Modelling Multivariate Ranking Functions with Min-Sum Networks

Xiaoting Shao[1], Zhongjie Yu[1], Arseny Skryagin[1], Tjitze Rienstra[2], Matthias Thimm[2], and Kristian Kersting[1]

[1] TU Darmstadt, Germany
{xiaoting.shao, yu, arseny.skryagin, kersting}@cs.tu-darmstadt.de
[2] Universitat Koblenz, Germany
{rienstra, thimm}@uni-koblenz.de

Abstract. Spohnian ranking functions are a qualitative abstraction of probability functions, and they have been applied to knowledge representation and reasoning that involve uncertainty. However, how to represent a ranking function which has a size that is exponential in the number of variables still remains insufficiently explored. In this work we introduce *min-sum networks* (MSNs) for a compact representation of ranking functions for multiple variables. This representation allows for exact inference with linear cost in the size of the number of nodes.

Keywords: Spohnian ranking functions · Graphical models.

1 Introduction

Spohnian ranking functions are a qualitative order-of-magnitude abstraction of probability functions. These can be used to measure uncertainty using ranks [15] represented by natural numbers or ∞, which can be understood as a degree of surprise: 0 for not surprising, 1 for surprising, 2 for very surprising, and so on, and ∞ for impossible. These functions have been applied to problems of knowledge representation and reasoning that involve uncertainty but where probabilities are unknown or irrelevant, such as belief revision and non-monotonic inference [12, 7, 4]. One of the fundamental issues when using ranking functions in practice is the representation of a ranking function, which has a size that is exponential in the number of variables. The same problem arises in probabilistic modeling, where it is solved by using probabilistic graphical models (PGMs) as compact representations of probability distributions [10]. Because ranks behave much like probabilities if $+$ is replaced with *min* and $\times$ with $+$, it is sometimes possible to adapt PGMs to represent and reason about ranking functions. For example, the ranking-based counterpart of a Bayesian network is called a *ranking network* or *OCF network* [7, 2, 9, 15].

In this paper we introduce *min-sum networks* (MSNs) for compact representation of ranking functions. They are an adaptation of *sum-product networks* (SPNs) [13]. An SPN is a rooted directed acyclic graph with a recursively defined structure: a node is either a *sum-node* with weighted edges pointing to

its children; a *product-node* with non-weighted edges pointing to its children; or a leaf node representing a univariate distribution. Compared to many PGM models, SPNs support exact inference with linear cost in the size of the number of nodes. This advantage, combined with the ability to handle missing data makes SPNs to a very attractive choice for modeling any data set. Indeed, several SPN learning techniques have shown comparable or better performance than other state-of-the-art models in tasks such as image classification and natural language processing [5, 3].

The tractability of SPN's inference carries over to MSNs. More precisely, the rank of an event or proposition according to the ranking function represented by an MSN can be computed with cost linear in the size of the number of nodes in the MSN. One issue, however, is that the MSN needs to be constructed first. To address this, we propose a method to learn an MSN based on a set of observations in such a way that more probable events are less surprising (lower ranked) than less probable events.

The overview of this paper is as follows. We present the necessary basics concerning ranking theory in Section 2. In Section 3 we define min-sum networks, while Section 4 deals with the problem of learning min-sum networks. We conclude in Section 5.

2 Ranking Functions

Ranking functions are a qualitative abstraction of probability functions where events receive *ranks* [15]. A rank is a non-negative integer or ∞ and can be understood as a degree of surprise: 0 for not surprising, 1 for surprising, 2 for very surprising, and so on, and ∞ for impossible. Formally, a ranking a *ranking function* (also known as an *ordinal conditional function* or *kappa function*) is defined as follows.

Definition 1. *A ranking function over a set Ω is a function $\kappa : \Omega \to \mathbb{N}_0^\infty$ such that $\kappa(w) = 0$ for at least one $w \in \Omega$. A ranking function κ is extended to a function over propositions or events (i. e., subsets of Ω) by defining $\kappa(X) = \infty$ if $X = \emptyset$, and $\kappa(X) = min(\{\kappa(w) \mid w \in X\})$, otherwise. The* rank of A conditional on B *is denoted $\kappa(A \mid B)$ and is defined by $\kappa(A \mid B) = \kappa(A \cap B) - \kappa(B)$.*

A ranking function κ induces beliefs using the principle that A is believed if and only if the complement $\overline{A} = \Omega \setminus X$ is surprising (i.e. $\kappa(\overline{A}) > 0$). Similarly, A is believed *conditional on* B if and only if $\kappa(\overline{A} \mid B) > 0$.

