [15].

Many works have explored knowledge transfer for learning PGMs. However, most of these studies have relied on the learning of associative PGMs [10, 12–14]. Limited work [8] has been done on learning causal PGMs from observational data. Although other algorithms have been proposed for learning MECs from multiple datasets, their aim is different of that for knowledge transfer algorithms. These algorithms aim to discover MECs that include the common causal relations in all datasets, assuming that all datasets include a representative number of samples [3, 16, 17].

The knowledge transfer algorithm proposed in [8] is a modification of the PC algorithm that assumes all auxiliary datasets have the same relevance for learning a target MEC, ignoring their differences in probability distributions. Moreover, like other PC-based algorithms, require large sample sizes for the independence conditional tests [5]. Score-based algorithms have shown to be more accurate for learning MECs with small samples than constraint-based algorithms as PC [11]. In this paper, we present a preliminary knowledge transfer algorithm, based on the score-based algorithm, Greedy Equivalence Search [2], for improving MECs learned with limited datasets. We propose locally transferring the instances of the best auxiliary datasets, considering their differences in probability distributions with that of the target dataset.

The paper is organized as follows. In Section 2 concepts related to graphs and the Greedy Equivalence Search algorithm are described. Our knowledge transfer algorithm is presented in Section 3. In Section 4, the experimental results are shown. Finally, the conclusions of this paper are presented in Section 5.

2 Preliminaries

2.1 Graph Concepts

Definition 1. *A* ***graph*** *is a pair $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ formed by a set of nodes $\mathbf{V} = \{V_1, ..., V_N\}$, and a set of edges $\mathbf{E} \subset \mathbf{V} \times \mathbf{V}$.*

Two nodes are **adjacent** in a graph $\mathcal{G}$, if there is an edge associating them. When a graph only contains directed edges in the form $(V_1 \rightarrow V_2)$, it is called a **directed graph**. In a directed edge in the form $V_1 \rightarrow V_2$, V_1 is said to be the **parent** of V_2, and V_2, the **child** of V_1. The set of parents of a node V is denoted as $\mathbf{Pa}(V)$.

Definition 2. *Within a graph $\mathcal{G}$, a* ***directed path*** *between two nodes V_1 and V_k is a sequence of nodes, $(V_1, V_2, ..., V_k)$, starting at V_1 and ending at V_k, where $k \geq 2$, and $V_i \rightarrow V_{i+1} \in \mathbf{E}$ for $i = 1, ..., k-1$.*

A directed path where the last node coincides with the first one is a **directed cycle**. A directed graph in which there are no directed cycles is called a **directed acyclic graph** (DAG). If an acyclic graph contains directed and undirected edges, it is called a **partially directed graph** (PDAG).

The undirected graph resulting from ignoring the direction of edges in a DAG is the **skeleton** of the DAG.

A **v-structure** in a DAG is an ordered triple of nodes (X, Y, Z), such that, the edges $X \to Y$ and $Y \leftarrow Z$ are in the DAG, and there is no edge between the nodes X, Z [2].

Definition 3. *A* ***Markov equivalence class*** *is a set of directed acyclic graphs that have the same skeletons and the same v-structures [6].*

2.2 Greedy Equivalence Search Algorithm

Greedy Equivalence Search (GES) [2] is a score-based algorithm for heuristically searching the best Markov equivalence class that represents a set of equivalent DAGs including a true causal probabilistic graphical model. Given a dataset $\mathbf{D} = \{d_1, ..., d_m\}$ containing m instances, where each d_i represent an assignment of value to each variable of a set $\mathbf{X} = \{X_1, X_2, X_3, ..., X_n\}$, the best MEC $\mathcal{G}^* = (\mathbf{X}, \mathbf{E})$ is found by maximizing a scoring function such that:

$$\mathcal{G}^* = \arg\max_{\mathcal{G} \in \mathcal{G}_C} score(\mathcal{G}, \mathbf{D}), \tag{1}$$

where $score(\mathcal{G}, \mathbf{D})$ is a scoring function that measures the adjustment of $\mathbf{D}$ with a candidate MEC $\mathcal{G}$, and $\mathcal{G}_C$ is the set of all MECs defined over $\mathbf{X}$.

In the GES algorithm, Bayesian Dirichlet equivalent and Uniform (BDeU) score function is used for learning MECs defined over discrete variables with complete datasets $\mathbf{D}$ (without missing values). BDeU score is a descomposable function that can be expressed as the product of local functions $BDeU(X_i, \mathbf{Pa}(X_i), \mathbf{D})$ that only depends of a node $X_i \in \mathbf{X}$ and their parents $\mathbf{Pa}(X_i)$ as follows [7]:

$$BDeU(\mathcal{G}, \mathbf{D}) = \prod_{i=1}^{n} \left\{ BDeU(X_i, \mathbf{Pa}(X_i), \mathbf{D}) \right\}, \tag{2}$$

$$BDeU(X_i, \mathbf{Pa}(X_i), \mathbf{D}) = \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}, \tag{3}$$

where n is the number of nodes in $\mathcal{G}$, q_i is the number of values of $\mathbf{Pa}(X_i)$, r_i is the number of values of X_i, N_{ijk} is the number of cases in which $X_i = k$ and its parents $\mathbf{pa}(X_i = k) = j$, $N_{ij} = \sum_k N_{ijk}$, and $\alpha_{ijk} = \frac{1}{r_i q_i}$ is a Dirichlet prior parameter with $\alpha_{ij} = \sum_k \alpha ijk$.

BDeU score assigns the same value to all equivalent DAGs in the same MEC. It is used in each iteration of the GES algorithm for evaluating the improvement of the score when an edge is added or deleted. In the first stage of GES, starting with a empty graph, the scoring function is used for heuristically searching the edges that could be added of a MEC. And in the second stage, for searching the edges that could be removed of a MEC.

3 Instance-Based Transfer Learning GES

Our proposed preliminary algorithm, denominated as Knowledge Transfer Learning with Weighted instances GES (KTL-WeGES), is an extension of the Greedy Equivalence Search (GES) algorithm that using the instances of two auxiliary observational datasets tries to improve the skeleton identification of Markov equivalence class (MEC) learned with limited dataset.

Under the assumptions of causal sufficiency and faithfulness conditions, the best target MEC $\mathcal{G}_T^*$ is found by maximizing a scoring function that combines the instances of target $\mathbf{D}_T$ and auxiliary $\mathbf{D}_S$ datasets:

$$\mathcal{G}_T^* = \arg\max_{\mathcal{G}_T \in \mathcal{G}_C} score(\mathcal{G}_T, \mathbf{D}_T, \mathbf{D}_S). \tag{4}$$

For combining the instances of target and auxiliary datasets, local knowledge transfer of the auxiliary datasets is explored. In this local knowledge transfer, weighted instances of the auxiliary datasets are used for finding the best local structure for a target MEC composed by a node $X_i \in \mathbf{X}$ with their parents $\mathbf{Pa}_T(X_i)$. The local BDeU score defined in the Equation 3 is used for evaluating the adjustment of the combination of weighted instances of the auxiliary $\mathbf{D}_S$ and target $\mathbf{D}_T$ datasets, with a candidate local structure for a target MEC. In this equation, N_{ijk} counting the combination of auxiliary and target instances as follows:

$$N_{ijk} = (N_{ijk})_T + W_i(N_{ijk})_S, \tag{5}$$

where $(N_{ijk})_T$ represents the number of cases in $\mathbf{D}_T$ in which $X_i = k$ and its parents $\mathbf{pa}_T(X_i = k) = j$, and $(N_{ijk})_S$, the number of cases in $\mathbf{D}_S$ in which $X_i = k$ and its parents $\mathbf{pa}_T(X_i = k) = j$. W_i encode the relatedness of the auxiliar dataset with the candidate local structure for a target MEC.

In the estimation of this relatedness, differences in the conditional probability distribution of X_i and its parents $\mathbf{Pa}_T(X_i)$, between the target dataset $P_T(X_i|\mathbf{Pa}_T(Xi))$ and the auxiliary dataset $P_S(X_i|\mathbf{Pa}_T(Xi))$, are considered. The difference between these distributions is evaluated with the Kullback-Leibler divergence D_{KLD} [1] as follows:

$$D_{KLD}(P_T(X_i|\mathbf{Pa}_T(X_i)), P_S(X_i|\mathbf{Pa}_T(X_i))) \approx \sum_{x_i, \mathbf{pa}_T(x_i)} log\left(\frac{P_T(x_i|\mathbf{pa}_T(x_i))}{P_S(x_i|\mathbf{pa}_T(x_i))}\right). \tag{6}$$

Using this difference, W_i is estimated by:

$$W_i = 2^{-|D_{KLD}(P_T(X_i|\mathbf{Pa}_T(X_i)), P_S(X_i|\mathbf{Pa}_T(X_i)))|}. \tag{7}$$

With this function, when the difference between target and auxiliary datasets increases, it is penalized with weights nearly to zero; and it assigns weights nearly to one, to small differences lower to one.

4 Experiment and Results

4.1 Generation of Synthetic Datasets

Synthetic datasets are generated from ground truth Bayesian networks which are BN with known structure and parameters. Target and auxiliary datasets are generated in the following form [10]. Target dataset is sampled from the ground truth BN, and auxiliary datasets, from related BNs. Related BNs are generated modifying in certain percent ($pMod$) the edges of the ground truth models, adding $pMod$ edges, followed by deleting edges in the same $pMod$ percent. Increasing the $pMod$, we generate BN less related to the ground truth model. From each related BN are estimated its parameters using a dataset sampled from the ground truth BN. Each auxiliary dataset is sampled from its corresponding related BN using forward sampling, in which the values of each variable X_i are sampled in ancestral order (parents before their children), in such form that its values x_i are drawn from $P(x_i|\mathbf{pa}(x_i))$.

4.2 Experimental Design

In this experiment, we hypothesized that the KTL-WeGES algorithm outperform the GES algorithm. The performance of the KTL-WeGES algorithm was evaluated in its ability for finding the skeleton of the ground truth models. In the evaluation, the Coma [4] and Asia [9] binary BNs with five and eight nodes, respectively, were used as ground truth models. The edges of the original BNs were modified in 10% and 40%, for generating the two related BNs. Considering extreme cases of relatedness (most and least related) were selected these parameters. Coma and Asia BNs and their corresponding related BNs are presented in Figures 1 and 2, respectively. Datasets with 1600 and 12800 samples for Coma and Asia were used for estimating the parameters of related BNs.

