Safe Semi-Supervised Learning of Sum-Product Networks

Martin Trapp[1,2], **Tamas Madl**[2], **Robert Peharz**[3], **Franz Pernkopf**[1], and **Robert Trappl**[2]

[1]Signal Processing and Speech Communication Laboratory, Graz University of Technology, Graz, Austria
[2]Austrian Research Institute for Artificial Intelligence, Vienna, Austria
[3]Computational and Biological Learning Lab, University of Cambridge, Cambridge, UK

Abstract

In several domains obtaining class annotations is expensive while at the same time unlabelled data are abundant. While most semi-supervised approaches enforce restrictive assumptions on the data distribution, recent work has managed to learn semi-supervised models in a non-restrictive regime. However, so far such approaches have only been proposed for linear models. In this work, we introduce semi-supervised parameter learning for Sum-Product Networks (SPNs). SPNs are deep probabilistic models admitting inference in linear time in number of network edges. Our approach has several advantages, as it (1) allows generative and discriminative semi-supervised learning, (2) guarantees that adding unlabelled data can increase, but not degrade, the performance (*safe*), and (3) is computationally efficient and does not enforce restrictive assumptions on the data distribution. We show on a variety of data sets that *safe* semi-supervised learning with SPNs is competitive compared to state-of-the-art and can lead to a better generative and discriminative objective value than a purely supervised approach.

1 INTRODUCTION

In several domains, unlabelled observations are abundant and cheap to acquire, while obtaining class labels is expensive and sometimes infeasible for large amounts of data. In such cases, semi-supervised learning can be used to exploit large amounts of unlabelled data in addition to labelled data. Examples include text [30] or image data [17, 27, 22], which are ubiquitous online, but also biological (genomics, proteomics, gene expression) data [31] and speech [28]. One of the challenges facing most semi-supervised learning approaches is scalability, many methods scale quadratically or even cubically with data set size, or require restrictive assumptions such as low dimensionality or sparsity [32, 17]. In fact, if the data violates the assumptions enforced by a learner, the use of additional unlabelled data can even degrade the classification performance.

Several approaches for semi-supervised have been proposed, including self-training, Transductive Support-Vector Machines (TSVM) [5], and graph-based methods. We refer to [32, 14] for comprehensive reviews on the state-of-the-art. As pointed out by [17], self-training is error-prone (it can reinforce poor predictions) and TSVM as well as graph-based methods are difficult to scale. In addition, TSVM, and its recent extensions [19] require that the decision boundary lie in a low density region, yielding sub-optimal accuracy if this is not met. Each of these methods can lead to decreased accuracy when adding unlabelled data. To overcome these limitations, [21] recently proposed a probabilistic formulation for *safe* semi-supervised learning of generative linear models.

In the family of deep probabilistic models, Sum-Product Networks (SPNs) [26] have recently gained popularity, due to their efficiency, i.e. linear-time inference, generality, i.e. they subsume existing approaches such as latent tree models and mixtures, and performance on various tasks including computer vision [26, 10], action recognition [2], speech [24], and language modelling [4].

For probabilistic models, including SPNs, semi-supervised learning with generative models is natural. Data points are assigned to whichever class maximizes $p(\boldsymbol{x}, y) = p(y)p(\boldsymbol{x}|y)$, with $p(\boldsymbol{x}|y)$ being a generative model for the data in class y. Subsequently, the labelled data points can be used to learn the model. Unfortunately, adding unlabelled data can significantly degrade classification accuracy instead of improving it [6].

In this paper, we introduce *safe* semi-supervised parameter learning for SPNs that is *safe*, scalable and non-restrictive. *Safe* means that adding unlabelled data can increase, but not degrade, model performance. The training time scales linearly with added data points and apart from the structure of the underlying SPN, no assumptions are made regarding the data distribution. Unlike other semi-supervised methods, the presented approach does not need low-density or clustering assumptions [3]. In addition to safety, we show competitive results when compared with state-of-the-art approaches in Section 4.

The structure of the paper is as follows: Section 2 introduces the notation used throughout the paper, describes recent approaches for parameter learning in SPNs and introduces the contrastive pessimistic likelihood estimation for *safe* semi-supervised learning of generative models. In Section 3 we propose *safe* semi-supervised learning for SPNs, give derivations for generative and discriminative parameter learning and present the algorithm MCP-SPN for training *safe* semi-supervised SPNs. Experiments are presented in Section 4 showing that *safe* semi-supervised SPNs are able to escape from degenerated supervised solutions, generally outperform purely supervised learning and achieve competitive performance on a variety of data sets. Section 5 concludes the paper and gives future prospects.

2 BACKGROUND

We use capital letters to denote random variables (RVs) and denote a set of RVs as $\boldsymbol{X} = \{X^d\}_{d=1}^D$. Moreover, we denote a realisation of a RV using lower-case letters and indicate a realisation of $\boldsymbol{X}$ using bold lower-case letters, e.g. $\boldsymbol{x} = \{x^d\}_{d=1}^D$. We denote the set of labelled observation as $\mathcal{X} = \{(\boldsymbol{x}_n, \boldsymbol{y}_n)\}_{n=1}^N$ and the set of unlabelled observation as $\mathcal{U} = \{\boldsymbol{u}_m\}_{m=1}^M$ where $\boldsymbol{x}_n$, $\boldsymbol{u}_m$ are the features and $\boldsymbol{y}_n$ the labels in one-hot-encoding. Additionally, we use $\boldsymbol{q} = \{\boldsymbol{q}_m\}_{m=1}^M$ to denote soft labels for the unlabelled observations. We generally write $p(x)$ instead of $p(X = x)$ and write $p(\boldsymbol{x})$ instead of $p(\boldsymbol{X} = \boldsymbol{x})$. For readability, we will refer to the value of an SPN using a calligraphic notation, $\mathcal{S}[\boldsymbol{x}]$, and write $S_i[\boldsymbol{x}]$ for the value of the i_{th} node in an SPN.

2.1 SUM-PRODUCT NETWORKS

SPNs are a deep probabilistic architecture which allows to capture expressive variable interactions, yet guaranteeing exact computations of marginals in linear time. SPNs have its foundation in network polynomials for efficient inference in Bayesian networks introduced by [7]. Poon and Domingos [26] generalized the idea and introduced SPNs over random variables (RVs) with finitely many states.