3 Min-Sum Networks

Here we provide the definition of MSN, which is a ranking-based variation on SPNs [13]. We first need to introduce some notation and terminology. Random variables will be denoted by uppercase letters (e.g. X_i). We restrict our attention to Boolean random variables. We use $\mathbf{X}$ to denote a collection $\{X_1, \ldots, X_n\}$

of random variables and denote by $val(\mathbf{X})$ the set of *realisations* of $\mathbf{X}$, i.e., $val(\mathbf{X}) = \{\mathrm{T}, \mathrm{F}\}^n$ (T for true and F for false). Elements of $val(\mathbf{X})$ are denoted by lowercase boldface letters (e.g. $\mathbf{x}$). A realisation of some subset of $\{X_1, \ldots, X_n\}$ will be called *evidence*. Given a random variable X_i we use x_i and $\overline{x}_i$ to denote indicator variables. We say that x_i (resp. $\overline{x}_i$) is *consistent* with evidence $\mathbf{e}$ iff $\mathbf{e}$ does not assign false (resp. true) to X_i.

A min-sum network (MSN) over variables $\mathbf{X} = \{X_1, \ldots, X_n\}$ is an acyclic directed graph N whose leaves are the variables $x_1, \ldots, x_n, \overline{x}_1, \ldots, \overline{x}_n$. Given a node i in N we denote by N_i the subgraph rooted at node i and we denote by $Ch(i)$ the set of children of node i. The internal nodes of N are either *min-nodes* or *sum-nodes*. Each edge from a *min-node* i to another node j has an associated *weight* w_{ij} which is a non-negative integer or ∞, satisfying $min(\{w_{ij}\}) = 0$. The rank of evidence $\mathbf{e}$ according to N is defined recursively: $N(\mathbf{x}) = \sum_{j \in Ch(i)} N_j(\mathbf{x})$ if the root of N is a *sum-node*; and $N(\mathbf{x}) = min(\{w_{ij} + N_j(\mathbf{x}) | j \in Ch(i)\})$, if the root of N is a *min-node*. Further, we define a leaf L_{X_i} to consist of one *min* node, two weights w_{x_i} and $w_{\bar{x}_i}$ and two indicators x_i, $\bar{x}_i$.

3.1 Validity, Consistency and Completeness

Similar to the SPNs, we introduce the three key properties, which allows us to link MSNs with the ranking theory and ensure the error-free inference: validity, completeness, and consistency. The link is that the values of $N(\mathbf{x})$ for all $\mathbf{x} \in val(\mathbf{X})$ define a ranking function by

$$\Phi_N(\mathbf{e}) = min(\{N(\mathbf{x}) \mid \mathbf{x} \in val(\mathbf{X}), \mathbf{e} \text{ is consistent with } \mathbf{x}\}).$$

We define all three properties in the following.

Definition 2. *An MSN N over variables $\mathbf{X}$ is* valid*, iff $N(\mathbf{e}) = \Phi_N(\mathbf{e})$ for all evidence $\mathbf{e}$ of $\mathbf{X}$.*

By contrast, the definitions of completeness and consistency are the same as those of SPNs:

Definition 3. *An MSN N is complete, iff all children of the same min-node have the same scope.*

Definition 4. *An MSN N is consistent, iff there is no sum-node and variable X such that x appears in one child of this node and $\overline{x}$ in another child.*

We combine the three properties and show in the following theorem that:

Theorem 1. *An MSN is valid if it is complete and consistent.*

Proof. Since there are only two mutually exclusive configurations $\{x_i = 1, \bar{x}_i = 0\}$ and $\{x_i = 0, \bar{x}_i = 1\}$, we define the delta functions

$$\delta_{x_i} := \begin{cases} 1, & x_i = 1 \\ \infty, & x_i = 0 \end{cases} \quad \text{and} \quad \delta_{\bar{x}_i} := \begin{cases} 1, & \bar{x}_i = 1 \\ \infty, & \bar{x}_i = 0 \end{cases}.$$