Taking into account that after modifying the ground truth BNs would increase the number of parents for some nodes. The sample size was estimated using $samplesize = 100(2^k)$, considering that a node in a related BN may have at most $k = n - 1$ parents (where n is the number of nodes in the BN). For each auxiliar dataset, 1600 samples from related BNs of Coma and 12800 samples from related BNs of Asia (using the same formula for the parameters estimation), were obtained. Ten datasets varying the sample size were obtained for the target domain. For Coma, the set of target datasets includes datasets with size $\{50, 100, 150, 200, 250, 300, 350, 400, 450, 500\}$, and for Asia, with $\{80, 160, 240, 320, 400, 480, 560, 640, 720, 800\}$. Ten runs of this scenario were used to evaluate the algorithms.

The models obtained by the algorithms were evaluated using normalized structural Hamming distance (NSHD), adjacency precision (TPR), and adjacency recall (TDR). Normalized structural Hamming distance is the minimum number of edge insertions, deletions, and changes needed to transform a model into another. Adjacency precision is the ratio $TP/(TP + FP)$, and the ratio $TP/(TP + FN)$ is the adjacency recall.

Where TP is the number of adjacencies that are in common in the estimated model and ground truth model without considering the edge orientation; FP is the number of adjacencies that are present in the estimated model but not in the ground truth model; and FN is the number of adjacencies that are present in the ground truth model but not in the estimated model [17].

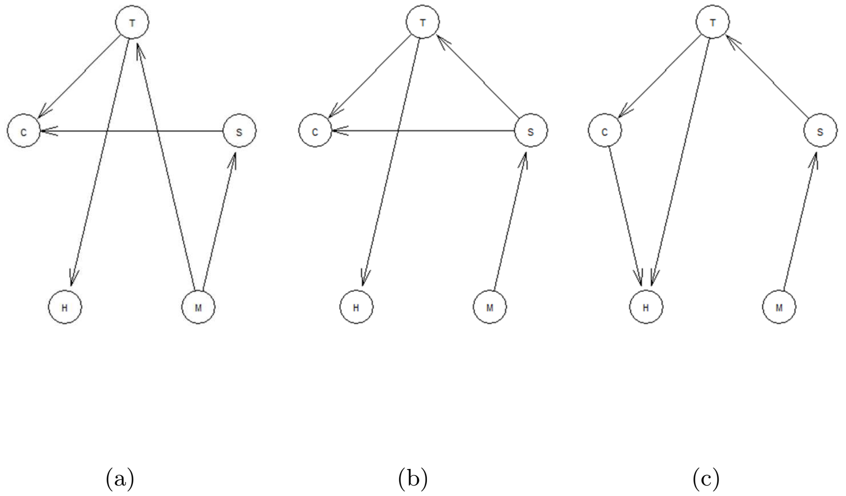
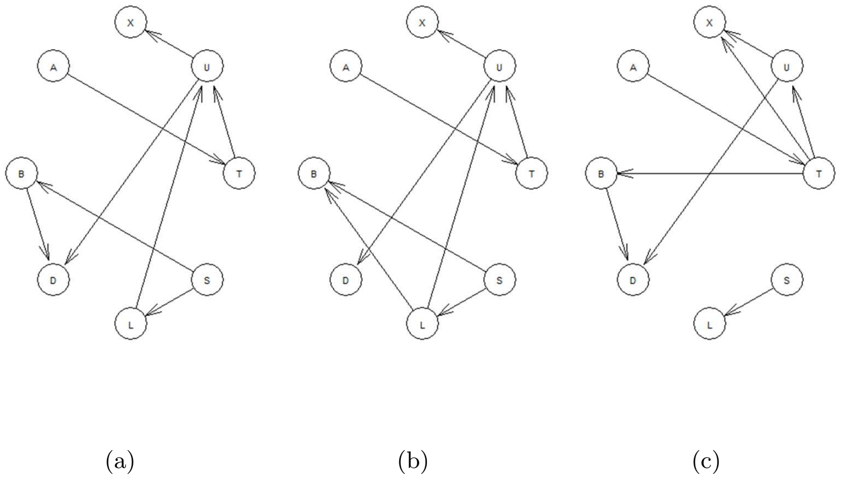


Fig. 1. (a) Coma and its related BNs created by modifying the edges in (b) 10% and (c) 40%.

4.3 Results

The experimental results are summarized in Tables 1 and 2 for Coma and Asia, respectively. In these tables, the averages for each metric, over the ten test target datasets and all experimental runs, obtained by transferring instances from the most related, the least related, and both auxiliary datasets, are presented.

The results show that KTL-WeGES seems to improve the skeleton identification of the ground truth models with respect to GES. In the case of Coma, considering the results for NSHD (the best NSHD is obtained when it is zero), KTL-WeGES seems to decrease the differences between the skeleton of the true and that one of the estimated model. The results for this model also show that, although the performance of the TPR decrease, KTL-WeGES are discovering more number of edges, increasing the TDR. The results for Asia show an improvement in the TPR and TDR rates.



(a) (b) (c)

Fig. 2. (a) Asia and its related BNs created by modifying the edges in (b) 10% and (c) 40%.

Table 1. Averages ± standard deviations of TPR, TDR, and NSHD for Coma.

Method	**TPR**	**TDR**	**NSHD**
GES	0.90 ± 0.13	0.59 ± 0.25	0.54 ± 0.25
KTL-WeGES (most related)	0.86 ± 0.11	0.94 ± 0.10	0.40 ± 0.31
KTL-WeGES (least related)	0.85 ± 0.09	0.98 ± 0.06	0.36 ± 0.28
KTL-WeGES (both auxiliar datasets)	0.84 ± 0.09	0.96 ± 0.08	0.38 ± 0.27

They also show that the differences between the skeleton of the true and that one of the estimated model increase, which indicate that the estimated model has more edges than the true model (spurious edges).

From the results, it also can be observed that KTL-WeGES with all strategies (transferring from all datasets, the best and least related auxiliary dataset) improves the TDR, being better transferring from the least related auxiliary dataset. Although, the NSHD and TPR results show that KTL-WeGES discovers more spurious edges when the number of nodes increases. It indicates that the scoring function prefers dense graphs, and hence KTL-WeGES has problems for deleting edges.

Regarding execution time for learning a single MEC on average, KTL-WeGES takes 0.46 and 10.78 seconds for learning models of Coma and Asia, respectively, with a 1.8 GHz Intel Core i7 processor with 8 GB RAM, using Matlab 2019a.

Table 2. Averages ± standard deviations of TPR, TDR, and NSHD for Asia.

Method	**TPR**	**TDR**	**NSHD**
GES	0.71 ± 0.27	0.58 ± 0.31	0.98 ± 0.44
KTL-WeGES (most related)	0.95 ± 0.07	0.90 ± 0.19	1.94 ± 0.33
KTL-WeGES (least related)	0.97 ± 0.05	0.90 ± 0.19	1.98 ± 0.35
KTL-WeGES (both auxiliar datasets)	0.97 ± 0.05	0.90 ± 0.19	1.99 ± 0.37

5 Conclusions

A preliminary instance-based transfer algorithm for improving Markov equivalence classes learned with limited datasets was presented. Our algorithm locally selects the instances from the two auxiliary datasets for searching the best set of parents of each node in a target MEC.

Experimental results show that our algorithm outperforms the GES algorithm in the skeleton identification for MECs, transferring weighted instances from the most related, the least related and both auxiliary datasets. Preliminary results suggest that our algorithm seems to be promising for discovering MECs.

As future work, we consider extending the local knowledge transfer of the weighted-instances for more than two auxiliary datasets and also analyzing other scoring functions and score-based algorithms that have shown better performance deleting false edges. Also, it is contemplated improving the algorithm for discovering the v-structures of MECs.

Acknowledgements. We acknowledge the support from the Secretaría de Educación, Ciencia, Tecnología e Innovación de la Ciudad de México SECITI/042/2018 (INGER-DI-CRECITES-001-2018) "Red Colaborativa de Investigación Traslacional para el Envejecimiento Saludable de la Ciudad de México" (RECITES). The first author acknowledges scholarship support from CONACYT as a PhD student.

References

1. Campos Ibáñez, L.M.: A scoring function for learning Bayesian networks based on mutual information and conditional independence tests. Journal of Machine Learning Research 7(Oct), 2149–2187 (2006)
2. Chickering, D.M.: Optimal structure identification with greedy search. Journal of Machine Learning Research 3(Nov), 507–554 (2002)
3. Claassen, T., Heskes, T.: Causal discovery in multiple models from different experiments. In: Advances in Neural Information Processing Systems. pp. 415–423 (2010)

4. Cooper, G.F.: Nestor: A computer-based medical diagnostic aid that integrates causal and probabilistic knowledge. Tech. rep., Stanford University CA, Dept of Computer Science (1984)
5. Glymour, C., Zhang, K., Spirtes, P.: Review of causal discovery methods based on graphical models. Frontiers in Genetics 10, 1–15 (2019)
6. He, Y., Jia, J., Yu, B.: Counting and exploring sizes of Markov equivalence classes of directed acyclic graphs. The Journal of Machine Learning Research 16(1), 2589–2609 (2015)
7. Heckerman, D., Geiger, D., Chickering, D.M.: Learning Bayesian networks: The combination of knowledge and statistical data. Machine learning 20(3), 197–243 (1995)
8. Jia, H., Wu, Z., Chen, J., Chen, B., Yao, S.: Causal discovery with Bayesian networks inductive transfer. In: International Conference on Knowledge Science, Engineering and Management. pp. 351–361. Springer (2018)
9. Lauritzen, S.L., Spiegelhalter, D.J.: Local computations with probabilities on graphical structures and their application to expert systems. Journal of the Royal Statistical Society: Series B (Methodological) 50(2), 157–194 (1988)
10. Luis, R., Sucar, L.E., Morales, E.F.: Inductive transfer for learning Bayesian networks. Machine learning 79(1-2), 227–255 (2010)
11. Malinsky, D., Danks, D.: Causal discovery algorithms: A practical guide. Philosophy Compass 13(1), e12470 (2018)
12. Niculescu-Mizil, A., Caruana, R.: Inductive transfer for Bayesian network structure learning. In: Artificial Intelligence and Statistics. pp. 339–346 (2007)
13. Oates, C.J., Smith, J.Q., Mukherjee, S., Cussens, J.: Exact estimation of multiple directed acyclic graphs. Statistics and Computing 26(4), 797–811 (2016)
14. Oyen, D., Lane, T.: Bayesian discovery of multiple Bayesian networks via transfer learning. In: 2013 IEEE 13th International Conference on Data Mining. pp. 577–586. IEEE (2013)
15. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on knowledge and data engineering 22(10), 1345–1359 (2010)
16. Ramsey, J.D., Hanson, S.J., Hanson, C., Halchenko, Y.O., Poldrack, R.A., Glymour, C.: Six problems for causal inference from fMRI. Neuroimage 49(2), 1545–1558 (2010)
17. Tillman, R., Spirtes, P.: Learning equivalence classes of acyclic models with latent and selection variables from multiple datasets with overlapping variables. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. pp. 3–15 (2011)
18. Zhang, K., Schölkopf, B., Spirtes, P., Glymour, C.: Learning causality and causality-related learning: some recent progress. National Science Review 5(1), 26–29 (2018)