Definition 1. *(Sum-Product Network [26]) A sum-product network (SPN) over variables $X^1, \ldots, X^d$ is a rooted directed acyclic graph whose leaves are the indicators $x^1, \ldots, x^d$ and $\bar{x}^1, \ldots, \bar{x}^d$ and whose internal nodes are sums and products. Each edge (i, j) emanating from a sum node i has a non-negative weight w_{ij}. The value of a product node is the product of the values of its children. The value of a sum node is $\sum_{j \in Ch(i)} w_{ij} v_j$, where $Ch(i)$ are the children of i and v_j is the value of node j. The value of an SPN $\mathcal{S}[x^1, \bar{x}^1, \ldots, x^d, \bar{x}^d]$ is the value of its root.*

SPNs can be generalized by replacing the leaf node indicators with arbitrary input distributions [25]. Thus, we consider SPNs with arbitrary leaf node distributions throughout the paper.

2.1.1 Generative Learning

The parameters of an SPN can be learned efficiently using Expectation Maximisation (EM) [26, 23]. We use the formulation of [23], where the updates for the parameters of the i_{th} sum node are defined as:

$$n_{ij} = w_{ij} \sum_{n=1}^{N} \frac{1}{\mathcal{S}[\boldsymbol{x}_n]} \frac{\partial \mathcal{S}[\boldsymbol{x}_n]}{\partial S_i} S_j[\boldsymbol{x}_n], \text{ and} \tag{1}$$

$$w_{ij} \leftarrow \frac{n_{ij}}{\sum_{l \in Ch(i)} n_{il}}. \tag{2}$$

Furthermore, the parameter update for an exponential family leaf node with scope d and parameter θ is given by the expected sufficient statistic and can be computed as:

$$g_i(\boldsymbol{x}) = \frac{1}{\mathcal{S}[\boldsymbol{x}]} \frac{\partial \mathcal{S}[\boldsymbol{x}]}{\partial S_i} S_i[\boldsymbol{x}], \text{ and} \tag{3}$$

$$\theta_i \leftarrow \frac{\sum_{n=1}^N g_i(\boldsymbol{x}_n) t(x_n^d)}{\sum_{n=1}^N g_i(\boldsymbol{x}_n)}, \tag{4}$$

where $t(x)$ denotes the sufficient statistics. We assume complete evidence for the RVs $\boldsymbol{X}$ and refer to [23] for a derivation of the updates with partial evidence.

2.1.2 Discriminative Learning

The parameters of a discriminative SPN can be learned by optimising the conditional log likelihood using back-propagation [10]. The set of variables of a discriminative SPN are divided into query variables $\boldsymbol{Y}$, hidden variables $\boldsymbol{H}$ and observed RVs $\boldsymbol{X}$. Therefore, the value of a discriminative SPN is denoted as $\mathcal{S}[\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{H} = \boldsymbol{h} | \boldsymbol{X} = \boldsymbol{x}]$. Furthermore, the conditional probability is estimated by setting all indicator functions of the hidden variables to $\mathbf{1}$ and computing

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{\mathcal{S}[\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{H} = \mathbf{1} | \boldsymbol{X} = \boldsymbol{x}]}{\mathcal{S}[\boldsymbol{Y} = \mathbf{1}, \boldsymbol{H} = \mathbf{1} | \boldsymbol{X} = \boldsymbol{x}]}, \tag{5}$$

where setting the indicators of the hidden variables to one allows the gradients of the conditional log likelihood to be computed in a single upward pass. For the sake of readability, we omit the hidden variables if their indicators are set to one and write $\mathcal{S}[\boldsymbol{y}|\boldsymbol{x}]$ for the value of a discriminative SPN instead.

Given a network structure, one can train a discriminative SPN by gradient ascent using the partial derivatives of the SPN with respect to the parameters of the network. The partial derivatives of the weights take the form

$$\frac{\partial \log p(\boldsymbol{y}|\boldsymbol{x})}{\partial w_{ij}} = \frac{1}{\mathcal{S}[\boldsymbol{y}|\boldsymbol{x}]}\frac{\partial \mathcal{S}[\boldsymbol{y}|\boldsymbol{x}]}{\partial w_{ij}} - \frac{1}{\mathcal{S}[\mathbf{1}|\boldsymbol{x}]}\frac{\partial \mathcal{S}[\mathbf{1}|\boldsymbol{x}]}{\partial w_{ij}}, \tag{6}$$

where $\frac{\partial \mathcal{S}}{\partial w_{ij}} = \frac{\partial \mathcal{S}}{\partial S_i} S_j$ is computed using back-propagation. By setting the gradient of the root node $\frac{\partial \mathcal{S}}{\partial S} = 1$, the gradients of the subsequent nodes are computed in a top-down order. At sum nodes the gradient is propagated to the children using $\frac{\partial \mathcal{S}}{\partial S_j} \leftarrow \frac{\partial \mathcal{S}}{\partial S_j} + w_{ij}\frac{\partial \mathcal{S}}{\partial S_i}$ and at product nodes using $\frac{\partial \mathcal{S}}{S_j} \leftarrow \frac{\partial \mathcal{S}}{S_j} + \frac{\partial \mathcal{S}}{\partial S_i}\prod_{l \in Ch(i)\setminus\{j\}} S_l$. As indicated, the gradient at a node j is accumulated based on the parents gradients. We refer to [10] for further details on the derivation of the gradients and derivations of *hard* gradient updates.

As in network polynomials for Bayesian networks [7], partial derivatives of any parameter in an SPN can be calculated using the chain rule, leading to a straight forward computation of parameter updates for the leaf node distributions, i.e.

$$\frac{\partial \mathcal{S}[\boldsymbol{y}|\boldsymbol{x}]}{\partial \theta} = \frac{\partial \mathcal{S}[\boldsymbol{y}|\boldsymbol{x}]}{\partial S_i}\frac{\partial p(x^d|\theta)}{\partial \theta}, \text{ and} \tag{7}$$

$$\frac{\partial \log p(\boldsymbol{y}|\boldsymbol{x})}{\partial \theta} = \frac{1}{\mathcal{S}[\boldsymbol{y}|\boldsymbol{x}]}\frac{\partial \mathcal{S}[\boldsymbol{y}|\boldsymbol{x}]}{\partial \theta} - \frac{1}{\mathcal{S}[\mathbf{1}|\boldsymbol{x}]}\frac{\partial \mathcal{S}[\mathbf{1}|\boldsymbol{x}]}{\partial \theta}. \tag{8}$$

In the case of univariate Gaussian distributions, the updates are computed by taking the partial derivatives of the mean and the variance of the distribution.

2.2 CONTRASTIVE PESSIMISTIC LIKELIHOOD ESTIMATION

Most semi-supervised learning approaches require strong assumptions, e.g. low density assumption for TSVM, and can lead to decreased performance with increasing number of unlabelled data samples if these assumptions are violated. Loog [21] has proposed Contrastive Pessimistic Likelihood Estimation (CPLE) in order to facilitate performance guarantees while only relying on the assumptions of an underlying generative model.

