Robustness in Sum-Product Networks with Continuous and Categorical Data

ISIPTA 2019 - Ghent, Belgium

R. C. de Wit[1] Cassio P. de Campos[1] D. Conaty[2] J. Martínez del Rincon[2]

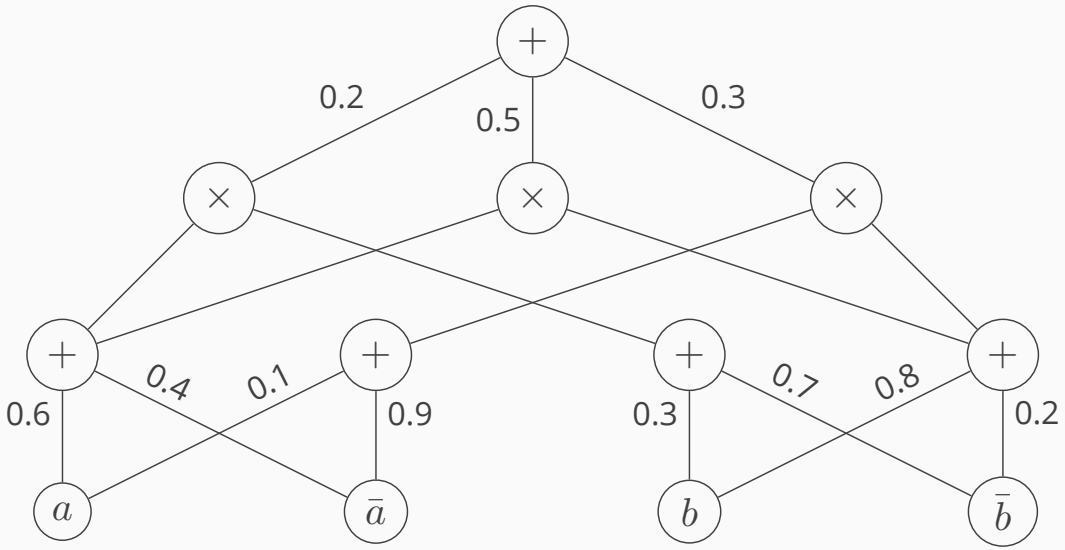
[1] Department of Information and Computing Sciences, Utrecht University, The Netherlands

[2] Centre for Data Science and Scalable Computing, Queen's University Belfast, U.K.

July 2019

SPNs

- Sum-Product Networks: sacrifice “interpretability” for the sake of computational efficiency; represent computations not interactions (Poon & Domingos 2011).
- Complex mixture distributions represented graphically as an arithmetic circuit (Darwiche 2001).



Sum-Product Network

Distribution $S(X_1, \ldots, X_n)$ built by

- an indicator function over a single variable
  - $I(X = 0), I(Y = 1)$ (also written $\neg x, y$),
- a weighted sum of SPNs with same domain and nonnegative weights (summing 1)
  - $S_3(X, Y) = 0.6 \cdot S_1(X, Y) + 0.4 \cdot S_2(X, Y)$,
- a product of SPNs with disjoint domains
  - $S_3(X, Y, Z, W) = S_1(X, Y) \cdot S_2(Z, W)$.

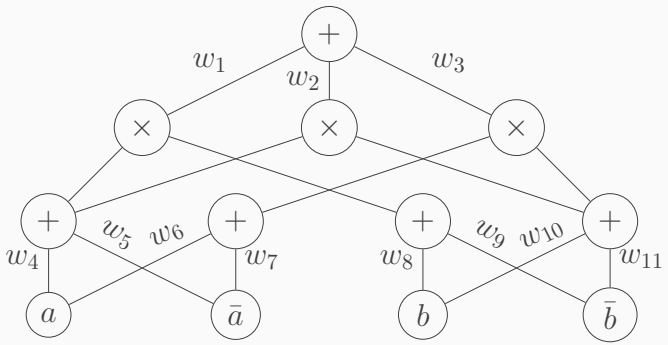
Sum-product networks - main computational points

- Computing conditional probability values is very efficient (linear time).
- Computing MAP instantiations is NP-hard in general (originally it was thought to be efficient), but efficient in some cases.

