Recursive Bayesian Networks: Generalising and Unifying Probabilistic Context-Free Grammars and Dynamic Bayesian Networks

Robert Lieck[*]
Digital and Cognitive Musicology Lab
École Polytechnique Fédérale de Lausanne
1015 Lausanne, Switzerland
research@robert-lieck.com

Martin Rohrmeier
Digital and Cognitive Musicology Lab
École Polytechnique Fédérale de Lausanne
1015 Lausanne, Switzerland
martin.rohrmeier@epfl.ch

Abstract

Probabilistic context-free grammars (PCFGs) and dynamic Bayesian networks (DBNs) are widely used sequence models with complementary strengths and limitations. While PCFGs allow for nested hierarchical dependencies (tree structures), their latent variables (non-terminal symbols) have to be discrete. In contrast, DBNs allow for continuous latent variables, but the dependencies are strictly sequential (chain structure). Therefore, neither can be applied if the latent variables are assumed to be continuous and *also* to have a nested hierarchical dependency structure. In this paper, we present Recursive Bayesian Networks (RBNs), which generalise and unify PCFGs and DBNs, combining their strengths and containing both as special cases. RBNs define a joint distribution over tree-structured Bayesian networks with discrete or continuous latent variables. The main challenge lies in performing *joint* inference over the exponential number of possible structures and the continuous variables. We provide two solutions: 1) For arbitrary RBNs, we generalise inside and outside probabilities from PCFGs to the mixed discrete-continuous case, which allows for maximum posterior estimates of the continuous latent variables via gradient descent, while marginalising over network structures. 2) For Gaussian RBNs, we additionally derive an analytic approximation of the marginal data likelihood (evidence) and marginal posterior distribution, allowing for robust parameter optimisation and Bayesian inference. The capacity and diverse applications of RBNs are illustrated on two examples: In a quantitative evaluation on synthetic data, we demonstrate and discuss the advantage of RBNs for segmentation and tree induction from noisy sequences, compared to change point detection and hierarchical clustering. In an application to musical data, we approach the unsolved problem of hierarchical music analysis from the raw note level and compare our results to expert annotations.

1 Introduction

Long-term dependencies with a nested hierarchical structure are one of the major challenges in modelling sequential data. This type of dependencies is common in many domains, such as natural

[*] corresponding author, code at https://github.com/robert-lieck/RBN

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

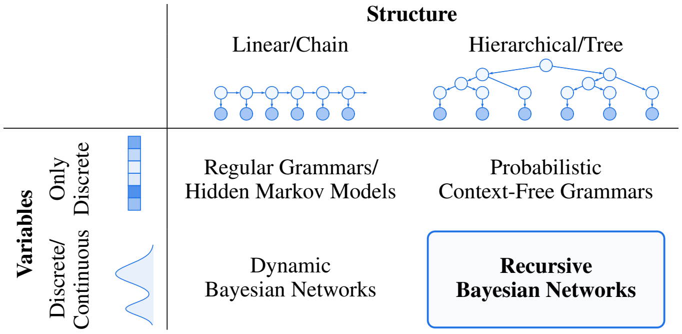
Structure
Linear/Chain
Hierarchical/Tree
Variables
Only Discrete
Discrete/ Continuous
Regular Grammars/ Hidden Markov Models
Probabilistic Context-Free Grammars
Dynamic Bayesian Networks
Recursive Bayesian Networks

Figure 1: RBNs generalise PCFGs by allowing for continuous latent variables and DBNs by incorporating nested hierarchical dependencies.

language [1], music [2, 3], or decision making [4, 5]. Two of the most widely used probabilistic models for sequential data are probabilistic context-free grammars (PCFGs) and dynamic Bayesian networks (DBNs), both having complementary strengths.

PCFGs are well-established and widely used for modelling hierarchical long-term dependencies in symbolic data [1, 6, 7, 8, 9, 5, 10]. They generalise local (Markov) transition models by allowing for infinitely many levels of nested hierarchical dependencies and a flexible number of latent variables. However, parsing methods such as the Cocke-Younger-Kasami (CYK) algorithm [11, 12, 13, 14, 6] rely on the discrete nature of the rules and variables.

