Robustness in Sum-Product Networks with Continuous and Categorical Data

Rob C. de Wit R.C.DEWIT@UU.NL
Cassio P. de Campos C.DECAMPOS@UU.NL
Department of Information and Computing Sciences, Utrecht University, The Netherlands

Diarmaid Conaty DCONATY01@QUB.AC.UK
Jesús Martínez del Rincon J.MARTINEZ-DEL-RINCON@QUB.AC.UK
Centre for Data Science and Scalable Computing, Queen's University Belfast, U.K.

Abstract

Sum-product networks are a popular family of probabilistic graphical models for which marginal inference can be performed in polynomial time. After learning sum-product networks from scarce data, small variations of parameters could lead to different conclusions. We adapt the robustness measure created for categorical credal sum-product networks to domains with both continuous and categorical variables. We apply this approach to a real-world dataset of online purchases where the goal is to identify fraudulent cases. We empirically show that such credal models can better discriminate between easy and hard instances than simply using the probability of the most probable class.

Keywords: Robustness, Sum-Product Networks, Credal Sets, Classification, Fraud Detection.

1. Introduction

Sum-Product Networks (SPNs) are a class of probabilistic graphical models that allow for the explicit representation of context-specific independence [9] while retaining efficient marginal inference [7, 10]. An SPN encodes an arithmetic circuit [3]: internal nodes perform (weighted) sums and multiplications, while leaves represent variable assignments (or marginal distributions of continuous variables). SPNs can be seen as a class of mixture of univariate distributions with tractable inference [4, 8, 11].

SPNs learned from data may generalise poorly and produce unreliable and overconfident conclusions. When variables are categorical, *Credal Sum-Product Networks* (CSPNs), a class of imprecise probability models, can be used to perform a (computationally efficient) robustness analysis of SPNs for classification [1, 2, 6, 5]. However, often real-world data comes with both discrete and continuous variables, which can be used to infer an SPN. We extend CSPNs towards domains with continuous variables. A CSPN is an SPN where the weights associated with sum nodes (i.e., the numerical parameters of the model) are allowed to vary inside a closed and convex set. Continuous variables are represented in leaf nodes and are assumed to be normally distributed. An experimental analysis is conducted using data from a major online retailer, where the goal is to discriminate between fraudulent and legitimate orders. This is a multi-million market and frauds can be very costly.

2. Continuous and Categorical CSPNs

The evaluation of an SPN (i.e., the computation of its value) for a given configuration of variables can be performed by a bottom-up message propagation scheme whereby each node sends to its parent its value. Leaf nodes send a density value (continuous variables) or the result of the indicator function (categorical variables). The whole procedure takes linear time and space. Conditional probabilities for categorical variables can be obtained in linear time by evaluating the network for each value of the query variable and the given evidence (then normalising the result). For CSPNs, more intricate algorithms have been devised to compute the expectation of any function over a single categorical variable. They can be promptly adapted to handle continuous variables, since the propagation of density values is similar to the propagation of probability values. In particular, SPNs (and their inferences) do not need to be normalised, so one needs simply to take the continuous leaf nodes and compute their density values, and then to "send" these values to their parents in the SPN. This is the only required adaptation, while the procedure in the internal nodes remains the same as for categorical CSPNs and the algorithms for credal classification work just as designed before in the literature [6]. In fact, this result can be proven by realising that observed continuous variables act similarly to an observed binary categorical variable, and so we obtain the following theorem.

Theorem 1 *Computing lower conditional expectations of a function over a single categorical variable in* CSPN*s with both categorical and continuous variables takes at most polynomial time when each internal node has at most one parent.*

Because of that, credal classification (i.e., computing the set of non-dominated classes) can be done in polynomial time too. On par with previous work [6], we use