So, we can express the value of the leaf L_i for any arbitrary variable X_i as

$$\begin{aligned} L_i &= \min\left(\delta_{x_i} \cdot w_{x_i}, \delta_{\bar{x}_i} \cdot w_{\bar{x}_i}\right) \\ &= \delta_{x_i} \cdot w_{x_i} \cdot \mathbb{1}_{\{\delta_{\bar{x}_i} \cdot w_{\bar{x}_i} > \delta_{x_i} \cdot w_{x_i}\}} + \delta_{\bar{x}_i} \cdot w_{\bar{x}_i} \cdot \mathbb{1}_{\{\delta_{\bar{x}_i} \cdot w_{\bar{x}_i} > \delta_{\bar{x}_i} \cdot w_{\bar{x}_i}\}}. \end{aligned} \tag{1}$$

Consequently, the leaf L_i can take the two values for the two configurations w_{x_i} or $w_{\bar{x}_i}$.

Utilizing the same idea from (1) we rewrite the *min-node* i as the partial sum as

$$\begin{aligned} &\min\left(\{w_{ij} + N_j(\mathbf{x}) | j \in Ch(i)\}\right) \\ &= \sum_{j \in Ch(i)} \left(w_{ij} + N_j(\mathbf{x})\right) \cdot \mathbb{1}_{\{\sum_{k \in Ch(i) \setminus \{j\}} (w_{ik} + N_k(\mathbf{x})) > w_{ij} + N_j(\mathbf{x})\}}. \end{aligned} \tag{2}$$

With (1) and (2) we write $N(\mathbf{x})$ as a series which will be referred to as an *expansion* of the MSN. Therefore, an MSN is valid if its expansion has the same value as $\Phi_N(\mathbf{e})$ for all evidence $\mathbf{e}$: each configuration has exactly one partial sum (condition 1), each partial sum is convergent for exactly one configuration (condition 2). From condition 2 we conclude that $N(\mathbf{x}) = w_{\mathbf{x}} < \infty$ and consequently $\Phi_N(\mathbf{e}) = \sum_{x \in \mathbf{e}} N(x) = \sum_{x \in \mathbf{e}} w_x = \sum_{k \in n(\mathbf{e})} w_k$, where $n(\mathbf{e})$ is the number of the configurations complying with condition 2. From condition 1, we conclude $n(\mathbf{e}) << |val(\mathbf{X})| = 2^n$, therefore $\Phi_N(\mathbf{e}) = \sum_{k \in n(\mathbf{e}): w_k < \infty} w_k = N(\mathbf{e}) < \infty$ and MSN is valid.

Now we prove by induction from the leaves to the root that, if the MSN is complete and consistent, then its expansion is its network series. The rest of the proof follows analogously to that one in [13], emphasising the necessity of *completeness* for *min-node* and *consistency* for *sum-node*.

3.2 Min-Sum Network Example: Wet Grass

The Wet Grass [1] example is a well-known example of probabilistic graphical models. It consists of a collection of four boolean random variables $\mathbf{X} = \{R, N, H, S\}$, where R stands for "it has been raining"; S for "Holmes' sprinkler was on"; N for "Holmes' neighbor's grass is wet"; and H for "Holmes' grass is wet". In this paper, we turn Wet Grass into a ranking example. Table 1 lists the ranks of all possible configurations of the four random variables. For instance, it is not surprising if it has not been raining, the sprinkler was off, and both lawns are not wet, i.e., $\kappa(\mathbf{x}) = 0$ for $\mathbf{x} = \{R = F, N = F, H = F, S = F\}$. However, it is impossible if it has not been raining, the sprinkler was off, but both lawns are wet, i.e., $\kappa(\mathbf{x}) = \infty$ for $\mathbf{x} = \{R = F, N = T, H = T, S = F\}$.