CPLE maintains soft labels (hypotheses) for each unlabelled data point, and assigns them pessimistically, using a training objective that maximizes the log likelihood on the data $L(\theta|\cdot)$ but minimizes the improvement provided by the unlabelled data. Therefore, CPLE yields in a *safe* semi-supervised objective.

Model parameters under CPLE are estimated according to:

$$\theta^* = \operatorname*{argmax}_{\theta \in \Theta} \arg \min_{\boldsymbol{q} \in \Delta_{K-1}^{M}} L(\theta|\mathcal{X}, \mathcal{U}, \boldsymbol{q}) - L(\theta^+|\mathcal{X}, \mathcal{U}, \boldsymbol{q}), \tag{9}$$

where $\boldsymbol{q}$ denotes soft labels for every unlabelled data point and θ^+ denotes the parameters of a purely supervised model derived only on $\mathcal{X}$. The introduction of soft labels, respects the fact that classes may be overlapping. In the case of K unique class labels each soft label vector $\boldsymbol{q}_m$ is an element of the $K-1$ simplex Δ_{K-1}.

Since the trained classifier assumes the worst-case improvement, its performance cannot degrade when adding unlabelled data. Loog [21] constrains the CPLE to generative models, and provides a concrete solution for a simple linear classifier based on linear discriminative analysis. In the following, we define a contrastive pessimistic objective for generative and discriminative SPNs, yielding in a *safe* semi-supervised learning procedure with linear computational complexity which only relies on the assumptions intrinsic to the given network structure.

3 SAFE SEMI-SUPERVISED SPNS

Given an SPN $\mathcal{S}[\boldsymbol{x}, \boldsymbol{y}]$ we can find the optimal parameters for generative *safe* semi-supervised learning using the CPLE objective defined in Equation (9). For clarity, we always use the plus operator to indicate parameters of the purely supervised solution, e.g. weights w^+, and indicate parameters of the *safe* semi-supervised solution using an asterisk. Due to the conservative choice of $\boldsymbol{q}$ by minimizing the improvement over the supervised result, and since we can always take $\theta^* = \theta^+$ in the worst case, this objective is guaranteed to lead to a *safe* solution. More formally, as shown in Loog [21], it is guaranteed that

$$L(\theta^*|\mathcal{X}, \mathcal{U}, \boldsymbol{q}) \geqslant L(\theta^+|\mathcal{X}, \mathcal{U}, \boldsymbol{q}). \tag{10}$$

Therefore, if log likelihoods are used in the CPLE objective the *safe* semi-supervised solution has at least the same log likelihood given $\mathcal{X}, \mathcal{U}$ and $\boldsymbol{q}$ as the purely supervised objective.

3.1 GENERATIVE SAFE SEMI-SUPERVISED LEARNING

In the following we derive the Expectation Maximisation (EM) updates for the generative *safe* semi-supervised

SPN. Therefore, let

$$\mathcal{S}[\boldsymbol{x},\boldsymbol{y}|\theta] = \sum_{k=1}^{K} \mathbb{1}_{y_k} w_k S_k[\boldsymbol{x}|\boldsymbol{y},\theta] \tag{11}$$

be the likelihood of a semi-supervised SPN for labelled observations $(\boldsymbol{x},\boldsymbol{y}) \in \mathcal{X}$. We denote $\mathbb{1}_{y_k}$ to be the indicator for class k which is one if y_k is true and zero otherwise. Furthermore, let

$$\mathcal{S}[\boldsymbol{u},\boldsymbol{q}|\theta] = \sum_{k=1}^{K} q_k w_k S_k[\boldsymbol{u}|\boldsymbol{q},\theta] \tag{12}$$

be the likelihood of a semi-supervised SPN for unlabelled observations $(\boldsymbol{u},\boldsymbol{q})$ with $\boldsymbol{q}$ being the soft labels of the data. Note that $\sum_k q_k = 1$ for all unlabelled observations, as each soft label vector is an element of the $K-1$ simplex. We can therefore define the generative log likelihood function of a semi-supervised SPN as the sum of the log likelihood given the labelled data and the unlabelled data. Formally, we define the generative log likelihood of a semi-supervised SPN as

$$\begin{aligned} L(\theta|\mathcal{X},\mathcal{U},\boldsymbol{q}) = &\sum_{n=1}^{N} \log \mathcal{S}[\boldsymbol{x}_n,\boldsymbol{y}_n|\theta] \\ &+ \sum_{m=1}^{M} \log \mathcal{S}[\boldsymbol{u}_m,\boldsymbol{q}_m|\theta] \end{aligned} \tag{13}$$

which allows for straightforward derivation of the EM updates. The updates of the weights of sum node S_i in $\mathcal{S}$ can be computed as in Eq. (2) using the following n_{ij}, i.e.

$$\begin{aligned} n_{ij} = w_{ij} &\sum_{n=1}^{N} \frac{1}{\mathcal{S}[\boldsymbol{x}_n,\boldsymbol{y}_n]} \frac{\partial \mathcal{S}[\boldsymbol{x}_n,\boldsymbol{y}_n]}{\partial S_i} S_j[\boldsymbol{x}_n|\boldsymbol{y}_n] \\ +w_{ij} &\sum_{m=1}^{M} \frac{1}{\mathcal{S}[\boldsymbol{u}_m,\boldsymbol{q}_m]} \frac{\partial \mathcal{S}[\boldsymbol{u}_m,\boldsymbol{q}_m]}{\partial S_i} S_j[\boldsymbol{u}_m|\boldsymbol{q}_m], \end{aligned} \tag{14}$$

where we omitted the parametrization of the network for better readability. Furthermore, we can update the parameters of an exponential family leaf node with scope d using the expected sufficient statistics as

$$g_i(\boldsymbol{x}_n,\boldsymbol{y}_n) = \frac{1}{\mathcal{S}[\boldsymbol{x}_n,\boldsymbol{y}_n]} \frac{\partial \mathcal{S}[\boldsymbol{x}_n,\boldsymbol{y}_n]}{\partial S_i} S_i[\boldsymbol{x}_n|\boldsymbol{y}_n], \tag{15}$$

$$g_i(\boldsymbol{u}_m,\boldsymbol{q}_m) = \frac{1}{\mathcal{S}[\boldsymbol{u}_m,\boldsymbol{q}_m]} \frac{\partial \mathcal{S}[\boldsymbol{u}_m,\boldsymbol{q}_m]}{\partial S_i} S_i[\boldsymbol{u}_m|\boldsymbol{q}_m], \tag{16}$$

$$\theta_i \leftarrow \frac{\sum_{n=1}^{N} g_i(\boldsymbol{x}_n,\boldsymbol{y}_n) t(x_n^d) + \sum_{m=1}^{M} g_i(\boldsymbol{u}_m,\boldsymbol{q}_m) t(u_m^d)}{\sum_{n=1}^{N} g_i(\boldsymbol{x}_n,\boldsymbol{y}_n) + \sum_{m=1}^{M} g_i(\boldsymbol{u}_m,\boldsymbol{q}_m)}, \tag{17}$$

where we assume complete evidence for the RVs $\boldsymbol{X}$ and $\boldsymbol{U}$.

