
On the Relationship Between Probabilistic Circuits and Determinantal Point Processes

Honghua Zhang and Steven Holtzen and Guy Van den Broeck
Computer Science Department
University of California, Los Angeles
{hzhang19, sholtzen, guyvdb}@cs.ucla.edu

Abstract

Scaling probabilistic models to large realistic problems and datasets is a key challenge in machine learning. Central to this effort is the development of *tractable probabilistic models* (TPMs): models whose structure guarantees efficient probabilistic inference algorithms. The current landscape of TPMs is fragmented: there exist various kinds of TPMs with different strengths and weaknesses. Two of the most prominent classes of TPMs are determinantal point processes (DPPs) and probabilistic circuits (PCs). This paper provides the first systematic study of their relationship. We propose a unified analysis and shared language for discussing DPPs and PCs. Then we establish theoretical barriers for the unification of these two families, and prove that there are cases where DPPs have no compact representation as a class of PCs. We close with a perspective on the central problem of unifying these tractable models.

1 INTRODUCTION

Probabilistic modeling has become a central area of study in machine learning. The key challenge in applying probabilistic modeling is scaling to large datasets and models: in many cases, probabilistic inference quickly becomes intractable as the models grow in size and sophistication, and in general the task is #P-hard [Roth, 1996]. The field of *tractable probabilistic modeling* (TPM) seeks to identify classes of probabilistic models that (1) guarantee efficient probabilistic reasoning, and (2) can compactly represent interesting rich probability distributions.

In recent years there has been a proliferation of development of different classes of TPMs called *tractable probabilistic circuits* [Vergari et al., 2020, Choi et al., 2020].

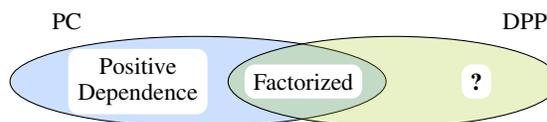


Figure 1: Known relationships between probabilistic circuits (PC) and determinantal point processes (DPP).

Each strikes a different balance between restrictions on the representation and increasing ease of learning and answering queries. Examples include bounded-treewidth graphical models [Meila and Jordan, 2000], sum-product networks (SPNs) [Poon and Domingos, 2011, Peharz et al., 2019], probabilistic sentential decision diagrams (PSDDs) [Kisa et al., 2014], arithmetic circuits [Darwiche, 2009], and cutset networks [Rahman and Gogate, 2016].

In a separate line of research, a TPM called *determinantal point processes* (DPPs) has been the topic of intense investigation, in particular because of their wide-ranging applications in machine learning [Kulesza and Taskar, 2012, Borodin, 2009, Krause and Guestrin, 2005]. They excel at representing certain types of distributions, but do so in ways that are distinct from how probabilistic circuits work. This raises the central questions of this paper: *Are DPPs and PCs really distinct in their ability to efficiently represent and reason about probability distributions?* Moreover, if we understand their relationship, *to what extent can their strengths and weaknesses be unified into one general TPM?* More broadly, one wonders, what is the essence of tractable probabilistic modeling?

Figure 1 summarizes what is known about this relationship. PCs can represent positive dependencies beyond the reach of DPPs, and simple factorized distributions can be represented in both TPMs. Our key contribution is to fill in the “?” that represents a gap in the literature: it is currently unknown whether circuits can represent DPPs.

Section 2 formally introduces DPPs and PCs and further motivates our research. Section 3 discusses the relative strengths and weaknesses of DPPs and PCs as tractable

families, and gives a semantic foundation for unifying them. Section 4 poses the problem of representing DPPs as circuits, and proves that such a representation is always inefficient. Finally, Section 5 outlines perspectives and future directions for unifying these two families.

2 BACKGROUND AND MOTIVATION

This paper studies probabilistic models that are representations of discrete probability distributions $\Pr(X_1, X_2, \dots)$, where X denotes a binary random variable. Assignments of values to the random variables are written $X = x$. Sets of binary random variables and their joint assignments are written in bold (e.g., $\mathbf{X} = \mathbf{x}$).

Our discussion focuses on two defining characteristics of probabilistic models: their *expressive efficiency* [Martens and Medabalimi, 2014] and *tractability*. We also refer to expressive efficiency as *succinctness* for short. A probabilistic model is efficient in terms of expressiveness (or succinct) for a class of distributions if it can compactly represent those distributions – i.e., the size of the model (for some appropriate definition of size) is polynomial in the number of random variables. A *query* – for instance, computing the marginal probability of an event – is *tractable* for a model if it can be computed in time polynomial in the size of the model.

In particular, we consider two well-known tractable probabilistic models that achieve their tractability in strikingly different ways: determinantal point processes (DPPs) and probabilistic circuits (PCs). Next, we formally introduce the semantics of their representation, and compare how they achieve their tractability, motivating the key research questions that this paper seeks to answer.

2.1 PROBABILISTIC REPRESENTATIONS

This section briefly describes DPPs and PCs through a unified notation and vocabulary.

2.1.1 DPPs as L-ensembles

Within machine learning, DPPs are most often represented by means of an *L-ensemble* [Borodin and Rains, 2005]:¹

Definition 1. A probability distribution \Pr over n binary random variables $\mathbf{X} = (X_1, \dots, X_n)$ is an *L-ensemble* if there exists a symmetric positive semidefinite matrix $L \in \mathbb{R}^{n \times n}$ such that for all $\mathbf{x} = (x_1, \dots, x_n) \in \{0, 1\}^n$,

$$\Pr(\mathbf{X} = \mathbf{x}) \propto \det(L_{\mathbf{x}}), \quad (1)$$

¹Although not every DPP is an L-ensemble, Kulesza and Taskar [2012] show that DPPs that assign non-zero probability to the empty set (the all-false assignment) are L-ensembles. Hence, this is a weak assumption in most applications.

where $L_{\mathbf{x}} = [L_{ij}]_{x_i=1, x_j=1}$ denotes the submatrix of L indexed by those i, j where $x_i = 1$ and $x_j = 1$. The matrix L is called the *kernel* for the L-ensemble.

To ensure that the distribution sums to one, it is necessary to divide Equation 1 by $\det(L + I)$, where I is the $n \times n$ identity matrix [Kulesza and Taskar, 2012].

A first barrier for linking DPPs and other probabilistic modeling frameworks is notational. While Definition 1 characterizes DPPs as distributions over n binary random variables \mathbf{X} , which is typical in the probabilistic graphical model literature, the DPP literature instead prefers to characterize them as distributions over *sets*. Fortunately there is a simple mapping between these interpretations.

