CREPO: An Open Repository to Benchmark Credal Network Algorithms

Rafael Cabañas RCABANAS@IDSIA.CH
Alessandro Antonucci ALESSANDRO@IDSIA.CH
Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), Switzerland

Abstract

Credal networks are a popular class of imprecise probabilistic graphical models obtained as a Bayesian network generalization based on, so-called *credal*, sets of probability mass functions. A Java library called CREMA has been recently released to model, process and query credal networks. Despite the NP-hardness of the (exact) task, a number of algorithms is available to approximate credal network inferences. In this paper we present CREPO, an open repository of synthetic credal networks, provided together with the exact results of inference tasks on these models. A Python tool is also delivered to load these data and interact with CREMA, thus making extremely easy to evaluate and compare existing and novel inference algorithms. To demonstrate such benchmarking scheme, we propose an approximate heuristic to be used inside variable elimination schemes to keep a bound on the maximum number of vertices generated during the combination step. A CREPO-based validation against approximate procedures based on linearization and exact techniques performed in CREMA is finally discussed.

Keywords: Probabilistic graphical models, credal networks, Bayesian networks, variable elimination, convex hull, imprecise probability.

1. Introduction

Probabilistic graphical models are popular tools for modern AI and machine learning [20]. Although mostly based on classical probability theory, graphical models relying on more general formalisms have been also proposed [9, 27]. Among them, *credal networks* [12] display a clear and interpretable semantics, directly extending that of Bayesian networks [25]. Moreover, credal networks have been recently used to address unidentifiable queries in structural causal models [28]. The counterpart of such interpretability and expressiveness is the intractability of exact inferences. Being a generalization of Bayesian network inference [11], credal network inference is NP-hard and, unlike the case of Bayesian networks, might remain NP-hard even for simpler topologies [23]. Nevertheless, a number of approximate algorithms has been proposed in the literature (see [24] for a recent survey), thus making possible to practically use these models in applications (e.g., [2, 3, 16]).

So far, the software implementations of these algorithms are mostly developed by different research groups and based on different tools and formats, this preventing extensive benchmarking and comparison. Yet, a Java library for credal network modelling, processing and inference, called CREMA, has been recently released [17]. The main goal of CREMA is to provide a standard and unifying framework to develop, test and compare existing credal network algorithms and promote the development of novel schemes. To fully accomplish this plan, we present here CREPO, an open and extensible repository including credal network specifications and the results of exact inferences on such models. We also deliver a Python tool to easily load CREPO data and interact with CREMA in order to easily evaluate and compare credal network algorithms. Besides the two algorithms already embedded in CREMA, i.e., a credal version of the variable elimination scheme, and an approximate procedure based on a reduction to linear programming [5], we finally present a very first example of CREPO-based benchmarking in CREMA for a novel heuristic algorithm. The heuristic is applied inside a variable elimination scheme: after any combination step, a geometric reduction is applied to keep a bound on the number of vertices, thus preventing any exponential blow up. This corresponds to an anytime procedure with respect to the value of such bound.

2. Credal Networks Inference

The (discrete) variables $\boldsymbol{X} := (X_1, \ldots, X_n)$ of a credal network are in one-to-one correspondence with the nodes of an acyclic directed graph $\mathcal{G}$. Variable X_i takes its values in $\mathcal{X}_i$, x_i being its generic value. Let Pa_{X_i} denote the *parents* of X_i according to $\mathcal{G}$. In a Bayesian network, a joint mass function $P(\boldsymbol{X})$ is obtained from the conditional probability tables $\{P(X_i|\mathrm{Pa}_{X_i})\}_{i=1}^n$. The Markov condition for the directed graph $\mathcal{G}$ induces a number of conditional independence relations leading to the factorization $P(\boldsymbol{x}) = \prod_{i=1}^n P(x_i|\mathrm{pa}_{X_i})$. A credal network is just a direct generalization of this formalism: each probability mass function $P(X_i|\mathrm{pa}_{X_i})$ is replaced by a (so-called credal) set of probability mass functions. Notation $K(X_i|\mathrm{pa}_{X_i})$ is used. A joint credal set $K(\boldsymbol{X})$ is consequently obtained by taking all joint mass functions factorizing as a Bayesian network over