Subsequently, the soft label for class k of an unlabelled sample m is updated pessimistically with gradient descent using the partial derivative of q_{mk} which is defined as

$$\frac{\partial L(\theta|\mathcal{X},\mathcal{U},\boldsymbol{q})}{\partial q_{mk}} = \frac{w_k S_k[\boldsymbol{u}_m|\boldsymbol{q}_m,\theta]}{\mathcal{S}[\boldsymbol{u}_m,\boldsymbol{q}_m|\theta]}, \text{ and} \tag{18}$$

$$\nabla q_{mk} = \frac{\partial L(\theta^*|\mathcal{X},\mathcal{U},\boldsymbol{q})}{\partial q_{mk}} - \frac{\partial L(\theta^+|\mathcal{X},\mathcal{U},\boldsymbol{q})}{\partial q_{mk}}. \tag{19}$$

Note that after each gradient update it is necessary to ensure that the soft labels for the unlabelled data points are on the $K-1$ simplex. For this purpose, the soft labels are projected back to the $K-1$ simplex using the approach by Duchi et al. [8].

3.2 DISCRIMINATIVE SAFE SEMI-SUPERVISED LEARNING

Conditional likelihoods instead of generative objectives are a more natural way of learning SPNs for classification tasks in the semi-supervised regime. Formally, the model parameters for discriminative *safe* semi-supervised SPNs are estimated according to

$$\operatorname*{argmax}_{\theta\in\Theta} \arg \min_{\boldsymbol{q}\in\Delta_{K-1}^{M}} CL(\theta|\mathcal{X},\mathcal{U},\boldsymbol{q}) - CL(\theta^+|\mathcal{X},\mathcal{U},\boldsymbol{q}), \tag{20}$$

where we intentionally use $CL(\theta|\cdot)$ to indicate the use of the conditional log likelihood. Extending the formulation for discriminative SPNs allows to define a discriminative learning approach for *safe* semi-supervised SPNs, i.e.,

$$\begin{aligned} CL(\theta|\mathcal{X},\mathcal{U},\boldsymbol{q}) = &\sum_{n=1}^{N} \log \mathcal{S}[\boldsymbol{y}_n|\boldsymbol{x}_n,\theta] \\ &+ \sum_{m=1}^{M} \log \mathcal{S}[\boldsymbol{q}_m|\boldsymbol{u}_m,\theta], \end{aligned} \tag{21}$$

where the conditional likelihood for labelled and unlabelled data, respectively, are given as

$$\mathcal{S}[\boldsymbol{y}|\boldsymbol{x},\theta] = \frac{S_k[\boldsymbol{x},\boldsymbol{y}|\theta]}{S_k[\boldsymbol{x},\boldsymbol{1}|\theta]}, \text{ and} \tag{22}$$

$$\mathcal{S}[\boldsymbol{q}|\boldsymbol{u},\theta] = \frac{S_k[\boldsymbol{u},\boldsymbol{q}|\theta]}{S_k[\boldsymbol{u},\boldsymbol{1}|\theta]}. \tag{23}$$

The partial derivatives for the weights of the discriminative semi-supervised SPN therefore become

$$\begin{aligned}\frac{\partial}{\partial w_{ij}} CL(\theta|\mathcal{X},\mathcal{U},\boldsymbol{q}) =&\\ \sum_{n=1}^{N} \frac{1}{\mathcal{S}[\boldsymbol{y}_n|\boldsymbol{x}_n]}\frac{\partial \mathcal{S}[\boldsymbol{y}_n|\boldsymbol{x}_n]}{\partial w_{ij}} &- \frac{1}{\mathcal{S}[\mathbf{1}|\boldsymbol{x}_n]}\frac{\partial \mathcal{S}[\mathbf{1}|\boldsymbol{x}_n]}{\partial w_{ij}}\\ + \sum_{m=1}^{M} \frac{1}{\mathcal{S}[\boldsymbol{q}_m|\boldsymbol{u}_m]}\frac{\partial \mathcal{S}[\boldsymbol{q}_m|\boldsymbol{u}_m]}{\partial w_{ij}} &- \frac{1}{\mathcal{S}[\mathbf{1}|\boldsymbol{u}_m]}\frac{\partial \mathcal{S}[\mathbf{1}|\boldsymbol{u}_m]}{\partial w_{ij}}.\end{aligned} \tag{24}$$

Similarly, we can derive the partial derivatives of the leaf node parameters by applying the chain rule, leading to the following parameter updates

$$\frac{\partial}{\partial \theta} CL(\theta|\mathcal{X},\mathcal{U},\boldsymbol{q}) = \tag{25}$$

$$\sum_{n=1}^{N} \frac{1}{\mathcal{S}[\boldsymbol{y}_n|\boldsymbol{x}_n]}\frac{\partial \mathcal{S}[\boldsymbol{y}_n|\boldsymbol{x}_n]}{\partial \theta} - \frac{1}{\mathcal{S}[\mathbf{1}|\boldsymbol{x}_n]}\frac{\partial \mathcal{S}[\mathbf{1}|\boldsymbol{x}_n]}{\partial \theta} \tag{26}$$

$$+ \sum_{m=1}^{M} \frac{1}{\mathcal{S}[\boldsymbol{q}_m|\boldsymbol{u}_m]}\frac{\partial \mathcal{S}[\boldsymbol{q}_m|\boldsymbol{u}_m]}{\partial \theta} - \frac{1}{\mathcal{S}[\mathbf{1}|\boldsymbol{u}_m]}\frac{\partial \mathcal{S}[\mathbf{1}|\boldsymbol{u}_m]}{\partial \theta}. \tag{27}$$

To pessimistically update the soft labels, one can use gradient descent on the partial derivatives similar as for the generative objective in Eq. (19).

3.3 ALGORITHM

The algorithm Maximum Contrastive Pessimistic SPN (MCP-SPN) for learning *safe* semi-supervised SPNs is illustrated in Algorithm 1 and consists of the following adversarial steps: (1) optimising the *safe* semi-supervised solution on the given soft labels by maximising a generative or discriminative objective (2) minimising the improvement of the semi-supervised solution over the purely supervised solution by adjusting the soft labels pessimistically. As an SPN is a multi-linear function in terms of the model parameters we can apply the generalisation of the minmax theorem for multi-linear functions [16] and interchange the maximisation and the minimisation in our algorithm.