Formally, given the finite *ground set* $\mathcal{Y} = \{1, \dots, n\}$, a DPP assigns a probability to each subset of \mathcal{Y} . The binary random variables \mathbf{X} define the random subset $\mathbf{Y} = \{i \in \mathcal{Y} : X_i = 1\}$; that is, we can view variable X_i as the indicator variable for item $i \in \mathcal{Y}$ being in the random set \mathbf{Y} . Similarly, each set assignment $\mathbf{Y} = A$ where $A \subset \mathcal{Y}$ corresponds to a binary assignment $\mathbf{X} = \mathbf{x}_A$ that sets $((X_i = 1)_{i \in A}, (X_j = 0)_{j \notin A})$.

Then, for any set value $A \subset \mathcal{Y}$, we can write

$$\Pr(\mathbf{Y} = A) = \Pr(\mathbf{X} = \mathbf{x}_A) \propto \det(L_{\mathbf{x}_A}) = \det(L_A),$$

where $L_A = [L_{ij}]_{i, j \in A}$ is the submatrix of L indexed by elements in A . For simplicity, we will denote $\mathcal{Y} - A$ by \bar{A} . Viewing a distribution from the perspective of subsets can be more intuitive for describing certain properties of DPPs – we will use both perspectives interchangeably.

Consider the following example of an L-ensemble defined over variables X_1, X_2 , and X_3 :

$$L = \begin{bmatrix} X_1 & X_2 & X_3 \\ 2 & 1.1 & 1.4 \\ 1.1 & 2.5 & 0.5 \\ 1.4 & 0.5 & 3 \end{bmatrix} \begin{matrix} X_1 \\ X_2 \\ X_3 \end{matrix}$$

Then, the probability of the assignment $\mathbf{X} = (1, 0, 1)$, or equivalently, subset $\mathbf{Y} = \{1, 3\}$ is given by:

$$\begin{aligned} \Pr(\mathbf{X} = (1, 0, 1)) &= \Pr(\mathbf{Y} = \{1, 3\}) \\ &= \frac{\det(L_{\{1,3\}})}{\det(L + I)} = \frac{1}{31.09} \begin{vmatrix} 2 & 1.4 \\ 1.4 & 3 \end{vmatrix} = 0.13. \end{aligned}$$

L-ensembles are clearly succinct: they are specified by $\mathcal{O}(n^2)$ parameters in kernel L . Moreover, evaluating the probability of an instantiation is efficient: it simply requires computing the determinant of a submatrix of kernel L , which takes $\mathcal{O}(n^3)$ time using Gaussian elimination.

One of the most important properties of DPPs is that they represent *global negative dependence*, which is discussed

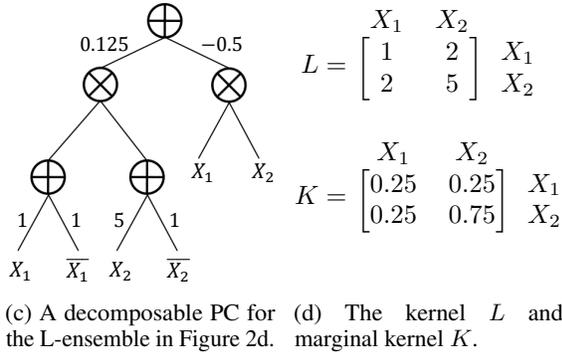
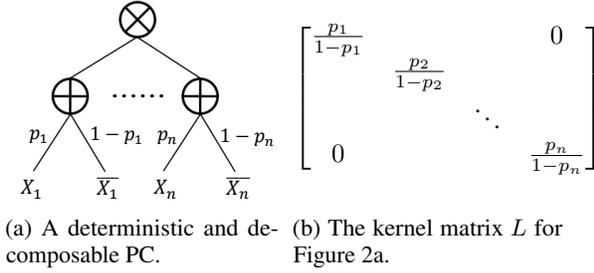


Figure 2: Motivating DPPs and their equivalent PCs.

in detail in Section 3. This property also brings a key limitation to DPPs: they cannot represent distributions with *any* positive dependence. Specifically, for a DPP, it must always be the case that $\Pr(X_i = 1, X_j = 1) \leq \Pr(X_i = 1) \Pr(X_j = 1)$. In Figure 1, probability distributions with positive dependence lie in the blue section.

2.1.2 Probabilistic Circuits

Next, we introduce the basics of probabilistic circuits [Vergari et al., 2020, Choi et al., 2020] – a class of probabilistic models that is strikingly different from L-ensembles.

Definition 2. A *probabilistic circuit* (PC) is a directed acyclic graph consisting of three types of nodes:

1. *Sums* \oplus with weighted edges to their children;
2. *Products* \otimes with unweighted edges to their children;
3. *Variable leaves*, here assumed to be X_i or \bar{X}_i .

For a given assignment $\mathbf{X} = \mathbf{x}$, the probabilistic circuit \mathcal{A} evaluates to a number $\mathcal{A}(\mathbf{x})$, which is obtained by (i) replacing variable leaves X_i by x_i , (ii) replacing variable leaves \bar{X}_i by $1 - x_i$, (iii) evaluating product nodes as taking the product over their children, and (iv) evaluating sum nodes as taking a weighted sum over their children.

Finally, a probabilistic circuit \mathcal{A} with variable leaves $\mathbf{X} = (X_1, \dots, X_n)$ represents the probability distribution

$$\Pr(\mathbf{X} = \mathbf{x}) \propto \mathcal{A}(\mathbf{x}). \quad (2)$$

Hence, evaluating the probability of an instantiation for a PC is simply a bottom-up pass, which is a linear-time algorithm in the size of the PC. As with DPPs, the semantics involve a normalizing constant. Section 2.2.2 will discuss how to compute this normalization efficiently by means of tractable marginalization.

A PC’s edge weights are called its *parameters*, which are usually assumed to be non-negative. We will relax this assumption in this paper, and also consider circuits with possibly negative parameters. Note that a PC with negative parameters does not necessarily represent a probability distribution.

An example PC that represents a fully-factorized distribution is given in Figure 2a. This circuit evaluates to

$$\Pr(\mathbf{X} = \mathbf{x}) = \mathcal{A}(\mathbf{x}) = \prod_i (x_i p_i + (1 - x_i)(1 - p_i)).$$

This fully-factorized distribution is also an L-ensemble, and the kernel L for which is shown in Figure 2b. Hence, we know that the intersection in Figure 1 is non-empty.