The ranking function of the random variables in Table 1 can be modeled with a manually designed valid MSN, shown in Figure 1. To query the rank of input evidence, a bottom-up pass needs to be operated. Denote $L_X^{w_i}$ the leaf of random variable X from the *sum-node* with weight w_i. For example, with $\mathbf{x} = \{R = F, N = T, H = F, S = T\}$, the rank of $\mathbf{x}$ according to the MSN is

Table 1. Rankings in the Wet Grass example.

R	T	T	T	T	T	T	T	T	F	F	F	F	F	F	F	F
N	T	T	T	T	F	F	F	F	T	T	T	T	F	F	F	F
H	T	T	F	F	T	T	F	F	T	T	F	F	T	T	F	F
S	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F
Rank	4	1	∞	∞	∞	∞	∞	∞	5	∞	8	2	3	∞	6	0

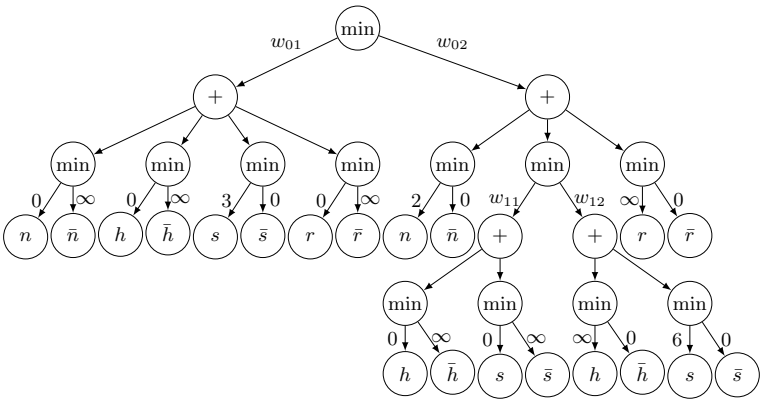

Fig. 1. An example of a valid MSN. The weights of the *min-nodes* are $w_{01} = 1$, $w_{02} = 0$, $w_{11} = 3$, $w_{12} = 0$.

calculated by

$$\begin{aligned} N(\mathbf{x}) = &\min(\{(L_N^{w_{01}} + L_H^{w_{01}} + L_S^{w_{01}} + L_R^{w_{01}}) + w_{01}, \\ &L_N^{w_{02}} + \min\{(L_H^{w_{11}} + L_S^{w_{11}}) + w_{11}, (L_H^{w_{12}} + L_S^{w_{12}}) + w_{12}\} + L_R^{w_{02}} + w_{02}\}) \\ = &\min(\{(\infty + 0 + \infty + 3) + 1, \\ &2 + \min\{(\infty + 0) + 3, (0 + 6) + 0\} + 0\}) \\ = &\min(\{\infty, 2 + 6 + 0\}) = 8, \end{aligned}$$

which means one gets very surprising if it has not been raining and the sprinkler was on, while Mr. Holmes' lawn is dry and his neighbor's is wet. Given the same evidence, the corresponding rank from Table 1 always matches the rank calculated by the valid MSN from Figure 1, i.e. $N(\mathbf{x}) = \kappa(\mathbf{x})$. Thus, the MSN in Figure 1 models exactly the ranks in Table 1.

4 Learning MSNs

A common way to estimate the parameters θ of a statistical model is to compute the Maximum Likelihood Estimation (MLE) in such a way that under the assumed statistical model the observational data is most probable. Denote $D \triangleq \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$ as the observational data, MLE is then defined as

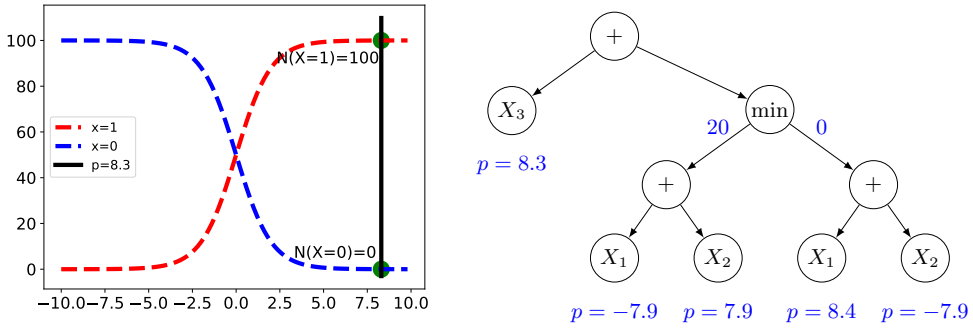

Fig. 2. Left: The ranking function for one variable. The vertical black line denotes the optimal parameter for X_3 in the MSN on the right side. Right: The MSN optimized using samples from the synthetic distribution. The parameters are marked blue.