Algorithm 1: MCP-SPN

```
Input: A valid SPN structure $\mathcal{S}$, labelled data $\mathcal{X}$,
       unlabelled data $\mathcal{U}$.
Output: Learned parameters and soft labels.
// learn purely supervised SPN
if generative then
    $\theta^+ \leftarrow \mathrm{argmax}_{\theta\in\Theta} \log \mathcal{S}[\boldsymbol{x},\boldsymbol{y}|\theta]$
else
    $\theta^+ \leftarrow \mathrm{argmax}_{\theta\in\Theta} \log \mathcal{S}[\boldsymbol{y}|\boldsymbol{x},\theta]$
end
// initialize soft labels
if optimistic then
    foreach $k \in \{1,\ldots,K\}$ do
        $\boldsymbol{q}_k \leftarrow \frac{S_k[\boldsymbol{u}|\theta^+]}{\mathcal{S}[\mathbf{1}|\boldsymbol{u},\theta^+]}$
    end
else
    $\boldsymbol{q} \sim \mathrm{Dir}(\frac{1}{K},\ldots,\frac{1}{K})$
end
// learn safe semi-supervised SPN
repeat
    // optimistic parameter learning
    if generative then
        // Eq. 14 and Eq. 17
        $\theta^* \leftarrow \mathrm{argmax}_{\theta\in\Theta} \log \mathcal{S}[\boldsymbol{x},\boldsymbol{y}|\theta] + \log \mathcal{S}[\boldsymbol{u},\boldsymbol{q}|\theta]$
    else
        // Eq. 24 and Eq. 27
        $\theta^* \leftarrow \mathrm{argmax}_{\theta\in\Theta} \log \mathcal{S}[\boldsymbol{y}|\boldsymbol{x},\theta] + \log \mathcal{S}[\boldsymbol{q}|\boldsymbol{u},\theta]$
    end
    // pessimistic soft label adjustment
    $\boldsymbol{q} \leftarrow \boldsymbol{q} - \alpha\nabla\boldsymbol{q}$                    // Eq. 19
    $\boldsymbol{q} \leftarrow$ projectOnSimplex($\boldsymbol{q}$, $\Delta^M_{K-1}$)
until convergence or early stopping
return $\theta^*$ and $\boldsymbol{q}$
```

Depending on the choice of the objective, the MCP-SPN procedure first finds a purely supervised solution by only maximising the chosen objective with respect to the labelled data. Secondly, we initialise all soft labels of the unlabelled data either using an optimistic approach or using random draws from a Dirichlet distribution. In the case of a generative objective the purely supervised solution can degenerate to a point mass estimator. It is therefore useful for generative SPNs to initialise the soft labels using random draws instead of starting from an optimistic labelling. After initialising all soft labels the MCP-SPN procedure finds a *safe* semi-supervised solution θ^* by alternating between the two adversarial steps. The function call *projectOnSimplex* refers to the approach in [8], which we use to project the soft label assignments back to the $K-1$ simplex (but other approaches for this task could also be used). Note that we found it useful to decrease the learning rate α of the pessimistic soft labels adjustment over time. In our experiments we therefore used a simple decay function $\alpha \leftarrow \alpha/\sqrt(iteration)$, if necessary more advanced approaches can be used instead. The source code for *safe* semi-supervised learning of SPNs is available on-line[1].

[1] `https://github.com/trappmartin/SSLSPN_UAI2017`

4 EXPERIMENTS

We analysed the performance of the *safe* semi-supervised learning approach qualitatively on synthetic data using the generative objective, and quantitatively on various data sets using both objectives.

4.1 Datasets and Model Generation

In addition to the synthetic two moons data set [15], we used various well known data sets from the UCI repository [20] to evaluate the performance of the *safe* semi-supervised parameter learning approaches. We preprocessed the data in the following way: (1) we removed features with zero variance, (2) we applied z-score normalisation. To ensure broad applicability of the approaches, we selected data sets which origin from a variety of domains and cover a wide range of number of samples and dimensions. Details on the selected data sets are shown in Table 1 where the last column lists the number of labelled samples used in all experiments. Note that the number of labelled samples per data set is calculated as in [21].

To consistently learn SPN structures for all experiments we extended the well-known learnSPN [11] algorithm for Gaussian distributed data, similar as in [29]. Additionally, we added a layer that conditions on the class labels resulting in structures that are suitable for supervised and semi-supervised learning [10]. As learnSPN produces large SPN structures, which might lead to over-fitting, we used a two step procedure for regularizing the resulting network. First, we estimate and apply a pruning depth of the network and secondly, we remove degenerated leaf distributions. We further ensured throughout all regularization steps that the resulting SPN is complete and decomposable.

4.2 QUALITATIVE RESULTS ON SYNTHETIC DATA

Due to the non-linearity, flexibility and complexity of SPNs with arbitrary leaf distributions, learning a *safe* semi-supervised objective for such networks, without enforcing prior assumptions on the data distribution, is much more difficult than for linear models such as Linear Discriminant Analysis (LDA) [21]. Therefore, we analysed the behaviour of *safe* semi-supervised SPNs qualitatively on the synthetic two moons data set [15]. Figure 1(a) shows the purely supervised solution for a small subset of labelled observations and the solution found using a generative *safe* semi-supervised SPN over time. For reference the oracle solution, which knows the labels of all observations, is depicted in Figure 1(b).

The purely supervised SPN clearly over-fits the few labelled examples and degenerated almost completely to a kernel density estimator. The *safe* semi-supervised parameter learning approach is initialised using soft labels drawn from a Dirichlet distribution, to allow the model to escape from the local optimum. As shown in Figure 1(c), the generative *safe* semi-supervised approach is able to find a reasonable solution after only three iterations even with a random initialisation of the soft labels. The model converges after only 20 iterations to a stable solution without enforcing restrictive assumptions on the data distribution.