2.1.3 Motivating Questions

The prior sections outlined the semantics of DPPs and PCs for computing the probability of an instantiation of random variables. However, these representations are currently fundamentally distinct. A PC is a computation graph: it directly describes how to carry out the computation of an instantiation probability. A DPP, on the other hand, leaves unspecified how to compute the determinant. This hints at the following research questions:

- (Q1) Can an algorithm that computes determinants be efficiently captured by a computation graph that is a probabilistic circuit?
- (Q2) Or, do there exist DPPs that PCs are not expressive enough to represent succinctly?

We saw in the previous section that DPPs cannot represent distributions with positive dependence. In contrast, PCs are able to represent *any* probability distributions over binary random variables, including the ones with positive dependence. Indeed, we can always trivially construct a PC with a large \oplus root node that enumerates all 2^n instantiations with their probabilities as edge weights. The real question is, however, can PCs represent DPPs efficiently? To answer this question, we first need to understand the requirements for tractable inference

2.2 TRACTABLE PROBABILISTIC INFERENCE

The previous section defined L-ensembles and PCs, and showed that evaluating the probability of complete assignments is tractable in both models. However, the purpose

of tractable probabilistic models is to answer more interesting queries efficiently. This section briefly discusses the probabilistic inference tasks that can be performed tractably for DPPs and PCs; in particular, we will focus on two tasks: computing marginals and MAP inference.

2.2.1 Tractability of DPPs

As an alternative to L-ensemble kernels, DPPs can more generally be represented by their *marginal kernels*.

Definition 3. A probability distribution \Pr over n binary random variables $\mathbf{X} = (X_1, \dots, X_n)$ is a *determinantal point process* if there exists a symmetric positive semidefinite matrix $K \in \mathbb{R}^{n \times n}$ such that for all $A \subset \{1, \dots, n\}$,

$$\Pr((X_i = 1)_{i \in A}) = \det(K_{\mathbf{x}_A}), \quad (3)$$

or, equivalently, from the perspective of subsets,

$$\Pr(A \subset \mathbf{Y}) = \det(K_A).$$

The matrix K is called the *marginal kernel* for the DPP.

Note that we are computing a marginal probability: variables X_i where $i \notin A$ are marginalized out. From the set perspective, we are asking whether the random subset contains the elements of A , not whether it is equal to A .

One can use a generalized version of Equation 3 to compute the general marginals $\Pr((X_i = 1)_{i \in A}, (X_j = 0)_{j \in B})$ efficiently, where $A, B \subseteq \{1, \dots, n\}$. We refer to Kulesza and Taskar [2012] for further details.

A seminal result in the DPP literature is that every L-ensemble with kernel L is a DPP with marginal kernel $K = L(L + I)^{-1}$ [Macchi, 1975]. In light of this, it is clear that computing marginals is efficient for L-ensembles in general, because computing the inverse and determinant can both be done in polynomial time. However, MAP inference, where the task is to find the most likely world given evidence, is known to be NP-hard for L-ensembles [Ko et al., 1995].

2.2.2 Tractability of PCs

Next we discuss the same queries for PCs – we will show they are tractable for entirely different, structural reasons.

For an arbitrary probabilistic circuit, most probabilistic inference tasks, including marginals and MAP inference, are computationally hard in the circuit size and therefore inefficient. In order to guarantee the efficient evaluation of queries it is therefore necessary to impose further constraints on the structure of the circuit. There are two well-known structural properties of probabilistic circuits [Darwiche and Marquis, 2002, Choi et al., 2020]:

Definition 4. For a probabilistic circuit,

1. A \otimes node is *decomposable* if its inputs depend on disjoint sets of variable nodes.
2. A \oplus node is *deterministic* if at most one of its inputs can be non-zero for any assignment to the variable leaf nodes.

We say a PC is decomposable if all of its \otimes nodes are decomposable; a PC is deterministic if all of its \oplus nodes are deterministic.

In this paper, we will also consider a special class of PCs: Probabilistic Sentential Decision Diagrams (PSDDs) [Kisa et al., 2014]. At a high-level, a PSDD is a deterministic *and* decomposable PC with some additional structural properties. Because of these structural properties, PSDDs come with additional tractability guarantees [Khosravi et al., 2019] and strong local properties, which will be explained later; for further details please refer to Kisa et al. [2014].

Let \mathcal{A} be a PC over X_1, \dots, X_n . If \mathcal{A} is decomposable, then we can efficiently compute its marginals: for disjoint $A, B \subset \{1, \dots, n\}$, the marginal probability $\Pr((X_i = 1)_{i \in A}, (X_j = 0)_{j \in B})$ is given by the evaluation of \mathcal{A} with the following input:

$$\begin{cases} X_i = 1, \overline{X}_i = 0 & \text{if } i \in A \\ X_i = 0, \overline{X}_i = 1 & \text{if } i \in B \\ X_i = 1, \overline{X}_i = 1 & \text{otherwise.} \end{cases}$$

Thus, for a decomposable PC, the time complexity of marginal computation is linear in the size of the circuit.

Though computing marginals is tractable for both L-ensembles and decomposable PCs, it is done very differently. For a decomposable PC, we only need to evaluate the circuit with respect to certain input configurations. That is to say, if we can represent an L-ensemble with a decomposable PC, we would have a unified representation for both the kernel L and marginal kernel K of an L-ensemble. For example, Figure 2c shows a decomposable PC that represents the L-ensemble in Figure 2d; say we want to compute the marginal distribution $\Pr(X_2 = 1)$: for the PC, we plug in $X_1 = 1, \overline{X}_1 = 1, X_2 = 1, \overline{X}_2 = 0$ and it evaluates to 0.75, which corresponds to the value of the entry $K_{22} = \Pr(2 \in \mathbf{Y})$.

For a deterministic *and* decomposable PC, MAP inference can be done in time polynomial in the size of the PC, which implies that MAP inference is also tractable for PSDDs. When a PC is unnormalized, we also need to compute its normalizing constant, the time complexity of which is linear in the size of a decomposable PC.

3 A UNIFIED TRACTABLE MODEL

A unique property of DPPs is that they are *tractable* representations of probability distributions that express *global negative dependence*. Graphical models are limited by their local nature thus cannot effectively model such global negative dependence. For example, consider a DPP that assigns non-zero probabilities to all *proper* subsets of $\{1, \dots, n\}$ and zero probability to the whole set: a Markov Random Field that is not the complete graph (or a factor graph without a factor that connects all nodes), cannot model this distribution. To further demonstrate what global negative dependence means we consider the geometric interpretation of determinants [Kulesza and Taskar, 2012].