Credal Sum-Product Networks

- Robustify SPNs by allowing weights to vary inside sets.
- Class of tractable imprecise graphical models (as credal nets, they also represent a set $K(X)$).

$$
\left\{
\begin{array}{c|cc}
 & (w_1, w_2, w_3) \in \mathsf{CH}(& [0.28, 0.45, 0.27], \\
 & [0.18, 0.55, .27], & [0.18, 0.45, 0.37]), \\
 & 0.54 \leq w_4 \leq 0.64, & 0.36 \leq w_5 \leq 0.46, \\
 & 0.09 \leq w_6 \leq 0.19, & 0.81 \leq w_7 \leq 0.91, \\
 & 0.27 \leq w_8 \leq 0.37, & 0.63 \leq w_9 \leq 0.73, \\
 & 0.72 \leq w_{10} \leq 0.82, & 0.18 \leq w_{11} \leq 0.28, \\
 & w_4 + w_5 = 1, & w_6 + w_7 = 1, \\
 & w_8 + w_9 = 1, & w_{10} + w_{11} = 1
\end{array}
\right\}
$$



Credal sum-product networks - main computational points

- Computing unconditional probability intervals is very efficient (quadratic time).
- Computing conditional probability intervals is very efficient under some assumptions (quadratic time).

Credal classification

Given configurations c', c'' of variables C and evidence e decide:

$$\forall P : P(c', e) > P(c'', e) \iff \min_{w} \left(S_w(c', e) - S_w(c'', e)\right) > 0.$$

Credal classification

Given configurations c', c'' of variables C and evidence e decide:

$$\forall P : P(c', e) > P(c'', e) \iff \min_{w} \left(S_w(c', e) - S_w(c'', e)\right) > 0.$$

Credal classification with a single class variable can be done in polynomial time when each internal node has at most one parent.

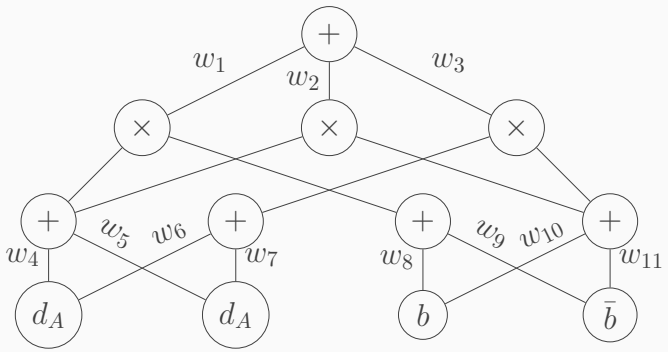
Note: Structure learning algorithms may generate sum-product nets of the above form!

Credal Sum-Product Networks with mixed variable types

Theorem 1

Credal classification with a single class variable can be done in polynomial time when each internal node has at most one parent in domains with mixed variable types (under mild assumptions).

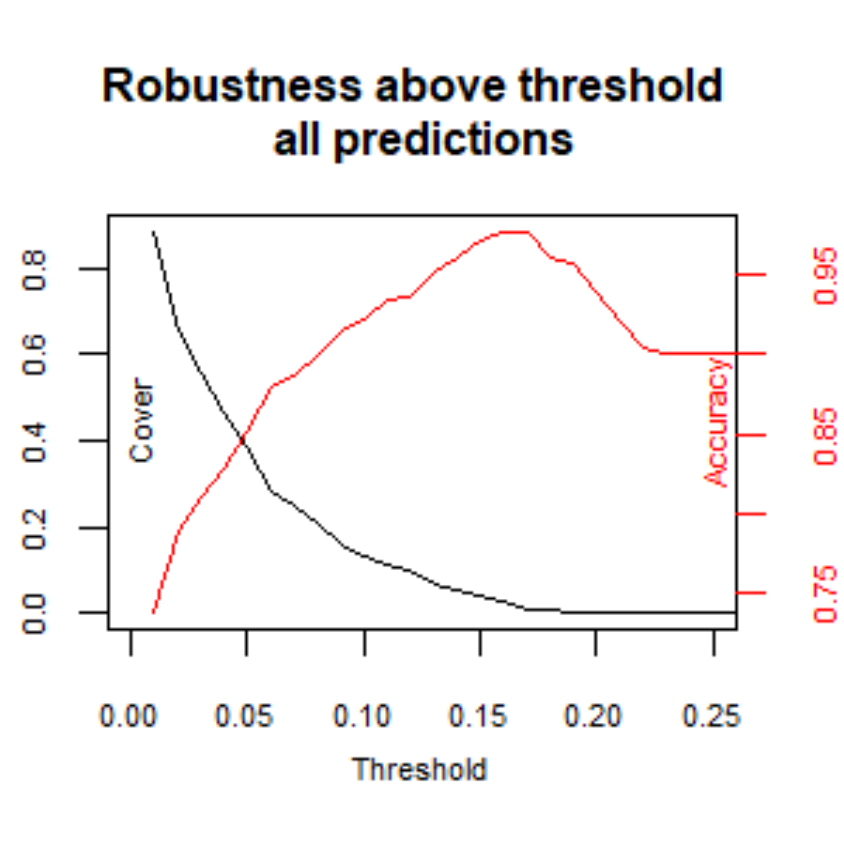
$$\left\{ \quad \left| \begin{array}{cc} (w_1, w_2, w_3) \in \mathsf{CH}(& [0.28, 0.45, 0.27], \\ [0.18, 0.55, .27], & [0.18, 0.45, 0.37]), \\ 0.54 \leq w_4 \leq 0.64, & 0.36 \leq w_5 \leq 0.46, \\ 0.09 \leq w_6 \leq 0.19, & 0.81 \leq w_7 \leq 0.91, \\ 0.27 \leq w_8 \leq 0.37, & 0.63 \leq w_9 \leq 0.73, \\ 0.72 \leq w_{10} \leq 0.82, & 0.18 \leq w_{11} \leq 0.28, \\ w_4 + w_5 = 1, & w_6 + w_7 = 1, \\ w_8 + w_9 = 1, & w_{10} + w_{11} = 1 \end{array} \right. \quad \right\}$$