In contrast, DBNs are sequential models with a fixed set of random variables that reoccur at each time step [15, 16]. The variables at each time step may be discrete or continuous, latent or observed, and may have an arbitrary non-cyclic dependency structure among each other, with additional links from the previous and to the next time slice. They comprise important model classes as special cases, such as hidden Markov models (HMMs) if there is only a single discrete latent variable or linear dynamical systems if all dependencies are linear Gaussians [17, 18, 16]. However, DBNs only allow for a fixed chain of Markov dependencies between time slices and cannot represent nested hierarchical structures.

In this paper, we present Recursive Bayesian Networks (RBNs), a novel class of probabilistic models that combines the strengths of PCFGs and DBNs by allowing for nested hierarchical dependencies in combination with arbitrary discrete or continuous random variables (Figure 1). Our main contributions are as follows:

1. With RBNs, we provide a unified theoretical framework for a large class of important sequence models, including PCFGs and DBNs.
2. We generalise inside and outside probabilities from PCFGs to continuous latent variables, allowing for maximum posterior (MAP) inference in arbitrary RBNs.
3. For Gaussian RBNs, we derive an analytic approximation for the marginal likelihood and marginal posterior distribution, allowing for robust parameter optimisation and Bayesian inference.
4. We provide a quantitative evaluation on synthetic data and an application to the challenging task of hierarchical music analysis.

1.1 Related Work

PCFGs have a long tradition for modelling nested hierarchical dependencies in symbolic data with a variety of parsing algorithms for inferring the structure and variables' values [13, 6, 14]. Beyond their application to sequential data, PCFGs have been generalised to graph structures [19, 20, 21, 22], which readily transfers to applications of RBNs. Latent vector grammars (LVeGs) [23] are an extension of latent variable grammars (LVGs) [24, 25] with continuous latent states. As for RBNs, approximate parsing is possible in the Gaussian case. However, both LVGs and LVeGs are special cases of RBNs and do not draw the connection to graphical models. More recently, the availability of automatic differentiation libraries, such as `PyTorch` [26], has lead to a number of applications where gradients are propagated through the entire parsing process [27, 28, 29, 30].

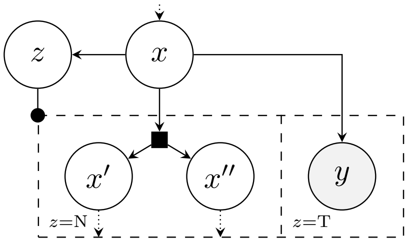

Figure 2: RBN in Chomsky normal form. The non-terminal transition $p_{\mathrm{N}}(x', x'' \mid x)$ and terminal transition $p_{\mathrm{T}}(y \mid x)$ are grouped into an RBN cell with a structural distribution $p_{\mathrm{S}}(z \mid x)$. We use gates [58] to describe structural distributions and extended factor graph notation [black squares; 59] for conditional joint distributions. Considering all possible ways how n observations can be generated by recursively applying the RBN cell produces an RBN chart, as shown in Figure 3.

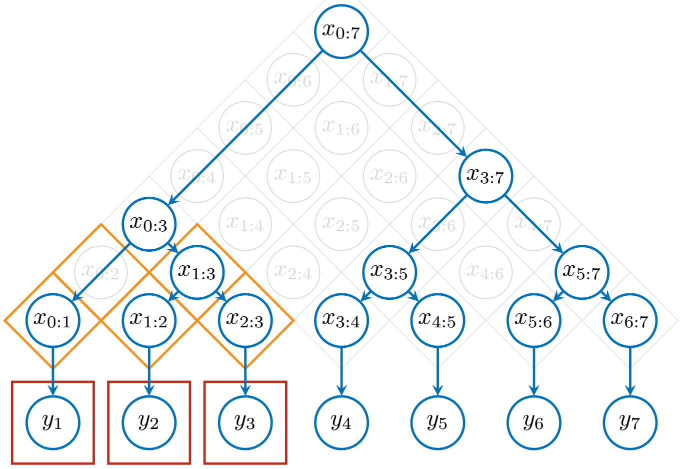

Figure 3: Chart of an RBN in CNF for sequential data of length $n = 7$. The network obtained by fixing one specific dependency structure is highlighted in blue; latent non-terminal variables that are not part of this particular structure are shown in grey. Orange and red boxes, respectively, indicate the subsets $\mathbf{X}_{0:3}$ and $\mathbf{Y}_{0:3}$ of latent non-terminal and observed terminal variables generated from $x_{0:3}$.