Consider an L-ensemble P_L . Since kernel L is positive semidefinite, by the Spectral Theorem, there exists a matrix B such that $L = B^T B$. We denote the columns of B by B_i , where $1 \leq i \leq n$. Then for $A \subset \mathcal{Y}$,

$$P_L(\mathbf{Y} = A) \propto \det(L_A) = \text{Vol}^2(\{B_i\}_{i \in A})$$

where $\text{Vol}^2(\{B_i\}_{i \in A})$ is the squared $|A|$ -dimensional volume of the parallelepiped spanned by $\{B_i\}_{i \in A}$. Intuitively, we can view B_i as the feature vector for element $i \in \mathcal{Y}$. If the elements in a set $A \subset \mathcal{Y}$ are very similar to each other, then the angles between their feature vectors are small; then, the volume spanned by their feature vectors are small, which implies that the probability of A (and any set that contains A) is small. From this perspective, we can see that DPPs always encourage *diversity* by assigning higher probabilities to sets of elements that are more different from each other. The ability to tractably model diversity makes DPP a useful class of TPMs in many applications [Mariet and Sra, 2016], such as document and video summarization [Chao et al., 2015, Lin and Bilmes, 2012], recommender systems [Zhou et al., 2010], and object retrieval [Affandi et al., 2014].

Despite being a tractable model that expresses diversity and negative dependencies, DPP is not very flexible in the sense it cannot model *any* positive correlations. Hence, there arises the question whether we can tractably model such diversity by a more flexible TPM that allows some positive correlations. Thus, we consider the problem of tractably representing DPPs (or L-ensembles) by PCs, which are much more flexible than DPPs.

3.1 DPPs AS PCs

In this section we discuss some potential solutions to the problem of representing L-ensembles by PCs; we start by introducing the *symbolic kernel* for an L-ensemble.

Definition 5. Let $L \in \mathbb{R}^{n \times n}$ be the kernel for an L-ensemble over X_1, \dots, X_n ; the corresponding *symbolic*

kernel L^* is given by:

$$L^*_{ij} = \begin{cases} L_{ij}X_i + \overline{X_i} & \text{if } i = j \\ L_{ij}X_iX_j & \text{if } i \neq j \end{cases}$$

With the symbolic kernel L^* , probabilities of an L-ensemble can be written in a different way:

$$\Pr(\mathbf{X} = \mathbf{x}) = \det(L + I)^{-1} (\det(L^*)|_{\mathbf{x}=\mathbf{x}}),$$

where the notation $|_{\mathbf{x}=\mathbf{x}}$ replaces the symbolic variables with assignments.

To get a concrete understanding of the symbolic kernel, we consider the example L-ensemble presented in Section 2.1.1; the corresponding symbolic kernel is

$$L^* = \begin{bmatrix} 2X_1 + \overline{X_1} & 1.1X_1X_2 & 1.4X_1X_3 \\ 1.1X_1X_2 & 2.5X_2 + \overline{X_2} & 0.5X_2X_3 \\ 1.4X_1X_3 & 0.5X_2X_3 & 3X_3 + \overline{X_3} \end{bmatrix}.$$

Then the probability of the instantiation $(1, 0, 1)$ is

$$\begin{aligned} & \Pr(X_1 = 1, X_2 = 0, X_3 = 1) \\ &= \det(L + I)^{-1} \det(L^*)|_{X_1=1, X_2=0, X_3=1} \\ &= \frac{1}{\det(L + I)} \begin{vmatrix} 2 & 0 & 1.4 \\ 0 & 1 & 0 \\ 1.4 & 0 & 3 \end{vmatrix} = \frac{1}{\det(L + I)} \begin{vmatrix} 2 & 1.4 \\ 1.4 & 3 \end{vmatrix} \end{aligned}$$

Note that each entry of the symbolic kernel L^* can be easily represented by a PC; hence, with the symbolic kernel, the problem of representing DPPs by polynomial-size PCs can be reduced to the problem of representing the determinant function (with entries of the matrix being the inputs) by a polynomial-size PC.

Our first observation is that a PC that computes the determinant function must have at least one *negative* parameter, which is a non-standard assumption for PCs: a PC with non-negative parameters would always output non-negative numbers for matrices with all positive entries, which, however, could have negative determinant.

Our second observation is that constructing a polynomial-size PC that computes the determinant is not easy. Gaussian elimination, for instance, would not work here as it requires branching and division. The Laplace expansion of the determinant does not require branching or division, but it has to compute the determinants of exponentially many submatrices. Nonetheless, a combinatorial algorithm proposed by Mahajan and Vinay [1997] gives us a polynomial-size PC for the determinant; unfortunately we observe that the induced PC that represents DPPs is neither deterministic nor decomposable.

Of course, one might say that we *do not have to* reduce the problem of representing DPPs by PCs to the problem

of constructing a PC that computes the determinant. We will show in the following section that, roughly speaking, polynomial-size PCs that represents DPPs cannot be deterministic or decomposable.

4 BARRIERS TO REPRESENTING DPPs WITH PCs

In this section, we prove that in general, DPPs cannot be tractably represented by certain subclasses of PCs. In particular, we consider three different classes of PCs: PSDDs, deterministic PCs and decomposable PCs. By “tractable representation,” we mean PCs of polynomial size with respect to the number of input variables.

4.1 PSDDs CANNOT TRACTABLY REPRESENT ALMOST ALL L-ENSEMBLES

Though we will show in the following subsections that neither deterministic PCs nor decomposable PCs can tractably represent DPPs in general, for PSDDs, we prove a much stronger result. Instead of constructing a small subclass of DPPs that cannot be tractably represented by PSDDs, we prove that *almost all* L-ensembles have no polynomial-size PSDD representations, which we formalize as the following theorem.

Theorem 1. *Let $n \in \mathbb{N}^+$, $M > 0$; let B be a matrix drawn from the uniform distribution over $[-M, M]^{n \times n}$. Let $L = B^T B$. Then with probability 1, the L-ensemble with kernel L cannot be tractably represented by any PSDD.*

The proof for Theorem 1 relies on *algebraically independent numbers*:

Definition 6. Let F be a field; define $F[t_1, \dots, t_k]$ to be the set of polynomials over *indeterminates* t_1, \dots, t_k with coefficients in F . We call 0 the *trivial polynomial*.

Definition 7. Let $S = \{a_1, \dots, a_k\} \subset \mathbb{R}$; S is said to be *algebraically independent over \mathbb{Q}* if $f(a_1, \dots, a_k) \neq 0$ for all non-trivial $f \in \mathbb{Q}[t_1, \dots, t_k]$. S is called *algebraically dependent* if it is not algebraically independent.