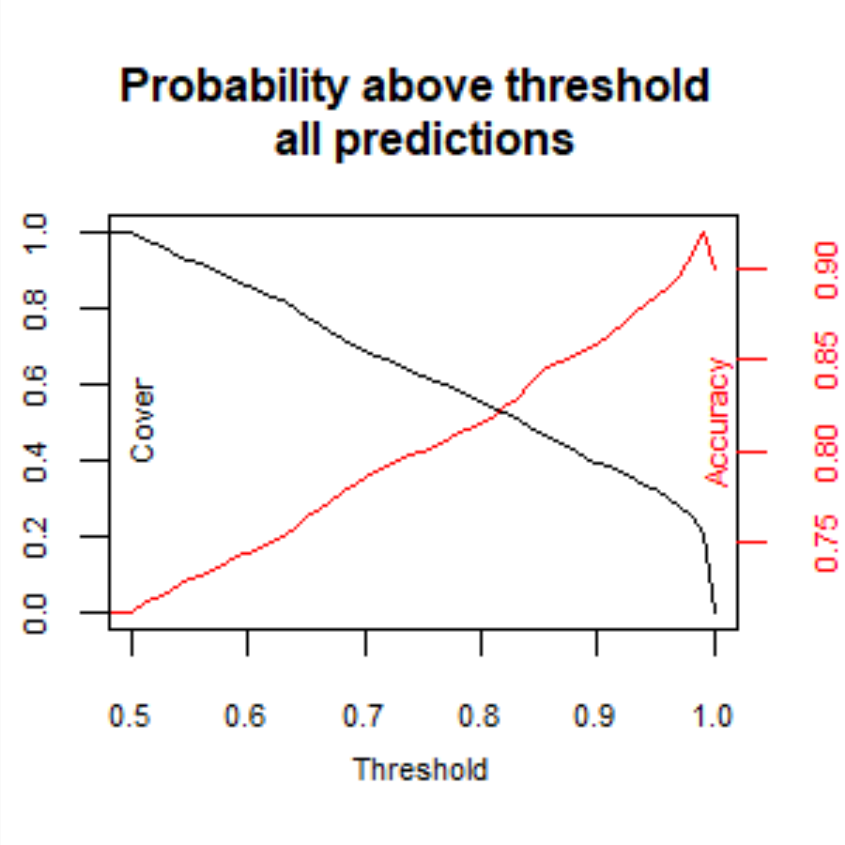


Experiments - bol.com

- 36707 orders analysed (51% legit, 49% fraud).
- Expert achieves 94% accuracy.
- 109 features reduced to 1 continuous (price) and 23 Boolean variables (with at least a 9:1 split).
- Robustness of a given testing instance is defined as the largest possible ϵ-contamination of local weights from an original sum-product network such that a single class is returned.