$\hat{\theta} \triangleq \arg\max_\theta \log p(D|\theta)$ [11]. In analogy, we can learn the parameters θ of an MSN by minimizing the rank on the observational data, that is,

$$\hat{\theta} \triangleq \arg\min_\theta N(D|\theta). \tag{3}$$

We assume the training examples D are independent and identically distributed. Then we can rewrite the rank as $N(D|\theta) = \sum_{i \in m} N(\mathbf{x}_i|\theta)$. Intuitively, minimizing $N(D|\theta)$ yields θ that makes the observed data most probable under the assumed MSN. The parameters θ of an MSN consist of weights $\mathbf{w}$ at *min-nodes* and the univariate distributional parameter p in the leaf nodes. We define the ranking function on the leaves as the following form

$$(2 * (\text{sigmoid}(p) - 0.5) * (X - 0.5) + 0.5) * C, \tag{4}$$

where $C = 100$. At inference time, we round up the output to get natural numbers. Figure 2 (left) shows a plot of this ranking function.

To find the θ that minimize the objective function $N(D|\theta)$, gradient descent is commonly used if the target function is differentiable. We implemented MSNs in python and Tensorflow so that differentiation and gradients can be computed automatically by Tensorflow. We now demonstrate a concrete example using this learning method. First we construct a synthetic distribution with three Bernoulli variables X_1, X_2 and X_3 of which X_1 and X_2 are dependent on each other and X_3 is independent of any. The joint probabilistic distribution can be factorized as $P(X_1, X_2, X_3) = P(X_1) * P(X_2|X_1) * P(X_3|X_1, X_2) = P(X_1) * P(X_2|X_1) * P(X_3)$ using the chain rule and independence information. 300 samples are generated from this distribution for learning the parameters of the MSN. The sample counts for each configuration of the three variables are shown in table 2. We take an MSN with randomly initialized parameters and use gradient-based method to

Table 2. The sample counts and the ranks computed by the MSN of every possible configuration of a synthetic distribution.

X_1, X_2, X_3	0,0,0	0,0,1	0,1,0	0,1,1	1,0,0	1,0,1	1,1,0	1,1,1
rank	100	200	0	100	20	120	100	200
count	28	8	111	38	92	23	0	0

optimize the objective function 3 on the 300 samples. The optimization yields the network in figure 2 (right) with its parameters marked blue.

The ranks of all the possible configurations computed by this network are listed in table 2. The ranks of all the observed configurations are sorted as $N(0,1,0) < N(1,0,0) < N(0,1,1) = N(0,0,0) < N(1,0,1) < N(0,0,1)$, which correspond exactly to reversely sorted empirical probability of all the observed configurations $P(0,1,0) > P(1,0,0) > P(0,1,1) > P(0,0,0) > P(1,0,1) > P(0,0,1)$. Besides, we expect the ranks for the unseen configurations to be as high as possible because they are very unlikely to happen. Here, we have two unseen configurations $X_1 = 1, X_2 = 1, X_3 = 0$ and $X_1 = 1, X_2 = 1, X_3 = 1$ whose ranks are respectively 100 and 200. That means, $X_1 = 1, X_2 = 1, X_3 = 1$ is least likely to happen which is correct. But $X_1 = 1, X_2 = 1, X_3 = 0$ is more likely to happen than, for example, $X_1 = 1, X_2 = 0, X_3 = 1$, which is the only rank that does not match the empirical probability. Besides, optimization may get stuck at a local optimum which possibly again leads to ranking computations that are not consistent with empirical probabilities. We leave this challenge for future research.