Algebraically independent numbers over \mathbb{Q} are “almost everywhere” in the euclidean space:

Lemma 1. *Let $M > 0$, $n \in \mathbb{N}^+$. Consider the uniform measure (distribution) over $[-M, M]^n$; then the measure of the set $\mathcal{S} = \{(a_i)_{1 \leq i \leq n} \in [-M, M]^n : \{a_i\} \text{ is algebraically independent over } \mathbb{Q}\}$ is 1.*

By Lemma 1, we know that if we sample a (real) matrix uniformly at random, with probability 1, its entries form an algebraically independent set over \mathbb{Q} . To prove Theorem 1, we prove the following intermediate result:

Lemma 2. *Let $B \in \mathbb{R}^{n \times n}$ such that $\{B_{ij}\}$ forms a set of algebraically independent numbers over \mathbb{Q} . Then the size of any PSDD that computes the L-ensemble with kernel $L = B^T B$ is at least 2^{n-1} .*

It is clear that Theorem 1 immediately follows from Lemma 1 and Lemma 2, and we briefly sketch the proof for Lemma 2 here. The proof for Lemma 2 relies on the following proposition due to Shen et al. [2016], which connects the size of a PSDD to the number of distinct conditional probability distributions that it must encode:

Proposition 1 (Shen et al. [2016]). *For a PSDD over the variables X_1, \dots, X_n , there exists a variable X_q such that for each $\mathbf{x} \in \{0, 1\}^{n-1}$, there is a PSDD node that represents the conditional distribution $\Pr(X_q | (X_j)_{j \neq q} = \mathbf{x})$. In particular, if $\Pr(X_q = 1 | (X_j)_{j \neq q} = \mathbf{x}_1) \neq \Pr(X_q = 1 | (X_j)_{j \neq q} = \mathbf{x}_2)$, then the PSDD nodes that represent them must be distinct.*

By Proposition 1, we know that if we have a probability distribution such that for all $q \in \{1, \dots, n\}$, $\Pr(X_q = 1 | (X_j)_{j \neq q} = \mathbf{x})$ is different for all $\mathbf{x} \in \{0, 1\}^{n-1}$, then any PSDD that represents this distribution will have at least 2^{n-1} nodes. In particular, to prove Lemma 1, we only need to show that the L-ensemble in Lemma 2 satisfies this property; that is, Lemma 2 follows from Proposition 1 and the following result:

Proposition 2. *Let $\mathcal{Y} = \{1, \dots, n\}$ be the ground set. Consider the L-ensemble P_L over X_1, \dots, X_n with kernel $L = B^T B$, where $\{B_{ij}\}$ is a set of algebraically independent numbers over \mathbb{Q} . Let \mathbf{Y} be a random subset drawn from P_L . Then $\forall q \in \{1, \dots, n\}$, the following two equivalent statements hold:*

1. $P_L(X_q = 1 | (X_j)_{j \neq q} = \mathbf{x})$ is different for all $\mathbf{x} \in \{0, 1\}^{n-1}$.
2. $P_L(\mathbf{Y} = A^{in} \cup \{q\} | A^{in} \subset \mathbf{Y}, A^{out} \cap \mathbf{Y} = \emptyset)$ is different for all disjoint union $\mathcal{Y} = A^{in} \cup A^{out} \cup \{q\}$.

Instead of presenting the complete proof for Proposition 2, which is given in the appendix, we leave a short remark to conclude this subsection. The key point to our proof for Lemma 2 is that $f_{A,B,q}$, the polynomials defined in Proposition 3, do not evaluate to 0 at point (B_{ij}) when the set $\{B_{ij}\}$ is algebraically independent over \mathbb{Q} . Hence by a continuity argument, it can be shown that as long as a point (P_{ij}) is close enough to (B_{ij}) , the L-ensemble with kernel $P^T P$ cannot be tractably represented by PSDDs. As rational points are dense, we can strengthen Theorem 1 by claiming that besides algebraically independent numbers, “many” L-ensembles with *only rational entries* cannot be represented by polynomial-size PSDDs either.

4.2 DETERMINISTIC PCs CANNOT TRACTABLY REPRESENT DPPs

In this section we prove that in general, DPPs cannot be tractably represented by deterministic PCs.

Theorem 2. *Let $\mathcal{Y} = \{1, \dots, n\}$ be the ground set. For all $n \geq 2$, there exists an L -ensemble P_n over the subsets of \mathcal{Y} such that the number of the parameters in any deterministic AC that represents P_n is at least $2^n - 2$.*

To prove Theorem 2, we first consider the constraints that determinism put on PCs:

Lemma 3. *Let \mathcal{A} be a deterministic PC over variables \mathbf{X} ; let $\theta_1, \dots, \theta_m \in \mathbb{R}$ be the parameters in \mathcal{A} . Let f be the function that \mathcal{A} computes. Then, for any instantiation $\mathbf{X} = \mathbf{x}$, $f(\mathbf{x}) = 0$ or $\prod_{1 \leq i \leq m} \theta_i^{a_i}$ for some $a_i \in \mathbb{N}$.*

Before presenting the formal proof for Theorem 2, we first sketch the intuition here. For a deterministic PC over n variables, there are 2^n possible instantiations to its variables, which yields 2^n probabilities denoted by p_0, \dots, p_{2^n-1} . By Lemma 3, we know that every p_i is either 0 or some products of the parameters in the PC. If the number of parameters is less than 2^n , then the p_i s are related to each other by multiplication; that is, if $p_i \neq 0$, then $p_i^r = \prod_{j \neq i} p_j^{r_j}$ for some $r, r_j \in \mathbb{N}$. Now the proof for Theorem 2 reduces to constructing a class of L -ensembles such that for all i the equalities $p_i^r = \prod_{j \neq i} p_j^{r_j}$ do not hold, which we formalize as condition 2 in Lemma 4, and this is the key property that we are aiming for.

Lemma 4. *There exists a sequence (indexed by n) of matrices $L_n \in \mathbb{R}^{n \times n}$ where each L_n is of the form*

$$L_n = \begin{bmatrix} d_1 & 1 & \dots & 1 \\ 1 & d_2 & & 1 \\ \vdots & & \ddots & \vdots \\ 1 & \dots & & d_n \end{bmatrix}$$

and satisfies the following conditions

1. $\det((L_n)_A) > 1, \forall A \subset [n]$ s.t. $A \neq \emptyset$, which implies that L_n defines an L -ensemble.
2. $(\det((L_n)_A))^r \neq \prod_{B \subset [n], B \neq A} (\det((L_n)_B))^{r_B}, \forall A \subset [n], r, r_B \in \mathbb{N}$ s.t. $A \neq \emptyset, r \neq 0$.