ISSN 1870-4069

Causal Games and Causal Nash Equilibrium

Mauricio Gonzalez Soto[1], Luis E. Sucar[1], Hugo Jair Escalante[1,2]

[1] Instituto Nacional de Astrofísica Óptica y Electrónica,
Coordinación de Ciencias Computacionales,
Mexico

[2] Centro de Investigaciones y Estudios Avanzados del IPN,
Departamento de Computación,
Mexico

{mauricio, esucar, hugojair}@inaoep.mx

Abstract. Classical results of Decision Theory, and its extension to a multi-agent setting: Game Theory, operate only at the *associative* level of information; this is, classical decision makers only take into account probabilities of events; we go one step further and consider *causal* information: in this work, we define Causal Decision Problems and extend them to a multi-agent decision problem, which we call a causal game. For such games, we study belief updating in a class of strategic games in which any player's action *causes* some consequence via a causal model, which is unknown by all players; for this reason, the most suitable model is Harsanyi's Bayesian Game. We propose a probability updating for the Bayesian Game in such a way that the knowledge of any player in terms of probabilistic beliefs about the causal model, as well as what is caused by her actions as well as the actions of every other player are taken into account. Based on such probability updating we define a Nash equilibria for Causal Games.

Keywords: causal games, causal Nash equilibrium.

1 Introduction

Causal reasoning is a constant element in our lives as it is human nature to constantly ask *why*. Looking for causes is an everyday task and, in fact, causal reasoning is to be found at the very core of our minds [40, 8]. It has been argued that the brain itself is a causal inference machine which uses *effects* to figure out *causes* in order to actively engage with the world [10, 7, 25].

An important aspect of acting in the world is being able to make decisions under uncertain conditions [8, 25]. In their seminal work [39], von Neumann and Morgenstern answered how to make choices if *rational* preferences are assumed and the decision maker knows the stochastic relation between actions and outcomes: maximize expected utility. If such relation is unknown, then J.L. Savage showed in [33] that a rational decision maker must choose *as if* is maximizing her expected utility with respect to a *subjective* probability distribution.

pp. 123–133; rec. 2019-09-17; acc. 2019-09-30

The previous results of von Neumann-Morgenstern and Savage provide formal criteria for decision making if rationality is assumed and information about the environment is considered only at the *asociative* (i.e., probabilistic) level.

These criteria are the basis for many of the techniques used in Artificial Intelligence; for example, Reinforcement Learning algorithms learn *optimal policies* that satisfy the Bellman Equations [37, 31]; therefore, any action prescribed by an optimal policy achieves the maximum expected utility as shown in [41]. Several studies have considered how human beings use causal information when making decisions with uncertain outcomes. It is known that humans tend to prefer causal information over purely probabilistic data [38]; and, in fact, it is shown in [17] that acting in the world is conceived by human beings as *intervening* on it; Therefore, it does not come up as a surprise that humans are able to learn and use causal relations while making single choices as well as in sequential decisions as shown in [8, 35, 11, 28, 24, 15, 27, 16, 32, 14].

Decision problems faced by a rational agent usually involve the decisions made by other agents as well as other, possibly unknown, factors. As seen in several applications, interactive reasoning is a fundamental aspect of human every-day reasoning and it should be addressed by any intelligent agent as argued in [25]. We consider to be of interest the multi-agent setting for causal decision making; for this reason, we consider the interaction of several rational and causal-aware decision makers whose decisions affect each other.

Game Theory [29] deals with situations in which several *rational* decision makers, or players, *interact* while pursuing some well-defined objective; the case in which decision makers make a choice simultaneously without knowing the choice made by the other players is called a *strategic game*; a well-known strategic game is the famous *prisoners' dilemma* in which two detainees must choose between confessing or remaining silent and both know the consequences of any combination of actions, what is ignored by each player is the decision made by the other.

When players ignore both the actions made by other players as well as the knowledge that made them choose a certain action, is called a game with incomplete information, or a *Bayesian Game* which was introduced by Harsanyi in [18, 19, 20]. In this work we will use the Bayesian Game model in order to study what happens when several decision makers have certain knowledge about an environment which is controlled by some, unknown but fixed, causal mechanism. We will first study one-player games, or decision problems, in which the player's actions *cause* some consequence according to some unknown causal model; for this case, we will provide a rational choice criterion which will serve us to define a Nash equilibrium in Causal Games.

2 Causation and Classical Decision Problems

2.1 Causation

The notion of causation deals with regularities found in a given environment which are stronger than probabilistic (or associative) relations in the sense that a causal relation allows for evaluating a change in the *consequence* given that a change in the *cause* is performed, while probabilistic relations only capture patterns that appear on observed data. For example, when training only on observed samples (x, y), a Bayesian Network can be equally trained as $X \to Y$ or $Y \to X$, see [2] Apendix A for a theoretical argument.

We adopt here the *manipulationist* interpretation of Causality as expressed by Woodward in [42]. The main paradigm is clearly expressed in [6] as *manipulation of a cause will result in a manipulation of the effect.* Consider the following example from [42]: manually forcing a barometer to go down won't cause a storm, whereas the occurrence of a storm will cause the barometer to go down.

We restrict ourselves to probabilistic causation and adopt the formal definition of Causality given in [36]; i.e., a stochastic relation between events which is *irreflexive, antisymmetric* and *transitive*; such formal definition is encompassed by the manipulationist interpretation. Similar descriptions of the manipulationist approach can be found in [21] and [9]. Causal inference tools, such as Pearl's do-calculus, stated in [30], allows to find the effect of an intervention in terms of probabilistic information when certain conditions are met. For what remains, we assume the *causal axioms* found in [36] with the condition known as *causal sufficiency.*

2.2 Classical Decision Theory

Classical decision making consist of a set $\mathcal{A}$ of available options to a rational decision maker, and a family $\mathcal{E}$ of uncertain events which will affect the consequence of the action made by the decision maker; any knowledge by the decision maker of such uncertainties is available only at the associative, or probabilistic, level of information. We now state the formal framework for classical decision making as we will use it in order to build upon the causal version of it:

Definition 1. *An uncertain environment is the tuple $(\Omega, \mathcal{A}, \mathcal{C}, \mathcal{E})$. Where $\mathcal{A}$ is a non-empty set of available actions, $\mathcal{C}$ a set of consequences and $\mathcal{E}$ an algebra of events over Ω.*

When we consider the preferences of some decision maker over the set of consequences of some uncertain environment we have a Decision Problem under Uncertainty:

Definition 2. *A Decision Problem under Uncertainty is an uncertain environment $(\Omega, \mathcal{A}, \mathcal{C}, \mathcal{E})$ plus a preference relation $\succeq$ defined over $\mathcal{C}$.*

2.3 Causal Environments and Causal Decision Problems

In this section we define a Causal Environment to be an *uncertain environment* for which there exists a Causal Graphical Model (CGM) $\mathcal{G}$ which controls the environment. Details on CGMs can be found in [22].

Definition 3. *A Causal Environment is a tuple $(\Omega, \mathcal{A}, \mathcal{G}, \mathcal{C}, \mathcal{E})$ where $(\Omega, \mathcal{A}, \mathcal{C}, \mathcal{E})$ is an uncertain environment and $\mathcal{G}$ is a CGM such that the set of variables of $\mathcal{G}$ correspond to the uncertain events in $\mathcal{E}$.*

2.4 Rational Choice in Causal Environments

Consider a decision maker who knows that any action she takes will *cause* a certain action, but she does not explicitly knows the form of such causal relation, she only have probabilistic beliefs about such relation. We define in this section a formal framework for studying such situations.

Definition 4. *We define a Causal Decision Problem (CDP) as* $(\mathcal{A}, \mathcal{G}, \mathcal{E}, \mathcal{C}, \succeq)$ *where* $(\mathcal{A}, \mathcal{G}, \mathcal{E}, \mathcal{C})$ *is a Causal Environment and* $\succeq$ *is a preference relation.*

For the CGM in a given CDP we will distinguish two particular variables: one corresponding to the available actions, and one corresponding to the caused outcome. We are considering that only one variable can be intervened upon and that the values of such variable represent the actions available to the decision maker; i.e., the value forced upon such variable under an intervention represents the action taken by the decision maker. The intuition behind the definition of a Causal Decision Problem is this: a decision maker chooses an action $a \in \mathcal{A}$, which is automatically inputed into the model $\mathcal{G}$, which outputs the *causal outcome* $c \in \mathcal{C}$. We say a CDP is *finite* if the set $\mathcal{A}$ is finite. We now provide a decision criterion for rationally choosing in a Causal Decision Problem.

Theorem 1. *In a finite Causal Decision Problem* $(\mathcal{A}, \mathcal{G}, \mathcal{E}, \mathcal{C}, \succeq)$*, where* $\mathcal{G}$ *is a Causal Graphical Model, we have that the preferences* $\succeq$ *of a decision maker are Savage-rational if and only if there exists a probability distribution* P_C *over a family* $\mathcal{F}$ *of causal models such that for* $a, b \in \mathcal{A}$*:*

$$a \succeq b \text{ if and only if:}$$

$$\sum_{c \in \mathcal{C}} u(c) \left(\sum_{g \in \mathcal{F}} P_g(c|do(a)) P_C(g) \right),$$

$$\geq$$

$$\sum_{c \in \mathcal{C}} u(c) \left(\sum_{g \in \mathcal{F}} P_g(c|do(b)) P_C(g) \right),$$

where P_g *is the probability distribution associated with the causal model* g*.*

Proof. The decision maker is facing an environment in which any action she takes will stochastically cause an outcome $c \in \mathcal{C}$. For this reason, the decision making is facing a very particular case of decision making under uncertainty. Assuming rationality, we invoke Savage's Theorem [33, 23, 12] to obtain a utility function u^S and a probability measure P^S which satisfy that the preference relation is represented by the expectation of u^S with respect to P^S.