the same graph and allowing the conditional probability mass functions of each conditional probability table to take their values from the corresponding credal sets.[1] Marginal inference in a credal network is intended as the computation of the lower and upper bounds of the probability of a single variable with respect to the local credal sets, e.g., $\underline{P}(x_q) := \min_{P(X_i|\mathrm{Pa}_{X_i}) \in K(X_i|\mathrm{Pa}_{X_i})} \sum_{\boldsymbol{x} \setminus \{x_q\}} \prod_{i=1}^{n} P(x_i|\mathrm{pa}_{X_i})$. Conditional queries, e.g., $\underline{P}(x_q|x_E)$ are analogously defined. Those queries remains unaffected by the convex closure of the local credal sets [1]. Here we only consider finitely-generated models corresponding to convex sets with a finite number of vertices. As credal networks generalize Bayesian networks, the hardness of those tasks for Bayesian networks [11], also concerns credal network inference. Yet, unlike the case of Bayesian networks, the inference remains hard even on simple topologies [23], with the general task belonging to higher complexity classes [15]. To better understand this point, consider the classical *variable elimination* (VE) scheme for Bayesian networks [19], where sums and products involved in the computation of the target are swapped in order to reduce the size of the joint models involved in the sums. For an optimal choice of the (elimination) order in which the sums are performed, the complexity of the procedure is bounded by the exponential of the so-called *treewidth* of graph $\mathcal{G}$ [20]. The same scheme in a credal network require each combination to be performed separately for all the elements of the credal sets associated with the different variables. Although this can be done by considering only the vertices of the credal sets, this might be not sufficient to prevent an exponential blow up. To hinder such growth, the convex hull of any combination can be computed and the inner points removed. Yet, this does not provide theoretical guarantees, thus justifying the hardness results for credal network inference even for bounded treewidth topologies. For this reason approximate schemes have been proposed. One of the most popular is *ApproxLP*, a technique based on a linearization of the optimization required by the inference, that allows to reduce the query to a sequence of linear programming tasks and Bayesian network inferences on a model with the same topology [5].

3. The CREPO Repository

Let us introduce CREPO, an open and extensible repository of synthetic credal networks, especially designed for the benchmarking of inference algorithms. We first describe its main features and then discuss a number of relevant issues we addressed during its preparation.

1. This approach to credal network modelling, based on the notion of *strong independence*, allows for a direct sensitivity analysis interpretation of the results based on these models [12]. The most popular alternative is based on the notion of epistemic irrelevance [13]. See [23] for a discussion on the relations between the two approaches.

Technical features. CREPO is stored at Github to simplify its distribution and versioning.[2] This is important, as we expect to extend the repository, which now includes only relatively small and simple models (≤ 10 nodes, cardinalities between two and three, indegree and number of vertices for credal sets between two and six) in order to make possible to quickly compute exact inferences with the credal VE. CREPO includes 378 randomly generated networks. The models are specified in UAI-like formats developed for credal networks.[3] The repository also contains the results of exact inferences based on the credal VE and the approximate results achieved with ApproxLP. Although the benchmark data can be obtained with any existing git tool, a Python package[4] was developed to easily access the repository and interact with CREMA as shown in Figure 1.

```
import crepobenchmark as crb
# Download the inference results
data = crb.get_benchmark_data()
# Save a model from the repository
crb.save_model("vs_n4_mID2_mD6_mV4_nV2-1",
               "model.uai")
# Run exact inference
crb.run_crema("model.uai", target=0)
```

Figure 1: Python code to access CREPO.

Vertices and Linear Constraints. The local credal sets of a credal network can be assumed to be convex. Thus, we describe a (finitely generated) credal set by explicitly listing their vertices (so-called *V-representation*) or by a finite set of linear constrains on its points (*H-representation*). Algorithms based on credal VE cope with V-representations, as products and sums are achieved separately for all the possible combination of the vertices of different credal sets. Being based on linear programming with respect to local credal sets, optimization-based procedures such as ApproxLP need instead H-representation. Each CREPO network in therefore available in both representations. CREMA supports these formats and offers conversion tools based on polyhedral algorithms [6]. Yet, when moving from V- to H-representation, numerical issues might arise because of a dimensionality gap: the normalization constraint to be satisfied by its points makes a credal set $K(X)$ an object in the d-dimensional space with dimensionality smaller than or equal to $d-1$, where $d := |\mathscr{X}|$. To address those issues, it suffices to find the actual dimensionality of $K(X)$, i.e., the rank of the matrix containing its vertices, and obtain an orthonormal basis for the vertices, e.g., by Gram-Schmidt orthogonalization [8]. The H-representation can be therefore computed with respect to the new basis, where the matrix with the vertices has full rank and hence does not suffer numerical issues, and eventually reformulated in the original basis.

2. https://github.com/IDSIA/crepo
3. https://crema-toolbox.readthedocs.io
4. https://pypi.org/project/crepobenchmark

Random Generation of Credal Sets. The random credal networks in CREPO have been obtained by extending a Bayesian network tool [18]. To generate random credal sets, we stick with the V-representation and a uniform sampling of mass functions from the probability simplex is iterated unless the resulting credal set has the required number of vertices. Dedicated sampling procedure are adopted [26], as trivial strategies such as sampling uniform numbers separately for each state and then normalizing does not guarantee uniformity.