Now we proceed to prove Theorem 2.

Proof for Theorem 2. Let $L \in \mathbb{R}^{n \times n}$ be the n th kernel matrix that we constructed in Lemma 4; let P be the DPP defined by L . Let \mathcal{A} be a deterministic PC with parameters $\{\theta_i\}_{1 \leq i \leq m}$. Assume that \mathcal{A} computes the unnormalized P ; that is, assume $\mathcal{A}(\mathbf{x}) = \det(L_{\mathbf{x}})$ for all $\mathbf{x} \in \{0, 1\}^n$. Suppose (towards a contradiction) that $m < 2^n - 1$. We write the set $\{\det(L_A) : A \subset$

$[n]\}$ as $\{p_i\}_{0 \leq i < 2^n}$ where $p_0 = \det(L_\emptyset) = 1$. By Lemma 4, $p_i > 1 \forall i \neq 0$. By Lemma 3, we have: $p_i = \prod_{1 \leq j \leq m} \theta_j^{v_{ij}}$ for some $v_{ij} \in \mathbb{N}, 0 < i < 2^n$.

Put $v_i = [v_{i1}, \dots, v_{im}]^T \in \mathbb{Q}^m, \forall 0 < i < 2^n$. Let $S = \{v_i\}_{0 < i < 2^n}$. Note that $\forall 0 < i < 2^n, p_i > 1 \implies v_i \neq 0$. Since \mathbb{Q}^m is an m -dimensional vector space over \mathbb{Q} , $|S| = 2^n - 1 > m \implies S$ is linearly dependent. Without loss of generality, $v_1 = \sum_{2 \leq k < 2^n} q_k v_k$ for some $q_k \in \mathbb{Q}$, where some q_k are non-zero. Hence, $v_{1j} = \sum_{2 \leq k < 2^n} q_k v_{kj}$; then,

$$\begin{aligned} p_1 &= \prod_{1 \leq j \leq m} \theta_j^{v_{1j}} = \prod_{1 \leq j \leq m} \theta_j^{\sum_{2 \leq k < 2^n} q_k v_{kj}} \\ &= \prod_{1 \leq j \leq m} \prod_{2 \leq k < 2^n} \theta_j^{q_k v_{kj}} = \prod_{2 \leq k < 2^n} \left(\prod_{1 \leq j \leq m} \theta_j^{v_{kj}} \right)^{q_k} \\ &= \prod_{2 \leq k < 2^n} p_k^{q_k} \end{aligned}$$

Write $q_k = \frac{r_k}{s_k}$; put $l = \text{lcm}\{s_k : q_k \neq 0\}$. l is well-defined and clearly non-zero. Then,

$$p_1^l = \prod_{2 \leq k < 2^n} p_k^{r_k l}$$

That is, for some natural numbers $\{r_B\}_{B \subset [n]}, A \subset [n]$,

$$(\det(L_A))^l = \prod_{B \subset [n], B \neq A} (\det(L_B))^{r_B}$$

where $l \neq 0$; contradicting the choice of L . Hence $m \geq 2^n - 1$ must follow. Now assume that \mathcal{A} computes P (i.e. normalized); suppose $m < 2^n - 2$; then we can easily obtain a deterministic AC \mathcal{A}' that has $m + 1 < 2^n - 1$ parameters and computes the unnormalized P ; by contradiction, $m \geq 2^n - 2$ must follow. \square

4.3 NO TRACTABLE SPNs FOR DPPs

In this section, we prove that DPPs in general cannot be tractably represented by decomposable PCs with no negative parameters, often called sum-product networks (SPNs) [Poon and Domingos, 2011]. In particular, the uniform distribution over spanning trees on complete graphs is a DPP that has no polynomial-size SPN representation.

Theorem 3. *For all $n \geq 20$, let $\mathcal{Y} = \{1, 2, \dots, \binom{n}{2}\}$ be the ground set; then there exists a DPP over the subsets of \mathcal{Y} such that the size of any smooth SPN that represents the DPP is at least $2^{n/30240}$.*

An SPN is *smooth* if for each \oplus gate, all of its children mention the same set of variables [Darwiche, 2000]. Decomposable circuits can be made smooth efficiently [Shih

et al., 2019]. Hence, Theorem 3 immediately implies that in general, DPPs cannot be tractably represented by SPNs.

Now we give the connection to distributions on spanning trees. Let K_n be the complete graph on n vertices. For each edge e_{ij} with $1 \leq i < j \leq n$, we associate an indicator variable X_{ij} . By Cayley’s formula there are n^{n-2} spanning trees on K_n [Cayley, 1889]. Hence, we can define a distribution on $\mathbf{X} = (X_{ij})_{1 \leq i < j \leq n}$ by setting $\Pr(\mathbf{X} = \mathbf{x}) = 1/n^{n-2}$ if the set $\{e_{ij} : x_{ij} = 1\}$ forms a spanning tree on K_n and 0 otherwise. We call it the *uniform distribution over spanning trees* on the complete graph. Note that the distribution is defined on $\binom{n}{2}$ variables. The uniform distribution over spanning trees is a DPP [Snell, 1995]:

Theorem 4. *The uniform distribution over spanning trees on the complete graph K_n is a DPP, where the marginal kernel is given by:*

$$K(e_{ij}, e_{lk}) = \begin{cases} \frac{2}{n} & \text{if } i = k, j = l \\ \frac{1}{n} & \text{if } i = k, j \neq l \text{ or } i \neq k, j = l \\ -\frac{1}{n} & \text{if } j = k \text{ or } i = l \\ 0 & \text{otherwise} \end{cases}$$

where $1 \leq i < j \leq n, 1 \leq l < k \leq n$

To get a concrete understanding of the theorem, consider the following example where $n = 4$. Here for the complete graph with 4 vertices, the marginal kernel K is

$$\begin{matrix} & e_{12} & e_{13} & e_{14} & e_{23} & e_{24} & e_{34} \\ \begin{matrix} e_{12} \\ e_{13} \\ e_{14} \\ e_{23} \\ e_{24} \\ e_{34} \end{matrix} & \begin{bmatrix} 1/2 & 1/4 & 1/4 & -1/4 & -1/4 & 0 \\ 1/4 & 1/2 & 1/4 & 1/4 & 0 & -1/4 \\ 1/4 & 1/4 & 1/2 & 0 & 1/4 & 1/4 \\ -1/4 & 1/4 & 0 & 1/2 & 1/4 & -1/4 \\ -1/4 & 0 & 1/4 & 1/4 & 1/2 & 1/4 \\ 0 & -1/4 & 1/4 & -1/4 & 1/4 & 1/2 \end{bmatrix} \end{matrix}$$