In such a causal environment, the CGM $\mathcal{G}$ contains all of the information which connects actions, uncertain events and outcomes, and noting that we can identify any action a with $\{c_j | E_j : j \in J\}$ where J a countable set of indexes [3] we have that:

$$\mathbb{E}_{P^S}[u(c)] = \sum_{j \in J} u(c_j) P^S(E_j).$$

For each action $a = \{c_j|E_j : j \in J\}$, $P^S(E_j)$ is the probability of *causing* consequence c_j by choosing action a. In order for the decision maker to find the probability of a certain consequence c_j given that an action a is performed then she must have in mind a single causal model g and a way to assign probabilities over a family of causal models; i.e., the uncertainty component $P^S(E_j)$ is formed by two parts: a distribution P_C which represents the degree of belief of the decision maker about a specific model g being the true one, and within g, a distribution P_g used to calculate the probability of causing some consequence c_j given that action a is chosen. Using the Caratheodory Extension Theorem [1] a probability measure P_C whose support is a sufficiently general family of causal models $\mathcal{F}$ can be shown to exist. For $g \in \mathcal{F}$, the decision maker considers g to be the true causal model with probability $P_C(g)$, and within g, we use the classical von Neumann-Morgenstern Theorem in order to obtain the best action (see section 4.1 of [30] for details). Let P_g the probability distribution associated with the causal model g.

Then:

$$\mathbb{E}_{P^S}[u(c)] = \sum_{j \in J} u(c_j) P^S(E_j), \tag{1}$$

$$= \sum_{j \in J} u(c_j) \left(\sum_{g \in \mathcal{F}} P_g(c_j|do(a)) P_C(g) \right). \tag{2}$$

We have shown what is the expected utility for some action $a \in \mathcal{A}$, and by Savage's Theorem the result follows.

2.5 Interpretation

Theorem 1 says that a decision maker who faces a Causal Decision Problem is considering a probability distribution P_C over a family $\mathcal{F}$ and, within each structure, using the term $P_g(c|do(a))$ in order to find the probability of obtaining a certain consequence given that the intervention $do(a)$ is performed; in this way, the optimal action a^* is given by:

$$a^* = \operatorname{argmax}_{a \in \mathcal{A}} \sum_{c \in \mathcal{C}} u(c) \left(\sum_{g \in \mathcal{F}} P_g(c|do(a)) P_C(g) \right). \tag{3}$$

We note that a^* is obtained by taking into account the utility obtained by every possible consequences weighted using both the probability of causing such action within a specific causal model g and the probability that the decision maker assign to such $g \in \mathcal{F}$.

We are considering a *normative* interpretation for Theorem 1 according to which a decision maker must use any causal information in order to obtain the best possible action.

Such action must be obtained by considering the *beliefs* of the decision maker about the causal relations that hold in her environment (the distribution P_C), how such relations could produce the best action when considered *as if* they were true (distribution P_g), and the satisfaction (utility u) produced by the consequences of actions [22].

3 Classical Strategic Games

A *strategic game* is a model of a situation in which several players must take an action and afterwards they will be affected both by the outcome of their own action as well as the actions of the other players. In a strategic game it is assumed that no player knows the action taken by any other players; this is:

Definition 5. *a **strategic game** ([29]) consists of:*

- *A finite set N of n players.*
- *For each player, a nonempty set A_i of available actions.*
- *For each player, a preference relation $\succeq_i$ defined over $A = A_1 \times \cdots A_n$.*

Definition 6. *A Nash equilibrium of a strategic game $G = (N, (A)_{i \in N}, (\succeq_i)_{i \in N})$ is a vector of strategies $a^* = (a_1, a_2, ..., a_n)$ such that:*

$$(a^*_{-i}, a^*_i) \succeq_i (a^*_{-i}, b_i) \text{ for all } b_i \in A_i,$$

where $a_{-i} = (a_1, ..., a_{i-1}, a_{i+1}, ..., a_n)$.

This is, in a Nash equilibrium no player can find a better action given the actions taken by the rest of the players. We adopt here the *deductive* interpretation of an equilibrium, according to which an equilibrium results from rationality principles [4, 5].

4 Causal Games

In this section, we define a *causal strategic game* as a strategic game within a causal environment; this is, consider a *strategic* game between N rational players who are situated in a causal environment. We assume that it is *common knowledge* the causal nature of the environment as well as the rationality assumption for each player. We also assume that the causal mechanism, which represented by a Causal Graphical Model $\mathcal{G}$, remains fixed and it is unknown for each player. In this game, players ignore the actions taken by any other player, and since the causal model which controls the environment is unknown by every player, then players also ignore the information that players will use in order to take their respective actions: for this reason, we will work within the framework of *bayesian games*.

4.1 Bayesian Games

Strategic games are games in which no player knows the action taken by the other players; we now consider a type of game in which no player knows both the actions taken by any other player, nor the *private information* that made each player to take any action. Such model is called a *Bayesian Game*, introduced in [18, 19, 20]

Definition 7. *A Bayesian strategic game([29]), consists of:*

- *A finite set N of players.*
- *A finite set Ω of states of nature.*
- *For each player, a nonempty set A_i of actions.*
- *For each player, a finite set T_i and a function $\tau_i : \Omega \mapsto T_i$ the signal function of the player.*
- *For each player, a probability measure p_i over Ω such that $p_i(\tau_i^{-1}(t_i)) > 0$ for all $t_i \in T_i$.*
- *A preference relation $\succeq_i$ defined over the set of probability measures over $A \times \Omega$ where $A = A_1 \times \cdots A_n$.*

4.2 Bayesian Causal Games

In this section, we consider a *strategic game* between N rational players who are situated in a causal environment. A game is a model of a situation in which several players must take an action and afterwards they will be affected both by the outcome of their own action as well as the actions of the other players. In a strategic game it is assumed that no player knows the action taken by any other players; we also assume that the causal mechanism, which represented by a Causal Graphical Model $\mathcal{G}$, remains fixed and it is unknown for each player.

In this game, players ignore the actions taken by any other player, and since the causal model which controls the environment is unknown by every player, then players also ignore the information that players will use in order to take their respective actions: strategic games of this type are called *Bayesian Game*, introduced in [18, 19, 20]. In the games we will consider, the uncertainty of every player consists of two levels: on a first level, the true causal model $\mathcal{G}$; on a second level, what an action $do(a)$ causes if a certain CGM ω is considered to be the causal model.

We will consider the set Ω to be a family of possible causal models; in this way, $\omega \in \Omega$ being the true state of nature fixes a causal model which controls the environment in which the players make their choices. In classical Bayesian games, once $\omega \in \Omega$ is realized as the true state, then each player receives a signal $t_i = \tau_i(\omega)$ and the posterior belief $p_i(\omega|\tau_i^{-1}(t_i))$ given by $p_i(\omega)/p_i(\tau_i^{-1}(t_i))$ if $\omega \in \tau_i^{-1}(t_i)$. In the case for causal bayesian games, we must consider both the probability p_i of ω being the true state as well as the probability p_i^ω of observing a certain consequence when doing some action a_i if ω is the true model.

Following [29], we define a new game G^* in which its players are all of the possible combinations $(i, t_i) \in N \times T_i$, where the possible actions for (i, T_i) is A_i.

We see that fixing a player $i \in N$, the posterior probability $p(\omega|\tau_i^{-1}(t_i))$ induces a lottery over the pairs $(a^*(j, \tau_j(\omega)))_j, \omega)$ for some other $j \in N$. This lottery assigns to $(a^*(j, \tau_j(\omega)))_j, \omega)$ the probability $p_i(\omega)/p_i(\tau_i^{-1}(t_i))$ if $\omega \in \tau_i^{-1}(t_i)$. The classical Bayesian game will simply call a Nash equilibrium for the game G^* a Nash equilibrium of the original game; but we have the second level of uncertainty: the consequences caused by some action a through a causal model ω. We notice that the posterior probability itself induces a probability distribution defined over *actions* for each player once a *desired consequence* is fixed, this distribution, according to Theorem 1 is given by $p_i^{\omega}(c|do(a_i^*), a_{-i}^*)p_i(\omega|\tau_i^{-1}(t_i))$. This motivates the following definition of a *Causal Nash equilibrium*.

4.3 Causal Nash Equilibrium

For each player $i \in N$ in the strategic game, we define the following probability distribution over consequences:

$$p_i^a(c) = p_i^{\omega}(c|do(a_i), a_{-i})p_i(\omega) \text{ for } a \in A = A_1 \times \cdots \times A_N, \tag{4}$$

where p_i^{ω} is the probability of causing a certain consequence within a causal model ω and p_i are the player's *posterior beliefs* about the causal model that controls the environment, and *do*() is the well known intervention operator from [30]. We now define:

$$u_i^C(a) = \sum_{c \in C} u_i(c)p_i^a(c) \text{ for } a \in A = A_1 \times \cdots \times A_N. \tag{5}$$

Notice that u_i^C evaluates an action profile $a \in A$ in terms of: The knowledge about the causal model of each player represented by p_i, which allows each player to evaluate the probability of causing outcomes in terms of actions by using the *do* operator as well as the other actions taken by the other players, given by a_{-i} and the preferences of each player u_i. Using this new function, we define the equilibrium for a strategic game with causal information and Bayesian players as:

Definition 8. *An an action profile $a^* \in A$ is a Nash equilibrium for this causal strategic game if and only if:*

$$u_i^C(a^*) \geq u_i^C(a_i, a_{-i}^*) \textit{ for any other } a_i \in A_i. \tag{6}$$

This is, an action profile is a Nash equilibrium if and only if each player uses her current knowledge about the causal model of the environment in order to (causally) produce the best possible outcome given the actions taken by the other players. The existence of the Causal Nash Equilibrium is guaranteed if every A_i is a nonempty compact convex set in some $\mathbb{R}^n$ and if the preference relation induced by u_i^C is continuous and quasi-concave.

5 Conclusion

We have studied Decision Making under uncertainty in the case where a Causal Graphical Model is responsible for producing an outcome given an action (intervention) of the decision maker. We have provided a rational decision making criterion for the case in which the decision maker does not know the causal model, but has probabilistic beliefs about possible models.

Using our decision making result, and taking as a basis Harsanyi's model of a Bayesian Game in which every player has incomplete information about both the actions taken by other players as well as the information that made each player take his action we have been able to provide a definition of a Causal Nash Equilibrium in which every player is aware that there exists a Causal Mechanism that will produce some consequence once he takes an action.

Our decision making result (i.e., Theorem 1), besides motivating the Causal Nash Equilibrium, also provides an optimality criterion for learning algorithms in causal settings such as those presented in [26, 34, 13]. Our definition of Causal Equilibrium takes into account classical game theory through the incorporation of the classical von Neumann-Morgenstern utility function as well as the fundamental notion in Causation of Pearl's *do* operator.