Preliminary results





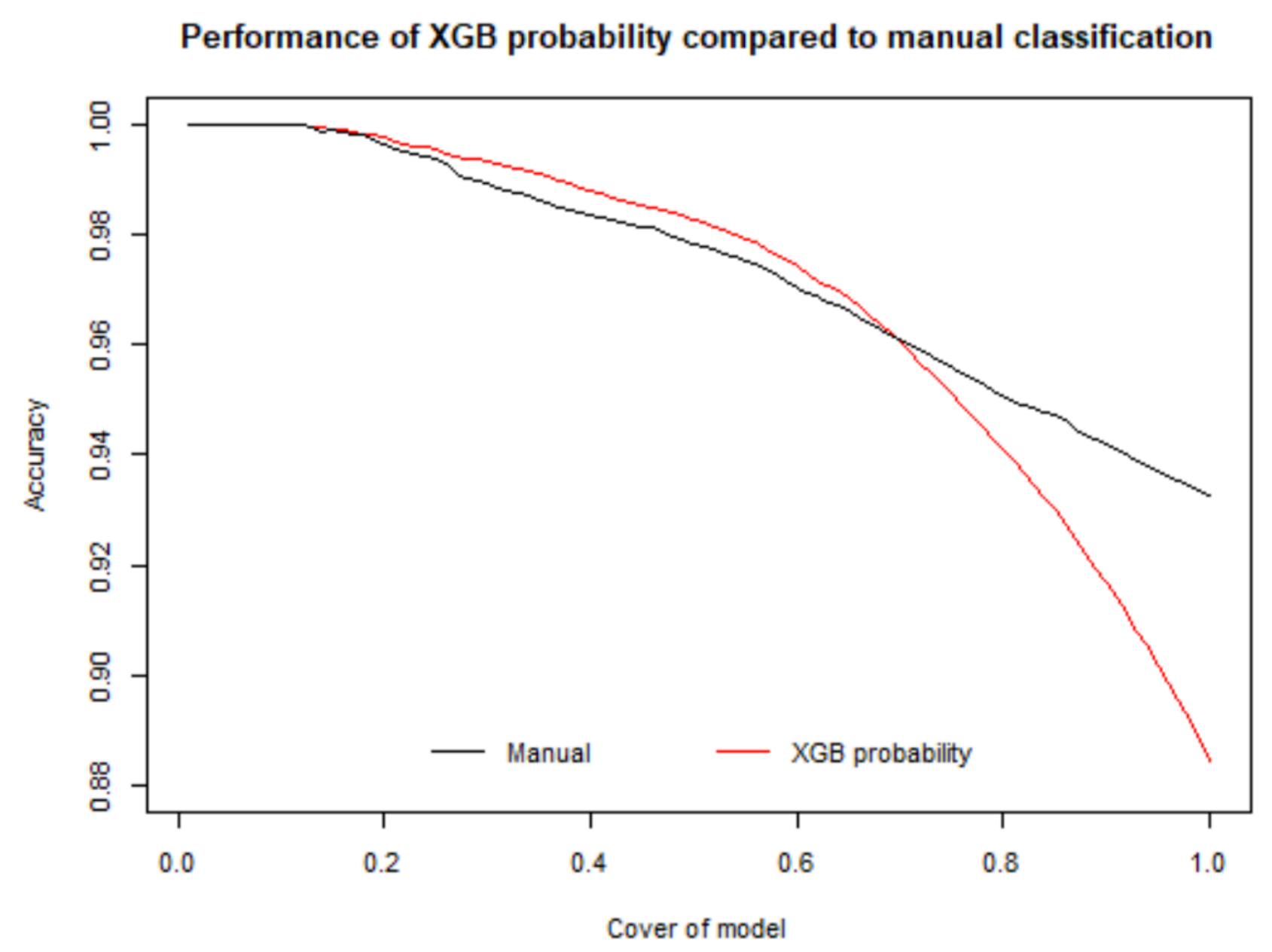
Preliminary discussion

- If we only issued automatic classification for instances with robustness above 0.1, we would achieve accuracy similar to the expert on 15% of all analysed orders.
- Robustness seems to work better than probability value itself to identify ‘easy-to-classify’ instances.

Preliminary discussion

- If we only issued automatic classification for instances with robustness above 0.1, we would achieve accuracy similar to the expert on 15% of all analysed orders.
- Robustness seems to work better than probability value itself to identify ‘easy-to-classify’ instances.
- However, this is not an obvious gain for the company: those 15% analysed orders which can be automatically classified well are typically easier and the expert may do better than 94% accuracy there (there is ongoing work to understand this better).

Gradient decision tree boosting



Thank you for your attention

cassiopc@acm.org