To compute the number of spanning trees that contain edges e_{12} and e_{24} , we compute the determinant of the submatrix indexed by e_{12} and e_{24} :

$$\Pr(X_{12} = 1, X_{24} = 1) = \begin{vmatrix} 1/2 & -1/4 \\ -1/4 & 1/2 \end{vmatrix} = 3/16$$

The total number of spanning trees on K_4 is $4^{4-2} = 16$; hence, the number of spanning trees that contain e_{12} and e_{24} is given by $3/16 \times 16 = 3$, illustrated by Figure 3.

Now we present the second result: the uniform distributions over spanning trees cannot be tractably represented by SPNs.²

Theorem 5 (Martens and Medabalimi [2014]). *Let P_n be the uniform distribution over spanning trees on K_n . For $n \geq 20$, the size of any smooth SPN that represents P_n is at least $2^{n/30240}$.*

²Note that Theorem 3 only holds for DPPs, because the uniform distribution over spanning trees is *not* an L-ensemble.

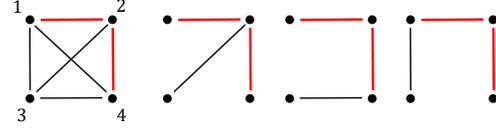


Figure 3: The complete graph on 4 vertices K_4 and the 3 spanning trees that contain edges e_{12} and e_{24}

Theorem 3 immediately follows from Theorems 4 and 5.

5 PERSPECTIVES

So far we have proved three negative results on the problem of representing DPPs by PCs and we briefly summarize them as follows:

1. For an L-ensemble with kernel $L = B^T B$, where $B \in \mathbb{R}^{n \times n}$ is randomly generated, with probability 1, the L-ensemble cannot be tractably represented by PSDDs.
2. There exists a class of L-ensembles that cannot be tractably represented by deterministic PCs with (possibly) negative parameters.
3. There exists a class of DPPs that cannot be tractably represented by SPNs with non-negative parameters.

This paper closes many avenues for finding a unified tractable probabilistic model. In this section we discuss some promising avenues that remain open. In particular, we propose the problem of representing certain *subclasses* of DPPs by polynomial-size PCs.

5.1 A SIMPLE CASE STUDY

As shown in Figure 2, there is one degenerate class of DPPs that we can tractably represent by deterministic and decomposable PCs: the fully factorized distributions. This subclass of DPP is uninteresting because it expresses no dependence among the variables. Inspired by the Matrix Determinant Lemma [Harville, 1998], which is Lemma 6 in the appendix, we consider a class of non-trivial L-ensembles that exhibits negative dependence among the variables. In particular, for the ground set $\mathcal{Y} = \{1, \dots, n\}$, we consider *rank-1 perturbation* (RIP) L-ensembles with kernels of the form $L = D + \lambda u u^T$, where $D = \text{diag}(d_1, \dots, d_n)$, $d_i \geq 0, \lambda \geq 0$ and $u = (u_1, \dots, u_n)^T \in \mathbb{R}^n$ a normal vector. It is clear that for kernel L , the off-diagonal entries $L_{ij} = \lambda u_i u_j$ can be set to non-zero. Hence, this special class of L-ensembles *does* exhibit negative dependence among variables to some extent, which is one of the key properties that makes DPPs a unique class of TPMs. Now we ask two questions: (1) can RIP L-ensembles be represented tractably by structured PCs; (2) can other more complex

m	Fully-Factorized KL	R1P Mixture KL	Ratio
1	0.23406	0.23240	1.00
2	0.14948	0.14778	1.01
8	0.03963	0.03690	1.07
16	0.01373	0.01077	1.27
24	0.00554	0.00381	1.45
32	0.00264	0.00162	1.62
40	0.00125	0.00062	2.02
47	0.00054	0.00027	1.99

Table 1: Results for the motivating experiment on approximating an L-ensemble over $N = 10$ variables. The second and third column gives the KL divergence between the L-ensemble and the approximating family. The fourth column highlights the trend by giving the ratio between the second and third column.

DPPs be represented as mixtures of R1Ps.

Let \mathbf{Y} be a subset drawn from the L-ensemble with kernel L ; let $A \subseteq \mathcal{Y}$, then by the matrix determinant lemma:

$$\Pr(\mathbf{Y} = A) = \frac{1}{Z} \left(\prod_{i \in A} d_i + \sum_{i \in A} u_i^2 \prod_{j \in A, j \neq i} d_j \right),$$

where the normalizing constant $Z = \det(D + \lambda uu^T + I)$. This formula immediately gives us a decomposable PC that represents L , which is shown in Figure 4 in the appendix; we denote the circuit by \mathcal{C} . Thus we have an example of an interesting DPP represented as a polynomial-size structured PC, answering (1) in the positive.

Now we explore question (2) empirically on a small example. In a preliminary study, we approximate an L-ensemble P_L with kernel $L = B^T B$, where $B \in \mathbb{R}^{K \times N}$ is randomly generated. To do this, we take a weighted sum of m instances of \mathcal{C} to approximate P_L , which we call a *R1P mixture*. For different values of m , we compare our mixture model against the weighted sum of $2m$ instances of the fully-factorized distribution (the baseline); this makes the comparison fair in the sense that both models have $2mN$ parameters. For both models, the parameters are learned by stochastic gradient descent to minimize the KL-divergence between the model and P_L . We initialize the parameters randomly and record the best results (minimum KL-divergence) from 20 instances of random restarts.

Table 1 gives the results of this preliminary experiment. It shows that this is a potential promising avenue for approximating DPPs with structured circuits, motivating future empirical and theoretical work. In particular, we see that the ratio between the KL-divergence for the baseline and the R1P mixture increases as the number of components in the mixtures increases.

5.2 DISCUSSION AND CONCLUSION

We conclude by highlighting future directions and contextualizing our preliminary experimental results. Though we have proved that PSDDs, deterministic PCs with (possibly) negative parameters and SPNs cannot tractably represent DPPs in general, some problems on the existence of polynomial-size PCs that represent DPPs are left open:

1. Can the uniform distribution over spanning trees on complete graphs be tractably represented by decomposable PCs with negative parameters?
2. Can L-ensembles be tractably represented by decomposable PCs (with or without negative parameters)?