Inference Tasks Selection. When solving marginal or conditional inference tasks in credal networks, an obvious preprocessing task consists in exploiting the d-separation properties of credal networks to simplify the problem. This basically consists in removing the barren nodes, remove the arcs leaving the observed variables, replacing the states of the observed variables with two states only corresponding to the actual observation and its negation, and finally take only the connected component of the model where the queried variable is located. These procedures are embedded in CREMA and the resulting graph is called *requisite* [24]. In order to have challenging inference tasks, in CREPO we select a marginal and a conditional task maximizing the size of the requisite graph. This roughly corresponds to take leaves as queries for marginal tasks, and roots for conditional ones with observations on the leaves.

4. Credal VE with k-Reduction

Let us call *reduction* any transformation of a convex set reducing its number of points and achieved by removing or replacing other points. Convex hull, intended here as the removal of the inner points after the convex closure, is a reduction not providing guarantees on the size of its output (apart from the trivial case of credal sets over Boolean variables). Although typically faster than convex hull, discarding Pareto-dominated points might encounter the same drawback [22]. This is the case also for Pareto-dominance relaxations [21]. In practice, good approximations can be obtained in this way, but it is not possible to predict in advance the minimum relaxation level keeping the running time under some threshold. To address such limitations, we propose here a k-reduction, intended as a reduction providing an upper bound on the number of points of the output. Sampling (or picking) k points from a credal set or from its vertex is an obvious k-reduction, while a more sophisticated approach is depicted in Figure 2 and defined as follows.

Definition 1 (k-reduction) *Given convex set K in input, take its vertices $e(K)$. Find the two distinct points p', p'' of $e(K)$ at minimum distance. Remove p' and p'' from $e(K)$ and return in output $e(K) \cup \{(p' + p'')/2\}$. Repeat this procedure until the number of points becomes equal to k.*

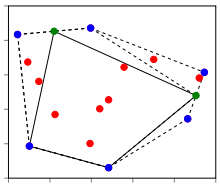

Figure 2: A 4-reduction: inner points of the original convex set are red, its vertices blue, midpoints green.

Credal VE might take exponential time even with bounded-treewidth models because of an unbounded growth in the number of points involved in the combinations. Taking the convex hull after any combination might be not enough, while adding the above k-reduction prevents such growth. As convex hull algorithms might be slow because of the high number of input points and/or space dimensionality [7], it is important to note that the procedure in Definition 1 does not require the convex hull to be executed again after the reductions, as the midpoint of the two vertices removed from the set is a vertex of the new set.[5] Moreover, those midpoints are inner points of the original convex set, thus the algorithm provides an inner approximation of credal inferences. The performance of the combination of such method with credal VE for two different threshold levels and Kullback-Leibler as (pseudo) distance in the reduction has been tested in CREMA on the CREPO tasks (version 0.0.3). Table 1 shows that the new method can nicely trade off accuracy and execution time (the *speed-up* is the ratio between execution time of credal VE and that of the algorithm).

Table 1: Benchmarking k-reduction in CREPO.

Method	**RMSE**	**Speed up**
ApproxLP	0.0058	32.107
10-reduction + CVE	0.0011	86.849
5-reduction + CVE	0.0045	147.066

5. Conclusion and Outlooks

A first repository to benchmark credal network inference has been presented together with a novel scheme to speed up credal variable elimination. As a future work, we intend to expand CREPO with new and more challenging inference tasks, and add to CREMA other algorithms, such as those in [4, 14], to promote a deeper understanding of the existing algorithms and the design of new schemes. For the k-reduction, we want to achieve a further validation and investigate the analogies between our geometrical approach and the information-theoretic method proposed in [10].

5. See extended version http://arxiv.org/abs/2105.04158.

References

[1] Alessandro Antonucci and Marco Zaffalon. Decision-theoretic specification of credal networks: A unified language for uncertain modeling with sets of Bayesian networks. *International Journal of Approximate Reasoning*, 49(2):345–361, 2008.

[2] Alessandro Antonucci, Andrea Salvetti, and Marco Zaffalon. Hazard assessment of debris flows by credal networks. In C. Pahl-Wostl, S. Schmidt, A. E. Rizzoli, and A. J. Jakeman, editors, *iEMSs 2004: Complexity and Integrated Resources Management, Transactions of the 2nd Biennial Meeting of the International Environmental Modelling and Software Society*, pages 98–103. iEMSs, 2004.