We hope this works contributes to recent efforts of giving Causation its well deserved place in Artificial Intelligence as well as motivating further research in computational aspects of Causal Decision Theory.

References

1. Ash, R.B., Doleans-Dade, C.A.: Probability and Measure Theory. Academic Press, 2nd edn. (2000)
2. Bengio, Y., Deleu, T., Rahaman, N., Ke, R., Lachapelle, S., Bilaniuk, O., Goyal, A., Pal, C.: A meta-transfer objective for learning to disentangle causal mechanisms (2019)
3. Bernardo, J.M., Smith, A.F.M.: Bayesian theory. Wiley Series in Probability and Statistics. (2000)
4. Binmore, K.: Modeling rational players: Part i. Economics & Philosophy 3(2), 179–214 (1987)
5. Binmore, K.: Modeling rational players: Part ii. Economics & Philosophy 4(1), 9–55 (1988)
6. Campbell, D.T., Cook, T.D.: Quasi-experimentation: Design & analysis issues for field settings. Rand McNally College Publishing Company Chicago (1979)
7. Clark, A.: Surfing uncertainty: Prediction, action, and the embodied mind. Oxford University Press (2015)
8. Danks, D.: Unifying the mind: Cognitive representations as graphical models. MIT Press (2014)
9. Freedman, D.: From association to causation via regression. Advances in applied mathematics 18(1), 59–110 (1997)
10. Friston, K.: The free-energy principle: a unified brain theory? Nature Reviews Neuroscience 11(2), 127–138 (2010)
11. Garcia-Retamero, R., Hoffrage, U.: How causal knowledge simplifies decision-making. Minds and Machines 16(3), 365–380 (2006)

12. Gilboa, I.: Theory of Decision under Uncertainty. Cambridge University Press (2009)
13. Gonzalez-Soto, M., Sucar, L.E., Escalante, H.J.: Playing against nature: causal discovery for decision making under uncertainty. In: Machine Learning for Causal Inference, Counterfactual Prediction and Autonomous Action (CausalML) Workshop at ICML 2018 (2018), https://arxiv.org/abs/1807.01268
14. Hagmayer, Y., Fernbach, P.: Causality in decision-making. In: Waldmann, M.R. (ed.) The Oxford Handbook of Causal Reasoning. Oxford University Press (2017)
15. Hagmayer, Y., Meder, B.: Causal learning through repeated decision making. In: Proceedings of the Annual Meeting of the Cognitive Science Society. vol. 30 (2008)
16. Hagmayer, Y., Meder, B.: Repeated causal decision making. Journal of Experimental Psychology: Learning, Memory, and Cognition 39(1), 33 (2013)
17. Hagmayer, Y., Sloman, S.A.: Decision makers conceive of their choices as interventions. Journal of Experimental Psychology: General 138(1), 22 (2009)
18. Harsanyi, J.C.: Games with incomplete information played by "Bayesian" players, I–III Part I. the basic model. Management science 14(3), 159–182 (1967)
19. Harsanyi, J.C.: Games with incomplete information played by "Bayesian" players part II. Bayesian equilibrium points. Management Science 14(5), 320–334 (1968)
20. Harsanyi, J.C.: Games with incomplete information played by 'Bayesian'players, part III. the basic probability distribution of the game. Management Science 14(7), 486–502 (1968)
21. Holland, P.W.: Statistics and causal inference. Journal of the American Statistical Association 81(396), 945–960 (1986)
22. Koller, D., Friedman, N.: Probabilistic graphical models: principles and techniques. MIT press (2009)
23. Kreps, D.M.: Notes On The Theory Of Choice. Routledge New York (1988)
24. Lagnado, D.A., Waldmann, M.R., Hagmayer, Y., Sloman, S.A.: Beyond covariation. Causal learning: Psychology, philosophy, and computation pp. 154–172 (2007)
25. Lake, B.M., Ullman, T.D., Tenenbaum, J.B., Gershman, S.J.: Building machines that learn and think like people. Behavioral and Brain Sciences 40 (2017)
26. Lattimore, F., Lattimore, T., Reid, M.D.: Causal bandits: Learning good interventions via causal inference. In: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems 29, pp. 1181–1189. Curran Associates, Inc. (2016)
27. Meder, B., Gerstenberg, T., Hagmayer, Y., Waldmann, M.R.: Observing and intervening: Rational and heuristic models of causal decision making. The Open Psychology Journal 3, 119–135 (2010)
28. Nichols, W., Danks, D.: Decision making using learned causal structures. In: Proceedings of the Annual Meeting of the Cognitive Science Society. vol. 29 (2007)
29. Osborne, M.J., Rubinstein, A.: A course in game theory. MIT press (1994)
30. Pearl, J.: Causality: Models, Reasoning and Inference. Cambridge University Press, New York, NY, USA, 2nd edn. (2009)
31. Puterman, M.L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons, Inc., New York, NY, USA, 1st edn. (1994)
32. Rottman, B.M., Hastie, R.: Reasoning about causal relationships: Inferences on causal networks. Psychological bulletin 140(1), 109 (2014)
33. Savage, L.: The Foundations of Statistics. New York: John Wiley & Sons (1954)
34. Sen, R., Shanmugam, K., Dimakis, A.G., Shakkottai, S.: Identifying best interventions through online importance sampling. In: International Conference on Machine Learning. pp. 3057–3066 (2017)

35. Sloman, S.A., Hagmayer, Y.: The causal psycho-logic of choice. Trends in Cognitive Sciences 10(9), 407–412 (2006)
36. Spirtes, P., Glymour, C.N., Scheines, R.: Causation, prediction and search. MIT Press (2000)
37. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An introduction. MIT Press (1998)
38. Tversky, A., Kahneman, D.: Causal schemas in judgments under uncertainty. Progress in social psychology 1, 49–72 (1980)
39. Von Neumann, J., Morgenstern, O.: Theory of games and economic behavior. Princeton University Press (1944)
40. Waldmann, M.R., Hagmayer, Y.: Causal reasoning. In: Reisberg, D. (ed.) The Oxford Handbook of Cognitive Psychology. Oxford University Press (2013)
41. Webb, J.N.: Game theory: Decisions, Interaction and Evolution. Springer Undergraduate Mathematics Series (2007)
42. Woodward, J.: Making things happen: A theory of causal explanation. Oxford Studies in Philosophy of Science, Oxford University Press (2003)

ISSN 1870-4069

Evaluation of the Intervention Operator in Causal Bayesian Networks

Jenny B. Vázquez Aguirre, Nicandro Cruz Ramírez

Universidad Veracruzana, Centro de Investigación en Inteligencia Artificial,
Mexico

jennybey13@gmail.com,ncruz@uv.mx

Abstract. Causal estimation is possible through the use of Bayesian Networks; an alternative is to use an intervention operator originally proposed by Pearl. This operator acts by manipulating a variable that is a candidate cause of another, eliminating from this potential cause any influence from other variables. This tool promises to be a powerful method for estimating causality; however, as far as we know, it does not have a validation that allows us to know its scope and limitations. This work presents the implementation of the intervention operator and its evaluation in different databases. This last one tries to measure the performance of the efficiency to determine causal routes using for it the estimation of the Causal Effects and the Bayes Factor. Our results allow us to identify operator improvements to be used in a general causal estimation scheme and not only in Bayesian Networks that meet certain characteristics.

Keywords: Bayesian networks, causality, intervention.

1 Introduction

The study of causality has its origins approximately 300 years ago with the works of Hume and Kant, who tried to explain how it is that causal knowledge is acquired naturally. This gave rise to various investigations that throughout history have tried to understand and replicate causality. Artificial Intelligence (AI) is an area interested, among other things, in the study of mental processes including causal learning. Intuitively, a causal relationship occurs when X causes Y. However, we cannot always identify such relationships, as they may be spurious or contain confounding factors that may be imperceptible to observations. Pearl argues that the best way to reproduce the causal inference is through the computer by first understanding the logic of causal thinking [1].

Pearl represents the natural causal process through a graphical representation called "the ladder of causation", that explains how the natural causal process is carried out and how it could be mapped artificially [1]. Below are described the ladder of causation levels:

- Level 1. Represents the identification of causal relationships (seen or observed).
- Level 2. Refers to the interventions and predictions of the effects of the deliberate intervention of the environment.
- Level 3. Is the ability to choose from the alterations that produce the best results.

The works of Judea Pearl propose an intervention operator for the study of causal relationships located on the second level. He introduced an intervention of variables to a Bayesian Network through an operator called “do or set”, with which a new probability resulting from the intervention of variables, is obtained [2].

There are currently no reports of work in which the functioning of the operator can be appreciated in real cases; in this evaluation proposal we try to find the ideal conditions under which the operator works, we evaluate how through the intervention it is possible to access a set of new probabilities that allow to express themselves in causal terms. In this work, we made use of a set of databases provided by experts in different areas; one inclusion criterion for these was the existence of causal relationships between some of their variables.

The results found as interventions were encouraging in terms of obtaining causal probabilities; however, they raised the imminent need to create a model that learns causal relationships, supporting the expert in creating Causal Bayesian Networks automatically.

2 Theoretical Framework

2.1 Causal Bayesian Networks

Bayesian Networks (BN) were developed and introduced by Judea Pearl in the early 1980s to facilitate the prediction and abduction of intelligent AI systems [7]. BN's are models that combine graph theory and Bayesian probability. They are represented by Directed Acyclic Graphs (DAG) that allow us to know the structure of the variables hierarchically, identifying parents and children in their structure and the existing relationship between them. The structure of a network provides information on the probabilistic dependence of variables or the conditional independence of one variable given to another (or set of them) [8]. The force of influence between the connections of a network is contained in the conditional probabilities and is represented by each node given the set of its parents. The joint probability of a BN can be obtained using equation 1:

$$P(x_1,\ \dots, x_n) = \prod_j P\left(x_j | pa_j\right), \tag{1}$$

where pa_j represents the parents of node x_j in the BN.

As of the structure of a BN, it is possible to carry out association consultations (Bayesian inference), which are supported by the criterion of d-separation to verify the conditional independence in the connections. This paper will not address details about BNs, such as the d-separation criterion and Markovian parents. If you are interested in consulting these issues, a more detailed review is given in [3].

A Causal Bayesian Network (CBN) can be understood as a BN, with the property that the parents of each node represent a direct cause of it. From equation (1), we can assume that the parents of the variable X_j, are its direct causes. Otherwise, if there are no parents, the marginal probability $P(X_j)$ must be used.