Data Set	N	D	K	$2 \cdot D + K$
BUPA	345	6	2	14
Fertility	100	9	2	20
Haberman	306	3	2	8
ILPD	583	10	2	22
Ionosphere	351	34	2	70
Iris	150	4	3	11
Parkinsons	197	23	2	48
WDBC	569	32	2	66
Wine	178	13	3	29

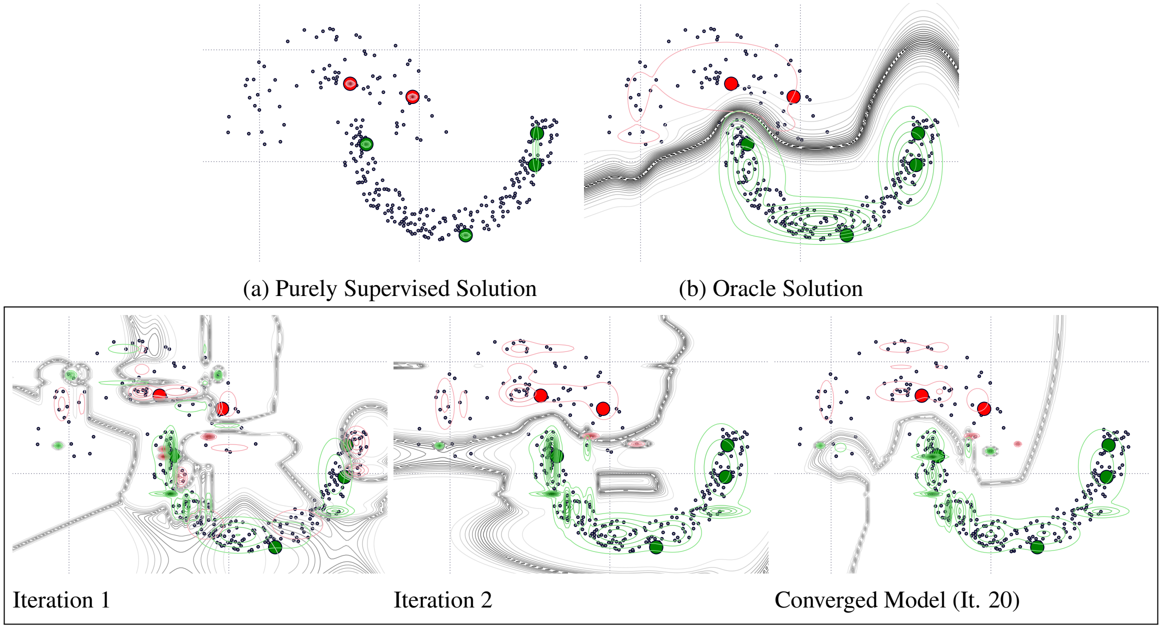
Table 1: Datasets, details on the number of samples (N), dimensionality (D), number of classes (K) and number of labelled samples used in all evaluations ($2 \cdot D + K$). The number of labelled samples is obtained according to [21].

4.3 GENERATIVE SPN PERFORMANCE FOR SAFE SEMI-SUPERVISED LEARNING

Experimental Setup We constructed truncated network structures using learnSPN [11]. The truncation levels have been estimated using the Akaike information criterion [1]. After the structure construction we initialised all soft labels using random draws from a Dirichlet distribution with equal concentration parameter for all classes.

Furthermore, we lower bounded the variance of the leaf distributions to the ith percentile of the nearest neighbour distances of all data points in $\mathcal{X} \cup \mathcal{U}$. We selected the smallest percentile such that the constructed lower bound is above zero. Imposing a lower bound on the variances of the leaf node distributions in such way prevents the univariate Gaussian distributions from degeneration with minimal influence on the model expressiveness.

We analysed our approach for generative semi-supervised learning of SPNs by: (1) splitting each dataset into training (80%) and testing set (20%), (2) draw $2 \cdot D + K$ labelled samples stratified from each training set as proposed in [21]. We used an additional labelled validation set of $2 \cdot D + K$ samples for early stopping. In addition to



(a) Purely Supervised Solution

(b) Oracle Solution

(c) Safe Semi-Supervised Solution

Figure 1: Qualitative results on the two moons data set. The colour of the dots indicate their class label and estimated density regions of the classes are shown using coloured density plots. Unlabelled samples are shown as small black dots. The decision boundary of the model is shown using a density plot coloured in grey. (a) Purely supervised solution over-fits the few training examples and degenerates to a point density estimator. (b) Oracle solution, (c) generative *safe* semi-supervised parameter learning is able to find a reasonable solution after only a few iterations without making restrictive assumptions on the data distribution.

the labelled samples, we used all remaining observations in the training set as unlabelled examples.

Results We compare the performance of the *safe* semi-supervised learning (SSL) approach against the purely supervised solution, an oracle solution and the solution found by the recently introduced inductive approach (MCPLDA) [21]. All models where evaluated on the test set. The resulting average log likelihood values are estimated over 100 independent runs. Table 2 lists the average log likelihood and the standard errors of all approaches. Note that the guarantee of the CPLE is on the training set including unlabelled observations. We expect however, the performance of the SSL approach on the test set in expectation to be better or similar to the purely supervised learner.

In most cases we could indeed find an improvement of the *safe* semi-supervised approach over the purely supervised solution. In the cases of *Parkinsons*, *WDBC* and *Wine* the purely supervised learner already finds solutions which are close to the oracle solution. This might be due to the relative simple geometric properties of those data sets. In this situation, our SSL approach converged to solutions which are close to the purely supervised solution. In some cases, e.g. *BUPA*, *Fertility*, *Haberman* and *ILPD*, we could find an improvement upon the oracle solution or near oracle solution performance. Furthermore, *safe* semi-supervised SPNs generally outperform MCPLDA on almost all data sets in terms of the log likelihood, with one exception being the *Iris* data set. Moreover, our approach generally reaches very stable results and achieves estimated standard errors lower than those of the supervised and the MCPLDA solution.

4.4 DISCRIMINATIVE SPN PERFORMANCE FOR SAFE SEMI-SUPERVISED LEARNING

We assess the classification performance of discriminative *safe* semi-supervised learning below, as optimising a discriminative objective is a more natural way for classification tasks.

Experimental Setup Similar to the quantitative evaluation of the generative approach, we constructed truncated structures for all experiments. To avoid over-fitting we used early truncation of the model, estimated according to the performance on the validation set. We further initialised all soft labels using optimistic predictions from the purely supervised model. To obtain training and test