These two open problems are important not only because they are the missing pieces for a complete answer to the problem of representing DPPs by polynomial-size PCs, but also because their answers will deepen our understanding about more general questions. An answer to question 1 will help us better understand the expressive power that PCs gain by allowing negative parameters.

If the answer to question 2 is positive; then we immediately have a class of decomposable PCs that can tractably represent L-ensembles, which implies that there exists a class of decomposable PCs that represents probability distributions with global negative dependence, which, as mentioned before, is intractable for some graphical models. Besides, starting from polynomial-size decomposable PCs that represents L-ensembles, we can obtain decomposable PCs that exhibit global negative dependence and at the same time allow for greater flexibility than L-ensembles do. If the answer to question 2 is negative, it will still deepen our understanding about the expressiveness of polynomial-size PCs.

To conclude, this work identifies and makes progress towards closing a gap between two well-known families of TPMs. We hope that this effort motivates future work in extending our understanding of the expressiveness of probabilistic circuits and motivates further efforts in combining and comparing different classes of TPMs.

Acknowledgements

We thank Andreas Krause, Arthur Choi, Brendan Juba, Pasha Khosravi, Anji Liu, Tao Meng, Antonio Vergari and Zhe Zeng for helpful feedback and discussion. In particular, we thank Richard Elman for detailed and insightful discussion on algebra. This work is partially supported by NSF grants #IIS-1943641, #IIS-1633857, #CCF-1837129, DARPA XAI grant #N66001-17-2-4032, Sloan and UCLA Samueli Fellowships, and gifts from Intel and Facebook Research.

References

- Raja Hafiz Affandi, Emily Fox, Ryan Adams, and Ben Taskar. Learning the parameters of determinantal point process kernels. In *International Conference on Machine Learning*, pages 1224–1232, 2014.
- Alexei Borodin. Determinantal point processes. *arXiv preprint arXiv:0911.1153*, 2009.
- Alexei Borodin and Eric M Rains. Eynard–mehta theorem, schur process, and their pfaffian analogs. *Journal of statistical physics*, 121(3-4):291–317, 2005.
- Arthur Cayley. A theorem on trees. *Quart. J. Math.*, 23: 376–378, 1889.
- Wei-Lun Chao, Boqing Gong, Kristen Grauman, and Fei Sha. Large-margin determinantal point processes. In *UAI*, pages 191–200, 2015.
- YooJung Choi, Antonio Vergari, and Guy Van den Broeck. Probabilistic circuits: A unifying framework for tractable probabilistic modeling. 2020.
- Adnan Darwiche. On the tractable counting of theory models and its application to belief revision and truth maintenance. *arXiv preprint cs/0003044*, 2000.
- Adnan Darwiche. *Modeling and reasoning with Bayesian networks*. Cambridge university press, 2009.
- Adnan Darwiche and Pierre Marquis. A knowledge compilation map. *Journal of Artificial Intelligence Research*, 17:229–264, 2002.
- David A Harville. Matrix algebra from a statistician’s perspective, 1998.
- Pasha Khosravi, YooJung Choi, Yitao Liang, Antonio Vergari, and Guy Van den Broeck. On tractable computation of expected predictions. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, 2019.
- Doga Kisa, Guy Van den Broeck, Arthur Choi, and Adnan Darwiche. Probabilistic sentential decision diagrams. In *Fourteenth International Conference on the Principles of Knowledge Representation and Reasoning*, 2014.
- Chun-Wa Ko, Jon Lee, and Maurice Queyranne. An exact algorithm for maximum entropy sampling. *Operations Research*, 43(4):684–691, 1995.
- Andreas Krause and Carlos Guestrin. *A note on the budgeted maximization of submodular functions*. Carnegie Mellon University. Center for Automated Learning and Discovery, 2005.
- Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286, 2012. ISSN 1935-8237. doi: 10.1561/22000000044. URL <http://dx.doi.org/10.1561/22000000044>.
- Hui Lin and Jeff A Bilmes. Learning mixtures of submodular shells with application to document summarization. *arXiv preprint arXiv:1210.4871*, 2012.
- Odile Macchi. The coincidence approach to stochastic point processes. *Advances in Applied Probability*, 7(1): 83–122, 1975.
- Meena Mahajan and V Vinay. A combinatorial algorithm for the determinant. In *SODA*, pages 730–738, 1997.
- Zelda E Mariet and Suvrit Sra. Kronecker determinantal point processes. In *Advances in Neural Information Processing Systems*, pages 2694–2702, 2016.
- James Martens and Venkatesh Medabalimi. On the expressive efficiency of sum product networks. *arXiv preprint arXiv:1411.7717*, 2014.
- Marina Meila and Michael I Jordan. Learning with mixtures of trees. *Journal of Machine Learning Research*, 1(Oct):1–48, 2000.
- Robert Peharz, Antonio Vergari, Karl Stelzner, Alejandro Molina, Xiaoting Shao, Martin Trapp, Kristian Kersting, and Zoubin Ghahramani. Random sum-product networks: A simple but effective approach to probabilistic deep learning. In *Proceedings of UAI*, 2019.
- Hoifung Poon and Pedro Domingos. Sum-product networks: A new deep architecture. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 689–690. IEEE, 2011.
- Tahrima Rahman and Vibhav Gogate. Learning ensembles of cutset networks. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- Dan Roth. On the hardness of approximate reasoning. *Artificial Intelligence*, 82(1-2):273–302, 1996.
- Yujia Shen, Arthur Choi, and Adnan Darwiche. Tractable operations for arithmetic circuits of probabilistic models. In *Advances in Neural Information Processing Systems*, pages 3936–3944, 2016.
- Andy Shih, Guy Van den Broeck, Paul Beame, and Antoine Amarilli. Smoothing structured decomposable circuits. In *Advances in Neural Information Processing Systems*, pages 11412–11422, 2019.
- J Laurie Snell. *Topics in contemporary probability and its applications*, volume 2. CRC Press, 1995.
- Antonio Vergari, YooJung Choi, Robert Peharz, and Guy Van den Broeck. Probabilistic circuits: Representations, inference, learning and applications. Tutorial at the The 34th AAAI Conference on Artificial Intelligence, 2020.
- Tao Zhou, Zoltán Kucsik, Jian-Guo Liu, Matúš Medo, Joseph Rushton Wakeling, and Yi-Cheng Zhang. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences*, 107(10):4511–4515, 2010.