Definition 1. Causal Bayesian Network [3]

Let $P(v)$ be a probability distribution on a set V of variables, and let $P_x(v)$ denote the distribution resulting from the intervention $do(X = x)$ that sets a subset X of variables to constants x. Denote by P_* the set of all interventional distributions $P_x(v)$, $X \subseteq V$, including $P(v)$, which represents no intervention (i.e., $X = 0$). A DAG G is said to be a causal Bayesian network compatible with P_* if and only if the following three conditions hold for every $P_x \in P_*$:

1. The probability distribution P_x is Markov relative with the DAG G [6].
2. The probability of all the variables that are part of an intervention is equal to 1 for the value established in: $P_x(v_i) = 1$ for all $V_i \in X$ provided that $V_i = v_i$ is consistent with $X = x$ [6].
3. The probability of all the remaining variables that are not established in the intervention is equal to the original probability (the variable given by their parents). $P_x(v_i|pa_i) = P\ (v_i \left| pa_i\right.)$ for all $V_i \notin X$ provided that pa_i is consistent with $X = x$ [6].

From Definition 1, the truncated factorization $P_x(v)$ can be calculated for any intervention $do(X = x)$. Formally remaining as the equation (2):

$$P_x(v) = \prod\{\, i|V_i \notin X\} \;\; P(v_i|pa_i), \tag{2}$$

for everything v consistent with x.

According to Pearl, the construction of causal DAG has several advantages. First, the judgments required for the construction of the models are more significant and accessible. In addition, the causal models indicate how these probabilities would change when performing external interventions [3]. The formal construction of these models is based on the assumption that parent-child relationships represent autonomous mechanisms, so it is possible to make changes in those relationships without changing or affecting the other existing relationships within the network.

The $do(x)$ operator simulates physical interventions in the network, eliminating some functions of the model and replacing it with constants $X = x$ while keeping the rest of the model unchanged. Due to the assumption of autonomy, the manipulated distribution of the intervened variable is independent of the rest of the network, so a pruning process can be applied, which implies the elimination of all the arcs (parents) received by the intervening variable [4].

The difference between observing and intervening is deduced from the last statement. For example, if we wanted to observe the effect of $B = b_0$ for a model of BN $P\ (a,b,c) = P\ (a)\ P\ (b \mid a)\ P\ (c \mid b)$, the probability would be obtained from of $P\ (A,C \mid B = b_0)$. However, by applying the assumptions of autonomy and pruning, the connection between variables B and C are eliminated, obtaining:

$$P(a,do(b),c) = P(a)P(c|do(b) = b_0). \tag{3}$$

From the new probability expression, it is possible to calculate the influence of the intervened variables on their effects. The inference rules necessary for the calculation of causal probability expressions may be consulted in detail in [6].

Whenever a feasible reduction is detected for $P(y|\hat{x})$the effect of X on Y is said to be identifiable.

Definition 2.8 Identificability [2]

The causal effect of X on Y is said to be identifiable is the quantity $P(y|\hat{x})$ can be computed uniquely from the joint distribution of the observed variables. Identifiability means that $P(y|\hat{x})$ can be estimated consistently from an arbitrarily large sample randomly draw from the joint distribution.

Then, the causal effect of a variable X on another variable Y is:

Definition 2.9 Causal Effects [3]

Given two disjoint sets of variables, X and Y, the causal effect of X on Y, denoted either as $P(y|\hat{x})$ or as $P(y|do(x))$, is a function from X to the space of probability distributions on Y. For each realization x of X, $P(y|\hat{x})$ gives the probability of Y = y induced by deleting from the model of (3) all equations corresponding to variables in X and substituting X = x in the remaining equations.

Finally, the calculation of the causal effect (or average causal effect) that one variable has on another, can be calculated from equation (4):

$$EC = P\big(Y = y\big|do(\hat{x})\big) - P\big(Y = y\big|do(\hat{x}')\big), \tag{4}$$

where:

$Y = y$ is a specific value of the effect.

$\hat{x}$ is a specific value of the intervened variable.

$\hat{x}$’ is another value of the intervened variable for the same value of y.

This last equation is the one that allows extracting the real estimation of the intervention of variables in the network, marking the difference with the simple observation.

3 Materials and Methods

3.1 Materials

The selection of the experimental units was made up of a set of "causal" databases. According to the expert opinions, these contained causal relationships in some of the variables.

We used a total of 5 datasets. The first called Ecological Integrity is a database of 23 variables (22 quantitative and one qualitative) with 290,687 cases; 4 variables as possible causes and 4 effects. This dataset contains information on ecological integrity in Mexico. Two other sets contain data related to Breast Cancer. One is a prospective sample and the other a retrospective sample, 3 variables were considered as possible causes and one as an effect of each of them. These databases contain 12 variables (11 quantitative and one qualitative) with 322 and 692 cases respectively.

Another dataset contained information on Gene Expression Levels, with a total of 12 quantitative variables (3 causes and one effect) and 31 cases. Finally, the Synthetic-data-BayesiaLab database was obtained from the BayesiaLab software to validate the results of this implementation. The database has 3 variables (one cause and one effect) and a total of 1000 cases. Causes and effects were determined by the experts who provided the data. Each variable was an experimental unit and the total was 34, in each run one value the variable intervened was fixed.

3.2 Methods

The pre-processing strategy started discretizing the quantitative variables in the datasets; for this we used Weka software, and the used methods were: Discretize and CAIM. The implementation was carried out in R, the algorithms and metrics used to BN’s construction were Hill-Climbing with the BIC, and K2 metrics using the maximum likelihood estimator; the programing language used to implement the causal routes search and the new probabilities estimation is R.

To carry out the validation of the results, the causal effects were obtained by equation 4, and additionally, calculated the Bayes factor was used. The Bayes Factor (BF) is the relationship between the probability of one hypothesis and another. It can be interpreted as a measure of force in favor of a hypothesis (model) of two competing hypotheses and is denoted by equation 5 [5]:

$$BF = \frac{P(D|H_1)}{P(D|H_0)}. \tag{5}$$

The BF can take any positive value, and a way of interpreting it is given by what is indicated in Table 1.

Table 1. Interpretation of BF.

Bayes Factor	Interpretation
>100	Extreme evidence for H1
30-100	Very strong evidence for H1
10-30	Strong evidence for H1
3-10	Moderate evidence for H1
1-3	Anecdotal evidence for H1
1	No evidence
1-0.33	Anecdotal evidence for H0
0.33 – 0.1	Moderate evidence H0
0.1 – 0.03	Strong evidence for H0
0.03 – 0.01	Very strong evidence for H0
<0.01	Extreme evidence for H0

4 Methodology

A single treatment was designed for the experiment and applied to all experimental units. The treatment has only one level and consists of the intervention of a variable in the network, fixing for each run a specific value (do(x)) of the variable intervened.

Each experimental run was carried out in three stages. The first consisted of the construction of the BR; the second the search for the possible causal route and the third, the estimation of the causal probability.

Phase 1. (Construction and validation of BN's)

A BN is built from the data set using the bnlearn R library.

Once the BN has been obtained, and before the intervention, the causal relationships are validated by the expert.

If the relationships in the network do not reflect a causal match with the expert's knowledge, the parameters with which the BN is constructed can adjust - such as the metric - or indicate the permitted or restricted causal relationships that must be respected in this one.

Phase 2. (Search for Causal Routes)

From the BN's validated in Phase 2, the cause variable and the value to be intervened must be indicated, as well as the effect variable.

With the support of the causal.effect library, the search for the Causal Route (CR) in the BN is carried out.

If there is a CR, the system delivers the new probability equation, which shows the causal probabilities must be calculated. An example of the form of the equation is presented below:

$$“sum_{\{x_i,x_j\}}P\big(y\big|x_k,x_i,x_j\big)P\big(x_j\big|x_k\big)P(x_i|x_k)”, \tag{6}$$

where x_i, x_j, x_k denote CBN variables, which are not the effects. The effect is represented by the variable y.

Phase 3. (Calculation of causal probabilities)

To calculate the causal probability tables, the expression is broken down of equation 3, separating the conditional probabilities and the sum about which the calculation will run.

Calculate - from the data - the conditional probability tables for each element of the causal expression (equation 3). Carry out the normalization of each table, calculate the causal probabilities.

Once the probability tables are obtained, calculate the Causal Effects and Bayes Factor to find potential causes.

Finally, the probability values are compared before and after the intervention, the library "querygrain" make queries in the BN using probability propagation.

5 Results

Once the BN's were obtained, the search for possible causal routes was carried out. Figure 1 shows the BN (left) and CBN (right) for the Ecological Integrity base; the pink nodes shown in the CBN represent the set of variables that are part of the Causal Route, and that were used to obtain the causal probabilities. The variable used for this example was Landscape Transformation (landtrnas), and the variable on which its effect was calculated was Ecological Integrity (eiclas).

The new expression of probability resulting from the intervention, obtained through the inference rules, is presented below:

"sum{divfun, resistenci}P(eiclas|cropland, rangeland, irrigation, landtrans, divfun, resistenci)P(resistenci|cropland, rangeland, irrigation, landtrans, divfun)P(divfun|cropland, rangeland, irrigation, landtrans)" (7)

The causal probabilities were calculated from equation 6. This exercise allows us to appreciate the differences between intervention and observation. Table 2 shows the results of the intervention to a cause variable (do($\text{landtrans} = (0{,}2 - 0{,}4)$)) and the observation, this last one calculated through the propagation of probabilities in the BN.

From the results of Table 2, it may be thought, that the intervened variable, is a potential cause of the effect. However, this cannot be proven until calculating the Causal Effects (CE) and the Bayes Factor (BF).

All datasets were analyzed in the same way, from the generation of BN's to the obtaining of the CE and BF. Table 3 shows the results of the search of these routes, illustrating the low percentage of possible variables to intervene, from networks created with traditional AI algorithms.

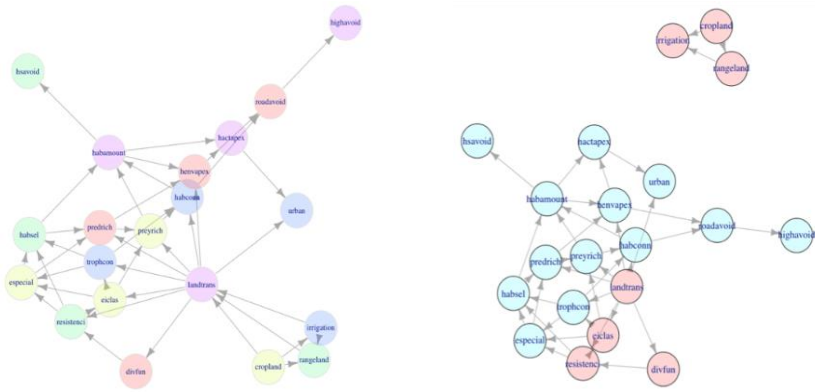


Fig. 1. Bayesian Networks (left) Causal Bayesian Network (right).