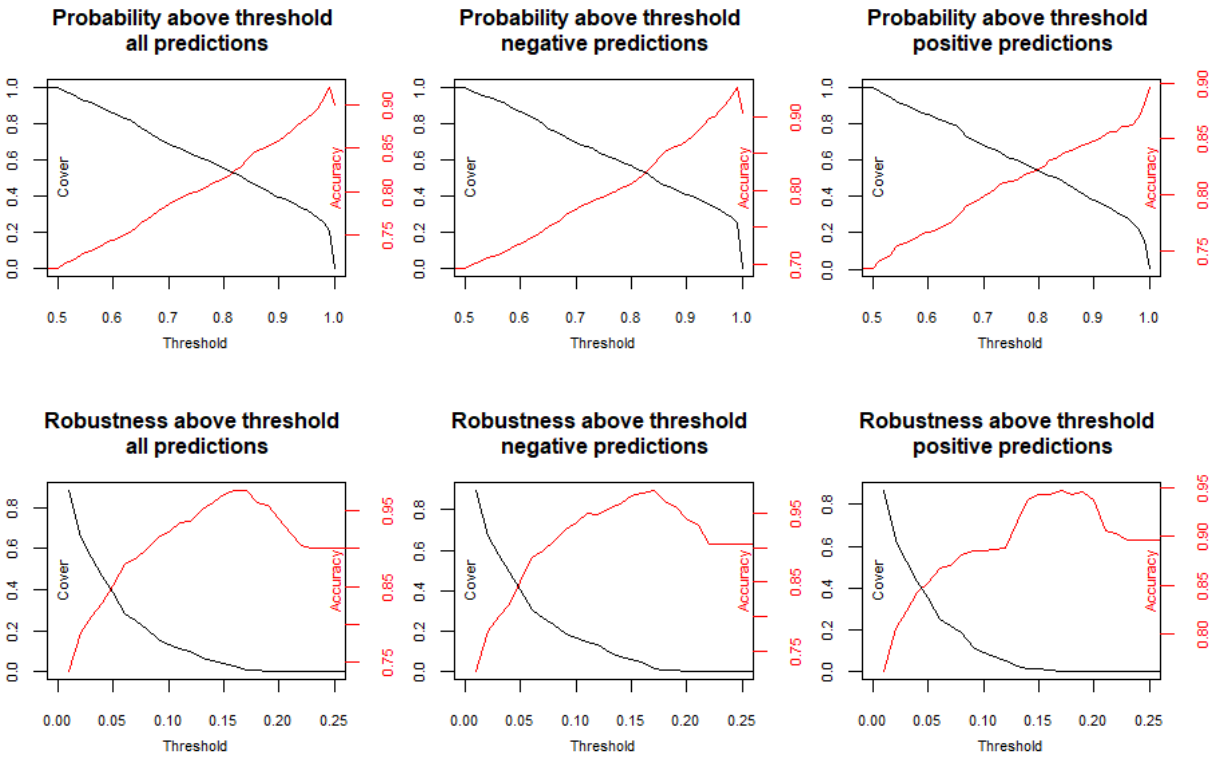

Figure 1: Graphs show cover, that is, the percentage of the cases that were classified if one only classifies the cases with measure (either probability or robustness value) above the given threshold, and accuracy over those cases. The final part of the curves is non-monotonic likely due to small sample size (very low cover).

CSPNs as means to define the robustness level of an issued classification as the most imprecise CSPN (based on the ε-contamination of the weights) for which a single class is non-dominated (one could also define some contamination for the continuous leaves, for instance by placing an interval of length ε around the *precisely* learned mean of the Gaussian distributions). The overall procedure runs a bisection with ε-contaminated CSPNs until converging to the (numerically approximate) maximum ε such that the prediction from the model is still unique (that is, all SPNs represented by the CSPN yield the same class prediction) [2, 6].

3. Case Study

Currently all orders placed at a major online retailer are evaluated through sophisticated hand-crafted business rules. The business rules can result in three outcomes: approval, outright cancellation, and manual review. A team of business analysts works around the clock to approve or cancel the orders that were flagged for manual review. For this case study, 36707 orders were collected and analysed, each of which was flagged for manual review by the business rules. For each of these orders, we collected the variables utilised by the existing expert system, the payment data for those orders, and the customer support data. This effort resulted in a total of 109 features. From those orders, the business analysts approved 18739 (51%), while 17968 (49%) were determined to be fraudulent and subsequently cancelled. The orders were labelled as follows: *true positive*: cancelled by analyst without customer complaint; *false positive*: cancelled by analyst, but the customer contacted customer support with a reasonable explanation; *true negative*: approved and subsequently paid for; *false negative*: approved, but not paid for (incurring loss to the company). The analysts achieved an accuracy of approximately 94% in this dataset. It should be noted, however, that the true accuracy could be much lower (but hard to measure), since not all customers might contact customer support upon cancellation of their order. Some of them might opt to forego their order, or simply order from a competitor.