Data Set	**Supervised**	**SSL**	**Oracle**	**MCPLDA**
BUPA	$-438.75 \pm 7 \cdot 10^{0}$	$\mathbf{-7.31} \pm 6 \cdot 10^{-2}$	$-8.80 \pm 2 \cdot 10^{-1}$	$-9.07 \pm 3 \cdot 10^{-2}$
Fertility	$-3.31 \pm 3 \cdot 10^{-2}$	$\mathbf{-3.06} \pm 7 \cdot 10^{-3}$	$-3.00 \pm 6 \cdot 10^{-3}$	$-12.68 \pm 5 \cdot 10^{-2}$
Haberman	$-138.63 \pm 4 \cdot 10^{0}$	$\mathbf{-5.05} \pm 6 \cdot 10^{-2}$	$-5.14 \pm 6 \cdot 10^{-2}$	$-7.83 \pm 1 \cdot 10^{-1}$
ILPD	$-5.62 \pm 3 \cdot 10^{0}$	$\mathbf{-1.15} \pm 2 \cdot 10^{-2}$	$-1.00 \pm 1 \cdot 10^{-2}$	$-37.54 \pm 1 \cdot 10^{-1}$
Ionosphere	$-2.83 \pm 5 \cdot 10^{-2}$	$\mathbf{-1.61} \pm 1 \cdot 10^{-2}$	$-1.52 \pm 9 \cdot 10^{-3}$	$-46.12 \pm 5 \cdot 10^{-2}$
Iris	$-20.65 \pm 9 \cdot 10^{-1}$	$-3.78 \pm 3 \cdot 10^{-2}$	$-2.17 \pm 1 \cdot 10^{-2}$	$\mathbf{-2.65} \pm 5 \cdot 10^{-2}$
Parkinsons	$\mathbf{-1.32} \pm 4 \cdot 10^{-3}$	$-1.34 \pm 4 \cdot 10^{-3}$	$-1.30 \pm 2 \cdot 10^{-3}$	$-2.27 \pm 5 \cdot 10^{-2}$
WDBC	$\mathbf{-1.90} \pm 1 \cdot 10^{-3}$	$-1.93 \pm 2 \cdot 10^{-3}$	$-1.88 \pm 3 \cdot 10^{-4}$	$-10.75 \pm 1 \cdot 10^{-2}$
Wine	$\mathbf{-2.47} \pm 4 \cdot 10^{-3}$	$\mathbf{-2.47} \pm 2 \cdot 10^{-3}$	$-2.44 \pm 9 \cdot 10^{-4}$	$-15.28 \pm 2 \cdot 10^{-2}$

Table 2: Averaged log likelihood and standard errors estimated on the test set over 100 independent trials. The best results for each data set obtained by a supervised or semi-supervised model are shown in **bold** face.

sets, we followed the same approach as described for the generative experiments. Similar to the generative evaluation, the randomly drawn labelled subset is obtained from the training set and the performance of each algorithm is estimated over 100 independent trials.

Results We compared the performance of our discriminative approach against the purely supervised solution, the oracle solution and the following state of the art approaches: Transductive SVM (TSVM) [5], Minimum Entropy Regularization (MER) [13] and the recently published Implicitly Constrained Least Squares (ICLS) [18]. To assess the performance of a classification method, we computed the F_1 score for binary classification tasks. In cases of multi-class data sets, we used the macro average F_1 score. To compute multi-class predictions for approaches designed only for binary classification we used the one-vs-rest approach. The average F_1 scores as well as the standard errors of all approaches are shown in Table 3.

The *safe* semi-supervised parameter learning approach achieves competitive results for almost all data sets. In general, our approach produces reasonable results and does not degenerate if certain assumptions are not met. Moreover, in several cases our discriminative approach achieves test F_1 scores which are comparable to those of the oracle solution, e.g. for *Haberman* and *Wine*. We could find the lowest performance of our approach on the *Fertility* data set. Note that the F_1 scores on *Fertility*, *Haberman* and *ILPD* are generally very low as those are imbalanced or skewed data sets.

In general, the proposed *safe* semi-supervised learning for SPNs is a powerful adversarial approach which scales linearly in the number of samples and is non-restrictive. Even though we achieved competitive results even on data sets where low density assumptions are met, e.g. *Wine*, further improvements may be achieved by trading off optimism and pessimism. One way of approaching this issue would be to add a weighting scheme into the CPLE formulation.

Even though optimising the conditional log likelihood inside the CPLE objective provides a reasonable criterion for classification tasks, this approach does not guarantee to improve the classification performance of the learner. It is therefore possible, that better classification performance can be achieved by using a multi-class squared-hinge loss, which was recently used in a related model [12].

5 CONCLUSION AND FUTURE WORK

In this paper, we introduced the first approach for semi-supervised parameter learning with Sum-Product Networks (SPNs). We presented generative and discriminative *safe* semi-supervised learning procedures which guarantee that adding unlabelled data can increase, but not degrade, the performance of the learner on the training set. Furthermore, our approach exploits the tractability of SPNs and scales linear in the number of data points and model parameters. In contrast to other semi-supervised learners, the proposed approach is non-restrictive and does not need prior assumptions on the data distribution. The approach allows broad applicability and is a generic *safe* semi-supervised learning procedure for all models which leverage the sum-product theorem [9] and therefore provides a semi-supervised learning procedure beyond SPNs.

We investigated the performance of our approach quantitatively and qualitatively. In the conducted qualitative analysis we found that the generative *safe* semi-supervised parameter learning approach is able to a find reasonable solution after only a few iterations and is able to escape from the degenerated supervised solutions. We further compared the performance of *safe* semi-supervised parameter learning for SPNs against state-of-the-art approaches.

Data Set	Supervised	SSL	Oracle	TSVM	ICLSC	MER
BUPA	$0.41\pm1\cdot10^{-2}$	$0.40\pm1\cdot10^{-2}$	$0.48\pm5\cdot10^{-3}$	$0.36\pm2\cdot10^{-2}$	$\mathbf{0.47}\pm7\cdot10^{-3}$	$0.42\pm1\cdot10^{-2}$
Fertility	$0.07\pm2\cdot10^{-2}$	$0.03\pm1\cdot10^{-2}$	$0.06\pm2\cdot10^{-2}$	$0.07\pm2\cdot10^{-2}$	$0.07\pm2\cdot10^{-2}$	$\mathbf{0.12}\pm2\cdot10^{-2}$
Haber.	$0.23\pm2\cdot10^{-2}$	$0.28\pm2\cdot10^{-2}$	$0.25\pm0.$	$0.20\pm2\cdot10^{-2}$	$\mathbf{0.33}\pm1\cdot10^{-2}$	$0.27\pm1\cdot10^{-2}$
ILPD	$0.17\pm2\cdot10^{-2}$	$0.20\pm1\cdot10^{-2}$	$0.24\pm4\cdot10^{-3}$	$0.23\pm2\cdot10^{-2}$	$0.29\pm1\cdot10^{-2}$	$\mathbf{0.33}\pm2\cdot10^{-2}$
Ionos.	$0.79\pm4\cdot10^{-3}$	$\mathbf{0.82}\pm4\cdot10^{-3}$	$0.87\pm0.$	$0.66\pm9\cdot10^{-3}$	$0.61\pm9\cdot10^{-3}$	$0.70\pm7\cdot10^{-3}$
Iris	$0.73\pm1\cdot10^{-2}$	$\mathbf{0.88}\pm1\cdot10^{-2}$	$0.93\pm0.$	$0.72\pm1\cdot10^{-2}$	$0.74\pm2\cdot10^{-2}$	$0.80\pm6\cdot10^{-3}$
Parkins.	$0.72\pm1\cdot10^{-2}$	$\mathbf{0.77}\pm4\cdot10^{-3}$	$0.82\pm4\cdot10^{-3}$	$0.74\pm1\cdot10^{-2}$	$0.66\pm2\cdot10^{-2}$	$0.68\pm1\cdot10^{-2}$
PID	$0.38\pm1\cdot10^{-2}$	$0.45\pm1\cdot10^{-2}$	$0.64\pm8\cdot10^{-4}$	$0.45\pm1\cdot10^{-2}$	$0.54\pm7\cdot10^{-3}$	$\mathbf{0.57}\pm9\cdot10^{-3}$
WDBC	$0.85\pm3\cdot10^{-3}$	$0.90\pm2\cdot10^{-3}$	$0.92\pm3\cdot10^{-4}$	$0.91\pm4\cdot10^{-3}$	$0.88\pm4\cdot10^{-3}$	$\mathbf{0.92}\pm3\cdot10^{-3}$
Wine	$0.82\pm7\cdot10^{-3}$	$\mathbf{0.97}\pm2\cdot10^{-3}$	$0.97\pm4\cdot10^{-3}$	$0.96\pm2\cdot10^{-3}$	$0.95\pm7\cdot10^{-3}$	$0.95\pm9\cdot10^{-3}$