Table 2. Observed and causal probabilities for the cause variable landtrans = (0.2 - 0.4) and the eiclass effect.

Effect = eiclas	*Probability BN*	*Probability CBN*
High	0.04	0.02
Low	0.10	0.06
Medium	0.07	0.03
Transformed	0.79	0.89

Table 3. Proven databases for intervention and causal routes found in each.

Datasets	*Effects*	*Cause*	*Total experiments*	*CR - found*
Ecological integrity	4	6	24	4
Breast cancer - prospective	1	3	3	1
Breast cancer - retrospective	1	3	3	1
Genetic expression levels	1	3	3	1
Synthetic data BayesiaLab	1	1	1	1

It is important to mention that in the first experimental version (the one presented in this document), BN was not considered to be carried out manually, that is, only with the knowledge of the experts. In a second attempt, the networks were modified, which considerably increased the number of intervened variables, and causal routes found.

From the new expressions of probability, resulting from the intervention of variables in the BN's, it is possible to estimate the new probabilities that we call causal. This was shown in the previous example (Table 2). Subsequently, the BF and the CE were calculated. To obtain them, it was necessary to calculate the causal probability for two different values of the cause and the same value of the effect.

Table 4 presents the results of some of these values. The column called effect shows the name of the variables for this purpose and the value for which their causal

Table 4. Bayes Factor and Causal Effects results for intervention.

Datasets	***Effect***	***Cause***	***Interventions***	***BF***	***CE***
Ecological integrity	Eiclas *high*	landtrans	(-inf-0.2] (0.2-0.4]	25.00	0.47
Breast cancer - prospective	Outcome *Malignant*	Size	Present Absent	0.33	- 0.37
Breast cancer - retrospective	Outcome *Malignant*	Nuclear.Size	Present Absent	7.91	0.72
Genetic expression levels	APOE *(319.22 - 387.64)*	BACE1	(166.94 - 221.33) (221.33 - 493.43)	3.4	0.40
Synthetic data BayesiaLab	Outcome *Patient Recovered*	Treatment	Yes No	1.12	0.08

probability was calculated. The column called interventions shows two values of the cause in which the causal probabilities were compared; columns BF and CE show the results of the calculation of these tests.

The BF can be used as a hypothesis test to contrast two models, making a comparison of this test with the CEs allowed testing the operator's consistencies in estimating causal probabilities and their interpretation of these as potential causes. According to the interpretation corresponding to the BF this is consistent with the CE, the values between 1 and 100 obtained with the BF must correspond to positive values of the CE, and that supports the evidence of potential causes for the intervened values found in the numerator of the BF or to the left of the CE.

6 Conclusions and Future Work

The variables intervention through the operator proved to be a good method of causal estimation if the conditions for the intervention are favorable. The tests carried out after the implementation meant that its effectiveness on estimating the causal probability could be confirmed.

This work not only explores the complex issue of causality but also provides an understanding of how to observe relationships, makes estimates based on observations and interprets them; it should not be a difficult task. However, finding the set of appropriate variables that could be probable causes, and carrying out interventions that provide information on the causal force of one value over another, does have a higher degree of difficulty. This is because estimating causality adequately requires much expert knowledge, and intuition, which cannot be reflected in the calculation of the causal probabilities.

Once it has been proven that it is possible to estimate the probability of the intervention, it becomes interesting to find a way to connect the first level of the ladder of causation with the second. To do this, it will be necessary to turn towards the areas that study the process in which the learning of the causal relations occurs naturally and look at these algorithms that allow the creation of a CBN that resembles causal learning, with the same precision that is achieved naturally. This will provide artificial entities with mechanisms that learn approximate causality to the same level of a human being.

References

1. Pearl, J.; Mackenzie, D.: The book of why: the new science of cause and effect. Basic Books (2018).
2. Pearl, J.: Causal diagrams for empirical research. Biometrika, 82(4), pp. 669–688 (1995).
3. Pearl, J.: Causality: models, reasoning and inference, 29, Springer (2000).
4. Pearl, J.; Glymour, M.; Jewell, N.P.: Causal inference in statistics: A primer. John Wiley & Sons (2016).
5. Kass, R.E.; Raftery, A.E.: Bayes factors. Journal of the American statistical association, 90(430), pp: 773–795 (1995).
6. Sucar, L.E.: Probabilistic graphical models. Springer. Reverend Bayes on inference engines: A distributed hierarchical approach. Cognitive Systems Laboratory, School of Engineering and Applied Science (2015).
7. Sucar, L.E.: Redes bayesianas. BS Araujo, Aprendizaje Automático: conceptos básicos y avanzados, pp. 77–100 (2006).

Regular Papers

ISSN 1870-4069

An Open-Source Lemmatizer for Russian Language based on Tree Regression Models

Iskander Akhmetov[1,2], Alexander Krassovitsky[1], Irina Ualiyeva[1],
Alexander Gelbukh[3], Rustam Mussabayev[1]

[1] Institute of Information and Computational Technologies, Almaty,
Kazakhstan

[2] Kazakh-British Technical University, Almaty,
Kazakhstan

[3] Instituto Politécnico Nacional, CIC, Mexico City,
Mexico

i.akhmetov@ipic.kz, www.gelbukh.com

Abstract. In this article, we consider the problem of supervised morphological analysis using an approach that differs from industry spread analogs. The article describes a new method of lemmatization based on the algorithms of machine learning, in particular, on the algorithms of regression analysis, trained on the open grammatical dictionary of Russian language. Comparison of obtained results was performed with existing alternative applications that are used nowadays for addressing lemmatization problems in NLP problems for Russian language. The proposed method shows some potential for further development as it has comparable quality but uses relatively simple machine learning algorithm and at the same time is not rule based involving no manual work. The source code for our lemmatizer is publicly available.

Keywords: lemmatization, text normalization, supervised machine learning, decision tree regression models.

1 Introduction

A common problem in the analysis of texts is a large feature space that corresponds to the dictionary used in text vectorizers (90–200 thousand attribute entities). A common approach to reduce vector space is to normalize texts. It shows considerable success in reducing vector space in cases when a relatively small amount of text available in datasets leads to more balanced and accurate models. In addition to dimensionality reduction of Vector Models it also reduces the size of the index, which speeds up all text processing operations.

Normalization, namely, word lemmatization is a one of the main text preprocessing steps needed in many downstream NLP tasks. The lemmatization is a process for assigning a lemma for every word form. A lemma is the canonical (normal, dictionary) form of the lexeme. For instance, in Russian language, the normal form of a noun corresponds to the form in the nominative case in the singular (for example, “сестры” *sestry* ‘sisters’ → “сестра” *sestra* ‘sister’), for adjectives, the normal form will correspond to the nominative case, masculine, singular (“сильными” *sil'nymi* ‘by strong’) → “сильный” *sil'nyj* ‘strong’), verbs have the normal form corresponding to the infinitive (“бегут” *begut* ‘they run’) → “бежать” *bezhat'* ‘to run’) [1]. A number of approaches exists for lemmatization [2–4], employing specific language morphological rules hardcoding (rule based approach), simple dictionary methods, and up to the contemporary deep learning methods.

Morphologically rich languages are difficult to implement lemmatization, because in addition to ambiguous morphology exception rules the semantics of words is highly dependent on the attached prefixes, affixes and suffixes. Our machine learning task is complicated by that fact that Russian language is influenced by a number of essential attributes related to the internal complexity of this natural language [5].

Two popular morphological analyzers for the Russian language are the pymorphy2 [6] and MyStem [7], the comparison with which is carried out in this article. MyStem is a tool for morphological data acquisition for Russian languages, pymorphy2 is a morphological analyzer for Russian and Ukrainian languages. Both of them are freely available for non-commercial and limited commercial use. MyStem is based on a dictionary, automatically converted to trie a structure.[1] Pymorphy2 is based on OpenCorpora dictionaries [8]. Both of them are based on manually elaborated set of heuristic rules, and on corpus statistics to eliminate extra morphological variants and obtain morphology of a wide lexical coverage.

In our work, the lemmatization is treated by building tree regression models [9], i.e., by supervised automatic learning with decision trees that are constructed corresponding to language grammatical features. A number of regression models have been compared by training on a well-built dictionary. Our method is a direct supervised approach of building word lemma regressor. In principle, this approach may be applied to any language, that captures the property of high variability inside its syntactic forms. Our approach estimates the possibility of computing syntactic models using only datasets in the form of wordform–lemma dictionaries.

This paper presents a comparative analysis of the lemmatization with Pymorphy2, MyStem and a publicly available implementation[2] of the method presented in this paper. For testing purposes lemma data set from is obtained by parser of ABBYY Compreno [10]. The ABBYY tool is taken as a gold standard of comparison approach, because nowadays is considered as state of art for the industrial techs. The dataset contains 225 publications taken from the Kazakhstan news portal tengrinews.kz marked by this parser. Our lemmatization procedure can be used in various scenarios; however, it is currently considered useful mainly as a preprocessing of Russian-language media

[1] A *trie* is a prefix tree `and` a special data structure used in information retrieval (IR) tasks [14].

[2] A working demo of our lemmatizer can be found at http://isa1.pythonanywhere.com/, and the source code is available on GitHub at https://github.com/iskander-akhmetov/Lemmatization-of-Russian-Language-by-Tree-Regression-Models/.

texts. The motivation to develop this lemmatizer is because the same entities of Russian language used in the media of some countries, such as Kazakhstan, are partially different from the entities of Russian language used in the media of Russian Federation.

2 Data and Methods

2.1 Dataset

For training models, the open grammatical dictionary of Russian language (ODICT) [11] was used, consisting of more than two million word forms and their lemmas. Examples of the dictionary entries are shown in Table 1.

Table 1. ODICT example entries.

Word	Lemma
елям (*yelyam* ‘to pine trees’)	ель (*yel’* ‘pine tree’, noun)
требовали (*trebovali* ‘they required)	требовать (*trebovat’* ‘to require, verb)
фактических (*fakticheskih* ‘of factual’, plural)	фактический (*fakticheskiy* ‘factual’, adjective)

To test the method, the corpus of the Kazakhstan news portal tengrinews.kz was taken, including 225 publications (20621 words). All publications were parsed via the ABBYY parser. To test for accuracy of the regression models open-corpora dataset [8] was used.