We selected the 24 most important features to build an SPN: one continuous (that is, the price) and 23 Boolean variables (selection of them was based on variance, requiring at least a 9:1 split). The SPN was learned using the procedure of [4], with the exception that independence tests are performed using one of Chi-square, Kruskal-Wallis or Kendall

(according to the variables involved), clustering is done with the Gower distance (so as to take into account both categorical and continuous variables), and leaves related to continuous variables are forced to be normally distributed (in this study, we have used a single continuous variable). Then the robustness value is calculated per testing instance using the same approach as in [5]. Both robustness and probability of most probable class are used in order to discriminate the quality of the predictions. Figure 1 shows the results obtained by issuing a classification only when the model output is deemed *robust*, that is, either the probability value of the SPN (first row of graphs) or the robustness value from the CSPN (second row of graphs) for that particular instance was above a threshold (all possible thresholds are plotted). This is equivalent to saying that we refrain from guessing for those cases of greater indecisiveness. Based only on probabilities, no value of threshold would lead to classification results as accurate as the business analyst. Note that the analyst does not know which instances are robust or not, so they need to predict all of them. On the other hand, using robustness from CSPNs, if we only issued a decision when robustness is above the threshold of 0.1, then the model achieves the same performance as the analyst and would cover (that is, issue predictions for) about 15% of all orders. This can potentially benefit the company by reducing the time required to analyse orders flagged for review. However, those instances could well be exactly the cases for which the analyst does a very good job, while being less effective in the others. This information was however not certain at the time of writing (because the accuracy of the analyst per instance is not obtained in such a reliable manner so as to take conclusions based on them, so we must refrain from taking any strong conclusion here). Therefore, this study represents a promising preliminary analysis of CSPNs with continuous and categorical variables. Such capability extends the applicability of CSPNs to many new domains and its effectiveness will be evaluated in future work.

References

[1] D. Conaty, D. D. Mauá, and C. P. de Campos. Approximation complexity of maximum a posteriori inference in sum-product networks. In *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence*, pages 322–331, 2017.

[2] D. Conaty, J. Martinez del Rincon, and C. P. de Campos. Cascading sum-product networks using robustness. In *Proceedings of Machine Learning Research 72*, pages 73–84, 2018.

[3] A. Darwiche and G. M. Provan. Query DAGs: A practical paradigm for implementing belief-network inference. In *Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence*, pages 203–210, 1996.

[4] R. Gens and P. Domingos. Learning the structure of sum-product networks. In *Proceedings of the 30th International Conference on Machine Learning*, pages 873–880, 2013.

[5] D. D. Mauá, F. G. Cozman, D. Conaty, and C. P. de Campos. Credal sum-product networks. In *Proceedings of the 10th International Symposium on Imprecise Probability: Theories and Applications*, pages 205–216, 2017.

[6] D. D. Maua, D. Conaty, F. G. Cozman, K. Poppenhaeger, and C. P. de Campos. Robustifying sum-product networks. *International Journal of Approximate Reasoning*, 101:163–180, 2018.

[7] A. Nath and P. Domingos. Learning tractable probabilistic models for fault localization. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pages 1294–1301, 2016.

[8] R. Peharz, R. Gens, F. Pernkopf, and P. Domingos. On the latent variable interpretation in sum-product networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–14, 2016.

[9] H. Poon and P. Domingos. Sum-product networks: A new deep architecture. In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence*, pages 337–346, 2011.

[10] F. Rathke, M. Desana, and C. Schnörr. Locally adaptive probabilistic models for global segmentation of pathological oct scans. In *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 177–184, 2017.

[11] H. Zhao, M. Melibari, and P. Poupart. On the relationship between sum-product networks and Bayesian networks. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 116–124, 2015.