Table 3: Macro-average F1 scores estimated on the test set over 100 independent trials. The best results for each data set obtained by a supervised or semi-supervised model are shown in **bold** face.

The proposed *safe* semi-supervised learning for SPNs achieves competitive performance compared to state-of-the-art approaches, and outperformed supervised SPNs in the majority of cases. Even though our approach is non-restrictive and does not need prior assumptions on the data distribution, *safe* semi-supervised SPNs can utilise low density regions if the structure of the network reflects geometric properties of the data distribution. However, as such assumptions are not enforced in the learning procedure, our *safe* semi-supervised learner is still capable of finding decision boundaries which cross high density regions.

Future research directions include: interleaving network structure learning with semi-supervised parameter learning, extensions to other learning objectives, investigating possibilities for trading off optimism and pessimism in the objective, dealing with covariate shift and analysing instability in *safe* semi-supervised SPNs and its comparison with GANs. Furthermore, we plan to apply our *safe* semi-supervised learning approach to high-dimensional classification problems from medicine, genetics and other domains.

Acknowledgments

This research is partially funded by the Austrian Science Fund (FWF): P 27530 and P 27803-N15.

References

[1] H. Akaike. A new look at the statistical model identification. *Transactions on Automatic Control*, 19(6):716–723, 1974.

[2] M.R. Amer and S. Todorovic. Sum-product networks for modeling activities with stochastic structure. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1314–1321, 2012.

[3] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. Adaptive computation and machine learning. MIT Press, 2006.

[4] W.C. Cheng, S. Kok, H.V. Pham, H.L. Chieu, and K.M.A. Chai. Language modeling with sum-product networks. In *Proceedings of Annual Conference of the International Speech Communication Association (Interspeech)*, pages 2098–2102, 2014.

[5] R. Collobert, F. Sinz, J. Weston, and L. Bottou. Large scale transductive svms. *Journal of Machine Learning Research*, 7:1687–1712, 2006.

[6] F. G. Cozman, I. Cohen, and M. Cirelo. Unlabeled data can degrade classification performance of generative classifiers. In *Proceedings of International Florida Artificial Intelligence Society Conference*, pages 327–331, 2002.

[7] A. Darwiche. *Modeling and reasoning with Bayesian networks*. Cambridge University Press, 2009.

[8] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the l 1-ball for learning in high dimensions. In *International Conference on Machine Learning (ICML)*, pages 272–279, 2008.

[9] A.L. Friesen and P. Domingos. The sum-product theorem: A foundation for learning tractable models. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 1909–1918, 2016.

[10] R. Gens and P. Domingos. Discriminative learning of sum-product networks. In *Proceedings of Ad-*

vances in Neural Information Processing Systems (NIPS), pages 3248–3256, 2012.

[11] R. Gens and P. Domingos. Learning the structure of sum-product networks. *International Conference on Machine Learning (ICML)*, pages 873–880, 2013.

[12] R. Gens and P. Domingos. Compositional kernel machines. In *Proceedings of International Conference on Learning Representations*, 2017.

[13] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 529–536, 2004.

[14] M.F.A. Hady and F. Schwenker. Semi-supervised learning. In *Handbook on Neural Information Processing*, pages 215–239. Springer, 2013.

[15] A. K. Jain and M. Law. Data clustering: A users dilemma. In *International Conference on Pattern Recognition and Machine Intelligence (PReMI)*, pages 1–10, 2005.

[16] B. Kalantari. Approximating nash equilibrium via multilinear minimax. *arXiv preprint*, 2016.

[17] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 3581–3589, 2014.

[18] J. H. Krijthe and M. Loog. Implicitly constrained semi-supervised least squares classification. In *Proceedings of International Symposium on Intelligent Data Analysis*, pages 158–169, 2015.

[19] Y.F. Li and Z.H. Zhou. Towards making unlabeled data never hurt. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 37(1):175–188, 2015.

[20] M. Lichman. UCI machine learning repository, 2013.

[21] M. Loog. Contrastive pessimistic likelihood estimation for semi-supervised classification. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(3):462–475, 2016.

[22] L. Maaloe, C. K. Sonderby, S. K. Sonderby, and O. Winther. Auxiliary deep generative models. In *Proceedings of The International Conference on Machine Learning (ICML)*, pages 1445–1453, 2016.

[23] 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 (TPAMI)*, 2016.

[24] R. Peharz, G. Kapeller, P. Mowlaee, and F. Pernkopf. Modeling speech with sum-product networks: Application to bandwidth extension. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3699–3703, 2014.

[25] R. Peharz, S. Tschiatschek, F. Pernkopf, and P. Domingos. On theoretical properties of sum-product networks. In *Proceedings of International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 744–752, 2015.

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

[27] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko. Semi-supervised learning with ladder networks. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 3546–3554, 2015.

[28] S. Thomas, M.L. Seltzer, K. Church, and H. Hermansky. Deep neural network features and semi-supervised training for low resource speech recognition. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6704–6708, 2013.

[29] A. Vergari, N. Di Mauro, and F. Esposito. Simplifying, regularizing and strengthening sum-product network structure learning. In *Proceedings of European Conference on Machine Learning (ECML-PKDD)*, pages 343–358, 2015.

[30] Z. Yang, W. Cohen, and R. Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 40–48, 2016.

[31] X. Zhang, N. Guan, Z. Jia, X. Qiu, and Z. Luo. Semi-supervised projective non-negative matrix factorization for cancer classification. *PloS one*, 10(9):e0138814, 2015.

[32] X. Zhu and A.B. Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009.