2.2 Method

Vectorization of words is performed character-by-character into a vector of fixed length 30 (feature space) and values as an order of a letter in the Russian alphabet with following zeros. After vectorization of various word forms and their initial forms obtained from the open dictionary, two arrays of vectors were obtained, which were randomly divided into training and test samples in a ratio 67 to 33. The resulting arrays were fitted into corresponding regression models. The following regression models were used: Decision Tree, Random Forest, Extra Tree, and Bagging from sklearn Python library [12].

We use tree-based methods for regression. These involve stratifying or segmenting the predictor space into a number of simple regions. We divide the predictor space—that is, the set of possible values for $X_1, \dots, X_p$—into J distinct and non-overlapping regions, $R_1, \dots, R_J$. For every observation that falls into the region R_j, we make the same prediction, which is simply the mean of the response values for the training observations in R_j. The goal is to find boxes $R_1, \dots, R_J$ that minimize the RSS, given by

$$\sum_{j=1}^{J}\sum_{i\in R_j}(y_i - \hat{y}_{R_j})^2,$$

where $\hat{y}_{R_j}$ is the mean response for the training observations within the j-th box.

Bagging, random forests, and boosting use trees as building blocks to construct more powerful prediction models. Each of these approaches involves producing multiple trees, which are then combined to yield a single consensus prediction.

3 Evaluation

We compared four variants of our regressor lemmatizers with the Pymorphy2 and MyStem. The performance of the lemmatizers was evaluated using the accuracy metric, roughly, the proportion of correct answers given by a lemmatizer. Table 2 presents the results for all regressor lemmatizers for four testing sets.

Table 2. Accuracy of regressor lemmatizers and available alternatives.

Regressor tagger	Cross-validation (5 folds, 2,337,988 words)	Train / Test split 67% / 33% (23,37,988 words)		ABBYY corpus check (20,621 words)	Open-Corpora check (347,409 words)
		Train	Test		
Pymorphy	–	–	–	0.8181	0.8967
MyStem	–	–	–	0.7250	0.8208
DecisionTree	0.3466	0.7296	0.6788	0.7561	0.7088
RandomForest	0.5991	0.8035	0.7562	0.3623	0.3556
ExtraTrees	0.6697	0.8759	0.8096	0.7544	0.6840
BaggingRegressor	0.6006	0.8045	0.7571	0.3682	0.3569

As it can be seen from Table 2, Extra Tree achieves 88% / 81%, Random Forest Regressor achieves 80% / 76%, Decision Tree Regressor 73% / 68% and Bagging Regressor, 80% / 76% on the train / test accuracy scoring. Experiments with variations on hyper parameters of the computation algorithms have shown that their optimization (i.e., a search for optimal values of tree depth and maximal splitting size) does not give essential improvement.

The cross validation showed descent result for the Decision tree algorithm and low results for the rest of the algorithms tested, but on train/test examination the results improved significantly. The rationalization behind this can be that cross validation performed on large datasets leaves testing on relatively large amount of unseen data performed several times (5 times in our case for the number of folds used) and if it happens to randomly select a difficult test set even for one time it can spoil the average significantly.

After comparing the results the models showed on ABBYY and OpenCorpora datasets, we saw that Decision Tree algorithm showed comparable results to ExtraTrees algorithm, but the obtained models differ 10 times in size: 450 MB for Decision Tree and 4.5 GB for ExtraTrees. Therefore, it was decided to use Decision Tree model for its relative simplicity and smaller size.

The reason for failure of RandomForest and Bagging Regressors on ABBYY and OpenCorpora tests is open question, but can be due to the difficulties of these algorithms with outliers and thus low robustness. Nevertheless, both of the failed algorithms are relatively hard to interpret and we leave it for further discussion and upcoming research.

Each feature receives weights according to its contribution to computed lemma. See Fig. 1, where the axes X and Y mean feature (letter position number in a word) and the computed weight, correspondingly.

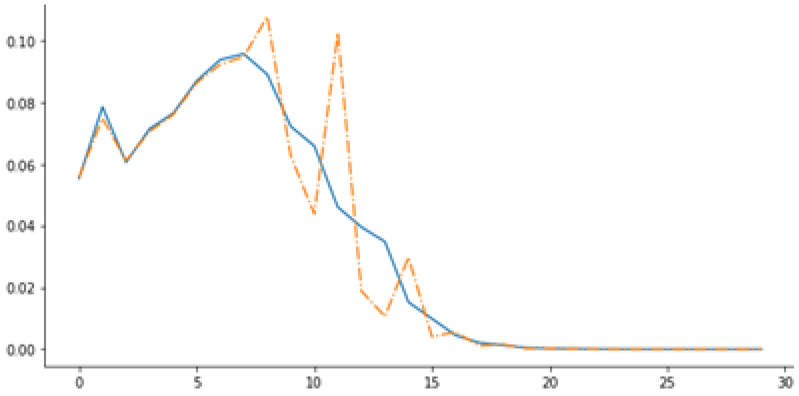


Fig. 1. The weights for features distribution (feature importance) reflect Russian word morphology (prefix, root, suffixes, etc.) are shown for Random Forest (dotted line) and Decision Tree (continuous line) regression models.

As can be seen from the figure, the model attributes more weight and thus account for importance of the beginning and especially the middle of the word - in Russian language it is usually the stem part [15]. Low significance of the letters in the positions 10 through 30 can be explained by the fact that the average word length in Russian is 10 letters and words of length more than 15 symbols are very rare.

Table 3. Comparison with alternative lemmatizers.

Lemmatizer	ABBYY corpus check (20621 words)	OpenCorpora check (347409 words)
Decision Tree Lemmatizer	**0.7561**	**0.7088**
Pymorphy2	0.8181	0.8967
MyStem	0.7250	0.8208

Despite the act that the proposed algorithm was less accurate than Pymorphy2 on ABBYY test and was left behind on the OpenCorpora test by both Pymorphy2 and MyStem, it is based on a relatively simple machine learning technique and ancient algorithm involving not much computational resources and still eliminates lots of manual rule hardcoding workload.

3.1 Error Analysis

In order to evaluate the performance of the method, the authors' lemmatizer was compared with the MyStem and Pymorphy2 lemmatizers, using the ABBYY parser to provide the testing data set. The number of wrongly lemmatized words is compared and shown by Venn diagram for these three lemmatizers by using the ABBYY test dataset; see Fig. 2.

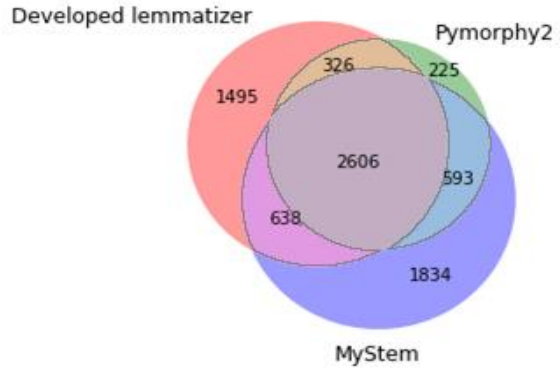


Fig. 2. The total number of errors and number of mutual errors in the testing dataset (20621 words) for our Developed lemmatizer (Decision Tree Regressor), MyStem and pymorphy2 are shown.

All three lemmatizers share 2,606 errors. The largest number of errors peculiar only to special lemmatizer belongs to MyStem (1,834 errors), which is followed by the Developed lemmatizer based on the Decision Tree algorithm (1,495 errors) and Pymorphy2 has mere 225 unique errors.

4 Conclusions

Decision tree regressors is not a silver bullet in machine learning, yet it can be a good tool in modelling language models in cases when it is too complicated to compose thousands of different rules manually. Our approaches estimate the possibility of computing syntactic models using datasets in the form of wordform–lemma dictionaries.

Number of experiments shown the developed new lemmatizer is able to solve the problem of lemmatization (especially for specific text topics), although it needs further training. Experiments can be continued for the corpus with a large number of publications and with the study of the speed of the algorithms.

Acknowledgment. This research is conducted within the framework of the grant No. BR05236839 "Development of information technologies and systems for stimulation of personality's sustainable development as one of the bases of development of digital Kazakhstan."

References

1. Smirnov, I.V.: Introduction to natural language analysis. RUDN University, Institute for Systems Analysis RAS, Moscow (2014)
2. Dave, R., Balani, P.: Survey paper of different lemmatization approaches. In: International Journal of Research in Advent Technology Science and Technology. Special Issue: First International Conference on Advent Trends in Engineering, Science and Technology, ICATEST 2015, pp. 366–370 (2015)
3. Ozturkmenoglu, O., Alpkocak, A.: Comparison of different lemmatization approaches for information retrieval on Turkish text collection. In: INISTA 2012, International Symposium on Innovations in Intelligent Systems and Applications (2012)
4. Plisson, J., Lavrac, N., Mladenić, D.D.: A rule based approach to word lemmatization. Proc. 7th International Multiconference on Information Society (2004)
5. Plungjan, A., Ljashevskaja, O.N., Sichinava, D.V.: On modern Russian language corpus morphological standard. Nauchno-tehnicheskaja informacija (Scientific and Technical Information). Series 2, Informacionnye processy i sistemy (Information processes and systems), pp. 2–9 (2005)
6. Korobov, M.: Morphological analyzer and generator for Russian and Ukrainian languages. In: Communications in Computer and Information Science, pp. 330–342 (2015)
7. Segalovich, I.: A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine. In: MLMTA 2003, International Conference on Machine Learning: Models, Technologies and Applications (2003)
8. Otkrytyj korpus (OpenCorpora), http://opencorpora.org/. Accessed 21 Jan 2019
9. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Mach. Learn. **63**(04), pp. 3–42 (2006)
10. ABBYY Company, https://www.abbyy.com/ru-ru/science/technologies/compreno/. Accessed 21 Jan 2019
11. Zaliznjak, A. A. Open grammar dictionary of the Russian language. http://odict.ru. Accessed 21 Jan 2019
12. Bird, S., Klein, E., Loper, E.: Natural language processing with Python. 2nd edn. O'Reilly (2017)
13. James, G., Witten, D., Hastie, T., Tibshirani, R.: An Introduction to Statistical Learning: with Applications in R, Springer Texts in Statistics, Springer (2013)
14. De La Briandais, R.: File searching using variable length keys. In: IRE-AIEE-ACM'59 (Western), Papers Presented at March 3–5, 1959, Western Joint Computer Conference. Association for Computing Machinery, pp. 295–298 (1959)
15. Sidorov, G.O.: Lemmatization in automatized system for compilation of personal style dictionaries of literary writers. In: Word of Dostoyevsky, Moscow, Russia, Russian Academy of Sciences, pp. 266–300 (1996)

Electronic edition
Available online: http://www.rcs.cic.ipn.mx