By definition, the *min-nodes* encode mixtures of their children, and the *sum-nodes* assume independence between their children [14]. Take the MSN in figure 2 (right) as an example, X_3 is independent of X_1 and X_2, that means we can get the marginal ranking function of X_3 by simply removing the other independent branch. This yields an univariate ranking function with $p = 8.3$. The marginal sample counts for $X_1 = 0$ and $X_1 = 1$ are respectively 231 (111+92+28) and 69 (38+23+8), which means the rank of $X_1 = 1$ should be zero and the rank of $X_1 = 0$ should be larger than zero. Recall figure 2 (left), the vertical black line denotes the optimal parameter $p = 8.3$ for X_3 and this parameter yields $N(X_1 = 1) = 100$ and $N(X_1 = 1) = 0$, which matches the empirical probability well.

5 Conclusion and Future Work

Based on the notion of SPN, we have introduced MSN for compact representation and tractable inference with ranking functions. Ranking functions are used in models of belief revision and non-monotonic inference [12, 7, 4], and we believe that these applications may benefit from using min-sum networks for representing ranking functions. One obstacle, however, is that min-sum networks must first be constructed, and for this purpose, we proposed a method to learn a

min-sum network based on a set of observations. There is a number of directions for future work. One is to improve our learning method. In particular, Giang and Shenoy [6] have studied desirable properties for transformations from probability functions to ranking functions. An interesting question is whether a min-sum network can be learned from an (empirical) probability distribution in such a way that these desirable properties are satisfied. Another interesting question is whether min-sum networks can be constructed or learned on the basis of qualitative information. For instance, the non-monotonic inference system called *System Z* involves determining the unique "most normal" ranking function that satisfies a given knowledge base containing default rules [12]. If this ranking function can be constructed directly as a min-sum network then this network could be used to answer certain types of queries with a cost that is linear with respect to the size of the network. Finally, our approach is based on adapting SPNs for representing ranking functions. It may also be possible to adapt SPNs in a similar way to represent other representations of uncertainty, such as possibility measures, belief functions and plausibility measures [8].

References

1. Barber, D.: Bayesian reasoning and machine learning. Cambridge University Press (2012)
2. Benferhat, S., Tabia, K.: Belief change in ocf-based networks in presence of sequences of observations and interventions: application to alert correlation. In: Pacific Rim International Conference on Artificial Intelligence (2010)
3. Cheng, W.C., Kok, S., Pham, H.V., Chieu, H.L., Chai, K.M.A.: Language modeling with sum-product networks. In: Fifteenth Annual Conference of the International Speech Communication Association (2014)
4. Darwiche, A., Pearl, J.: On the logic of iterated belief revision. Artificial Intelligence (1997)
5. Gens, R., Domingos, P.: Discriminative learning of sum-product networks. In: Advances in Neural Information Processing Systems (2012)
6. Giang, P.H., Shenoy, P.P.: On transformations between probability and spohnian disbelief functions. In: Proceedings of Uncertainty in artificial intelligence (1999)
7. Goldszmidt, M., Pearl, J.: Qualitative probabilities for default reasoning, belief revision, and causal modeling. Artificial Intelligence (1996)
8. Halpern, J.Y.: Reasoning about uncertainty. MIT press (2017)
9. Kern-Isberner, G., Eichhorn, C.: Intensional combination of rankings for ocf-networks. In: The Twenty-Sixth International FLAIRS Conference (2013)
10. Koller, D., Friedman, N.: Probabilistic graphical models: principles and techniques. MIT press (2009)
11. Murphy, K.P.: Machine learning: a probabilistic perspective. MIT press (2012)
12. Pearl, J.: A natural ordering of defaults with tractable applications to default reasoning. Proceedings of Theoretical Aspects of Reasoning about Knowledge (1990)
13. Poon, H., Domingos, P.: Sum-product networks: A new deep architecture. In: International Conference on Computer Vision Workshops (2011)
14. Spohn, W.: A survey of ranking theory. In: Degrees of belief, pp. 185–228. Springer (2009)
15. Spohn, W.: The laws of belief: Ranking theory and its philosophical applications. Oxford University Press (2012)