[3] Alessandro Antonucci, Ralph Brühlmann, Alberto Piatti, and Marco Zaffalon. Credal networks for military identification problems. *International Journal of Approximate Reasoning*, 50(4):666–679, 2009.

[4] Alessandro Antonucci, Yi Sun, Cassio P. De Campos, and Marco Zaffalon. Generalized loopy 2U: a new algorithm for approximate inference in credal networks. *International Journal of Approximate Reasoning*, 51 (5):474–484, 2010.

[5] Alessandro Antonucci, Cassio P. de Campos, David Huber, and Marco Zaffalon. Approximate credal network updating by linear programming with applications to decision making. *International Journal of Approximate Reasoning*, 58:25–38, 2015.

[6] David Avis and Charles Jordan. Mplrs: A scalable parallel vertex/facet enumeration code. *Mathematical Programming Computation*, 10(2):267–302, 2018.

[7] C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22(4):469–483, 1996.

[8] Åke Björck. Numerics of Gram-Schmidt orthogonalization. *Linear Algebra and Its Applications*, 197: 297–316, 1994.

[9] Christian Borgelt, Jörg Gebhardt, and Rudolf Kruse. Possibilistic graphical models. In *Computational Intelligence in Data Mining*, pages 51–67. Springer, 2000.

[10] Andrés Cano and Serafin Moral. Using probability trees to compute marginals with imprecise probabilities. *International Journal of Approximate Reasoning*, 29(1):1–46, 2002.

[11] Gregory F. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42:393–405, 1990.

[12] Fabio G Cozman. Credal networks. *Artificial intelligence*, 120(2):199–233, 2000.

[13] Jasper De Bock. Credal networks under epistemic irrelevance. *International Journal of Approximate Reasoning*, 85:107–138, 2017.

[14] Cassio P. de Campos and Fabio G. Cozman. Inference in credal networks using multilinear programming. In *Proceedings of the Second Starting AI Researcher Symposium*, pages 50–61. Citeseer, 2004.

[15] Cassio P. De Campos and Fabio G. Cozman. The inferential complexity of bayesian and credal networks. In *IJCAI*, volume 5, pages 1313–1318. Citeseer, 2005.

[16] Hector Diego Estrada-Lugo, Marco De Angelis, and Edoardo Patelli. Probabilistic risk assessment of fire occurrence in residential buildings: Application to the grenfell tower. In *13th International Conference on Applications of Statistics and Probability in Civil Engineering*, 2019.

[17] David Huber, Rafael Cabañas, Alessandro Antonucci, and Marco Zaffalon. CREMA: A Java library for credal network inference. In Manfred Jaeger and Thomas Dyhre Nielsen, editors, *Proceedings of the 10th International Conference on Probabilistic Graphical Models (PGM 2020)*, Proceedings of Machine Learning Research, Aalborg, Denmark, 2020. PMLR.

[18] Jaime Shinsuke Ide, Fabio G. Cozman, and Fabio T. Ramos. Generating random Bayesian networks with constraints on induced width. In *ECAI*, volume 16, page 323, 2004.

[19] Jürg Kohlas. *Information algebras: Generic structures for inference*. Springer Science & Business Media, 2012.

[20] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT, 2009.

[21] Radu Marinescu, Abdul Razak, and Nic Wilson. Multi-objective influence diagrams with possibly optimal policies. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[22] Denis D Mauá, Cassio P De Campos, and Marco Zaffalon. Updating credal networks is approximable in polynomial time. *International Journal of Approximate Reasoning*, 53(8):1183–1199, 2012.

[23] Denis D. Mauá, Cassio P. De Campos, Alessio Benavoli, and Alessandro Antonucci. Probabilistic inference in credal networks: new complexity results. *Journal of Artificial Intelligence Research*, 50:603–637, 2014.

[24] Denis D. Mauá and Fabio G. Cozman. Thirty years of credal networks: Specification, algorithms and complexity. *International Journal of Approximate Reasoning*, 126:133–157, 2020.

[25] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[26] Noah A. Smith and Roy W. Tromble. Sampling uniformly from the unit simplex. *Johns Hopkins University, Technical Report*, 29, 2004.

[27] Boutheina Ben Yaghlane and Khaled Mellouli. Inference in directed evidential networks based on the transferable belief model. *International Journal of Approximate Reasoning*, 48(2):399–418, 2008.

[28] Marco Zaffalon, Alessandro Antonucci, and Rafael Cabañas. Structural causal models are (solvable by) credal networks. In Manfred Jaeger and Thomas Dyhre Nielsen, editors, *Proceedings of the 10th International Conference on Probabilistic Graphical Models (PGM 2020)*, Proceedings of Machine Learning Research. PMLR, September 2020.