The process of parsing a PCFG or RBN can be formally rewritten as a sum-product network [SPN; 31, 32, 33]. Factor graph grammars [FGGs; 34] generalise PCFGs, case-factor diagrams [35] and SPNs by using a hyperedge replacement graph grammar [19] to describe a distribution over graph structures that is more general than that of RBNs (not only trees). However, none of the approaches addresses the problem of inference with continuous variables that we are facing in RBNs (exponentially many terms with exponentially many nested integrals).

A wide range of probabilistic and neural models operate with a fixed graphical structure and are loosely related to RBNs. Hidden tree Markov models [36, 37] generalise HMMs from chain to fixed tree structures. They model data at each node as observations of a latent Markov process on the underlying tree, which is part of the input data. Additionally estimating the underlying tree structure has been addressed in [38, 39, 40]. Recursive neural tensor networks [41] use a PCFG for parsing a given sequence of symbols to obtain a tree structure, which is then fixed and used as the backbone for a neural network. More generally, there is a number of methods for inferring a fixed structure for graphical models [42, 16, 43, 44], SPNs [45, 46, 47, 48, 32, 49], or graph neural networks [50, 51, 52]. All these methods have in common that a fixed structure is either given or estimated but not treated in a probabilistic Bayesian manner.

Some approaches attempt a Bayesian treatment of the unknown structure of a graphical model or SPN via dynamic programming [53, 54] or Markov chain Monte-Carlo sampling [55, 56, 57]. However, the structure is assumed to be independent of the latent variables (they only become dependent *conditional* on the data) and the latent variables cannot be used to *control* the structure, as it is the case in RBNs. The challenge of continuous variables also remains unsolved.

2 Recursive Bayesian Networks

RBNs are *template-based* graphical models that define a joint distribution over network structures and variables' values. The number of template variables is fixed, but the number of instantiated variables, their connectivity and values are governed by the joint distribution. As a rough analogy, RBNs can be thought of as DBNs that can not only be connected linearly to form a chain but also hierarchically to form a tree structure. Alternatively, they can be thought of as a PCFG in which each symbol is a (possibly continuous) random variable.

2.1 Definition

RBNs have three types of template variables: 1) latent non-terminal variables (discrete or continuous), 2) observed terminal variables (discrete or continuous), and 3) latent structural variables (always discrete). In the simplest case, illustrated in Figure 2, an RBN has one template variable of each type. Formally, an RBN is defined as follows:

Definition 1 (Recursive Bayesian Network)**.** *An RBN is a tuple* $(\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{T}, \mathcal{S}, p_{\mathrm{P}})$ *with*

$$\mathcal{X} : \textit{a set of latent non-terminal template variables} \tag{1}$$

$$\mathcal{Y} : \textit{a set of observed terminal template variables} \tag{2}$$

$$\mathcal{Z} : \textit{a set of latent structural template variables, paired up with the non-terminal variables} \tag{3}$$

$$\mathcal{T} : \textit{a set of transition distributions } p(v_1, \ldots, v_\eta \mid x) \textit{ from a single non-terminal variable } x \in \mathcal{X} \textit{ to a set of non-terminal and/or terminal variables } v_1, \ldots, v_\eta \in \mathcal{X} \cup \mathcal{Y} \tag{4}$$

$$\mathcal{S} : \textit{a set of structural distributions } p(z \mid x)\textit{, one for each non-terminal/structural pair} \tag{5}$$

$$p_{\mathrm{P}} : \textit{a prior/start distribution for exactly one non-terminal variable.} \tag{6}$$

The cardinality of a structural variable $z \in \mathcal{Z}$ *corresponds to the number of possible transitions from the associated non-terminal variable* $x \in \mathcal{X}$*;* η *in* (4) *is called the arity of the transition.*

Generating with an RBN is straightforward. We start by sampling the value of the first non-terminal variable x from the prior distribution $p_{\mathrm{P}}(x)$ and then repeat the following steps until no unprocessed non-terminal variables are left:

1. sample the value of the associated structural variable from $p(z \mid x)$
2. choose a transition distribution $p(v_1, \ldots, v_\eta \mid x)$ based on the structural variable's value
3. sample the variables $v_1, \ldots, v_\eta$ from the transition distribution
4. for all newly generated non-terminal variables, go to step 1.

The major challenge and focus of this paper is to perform joint inference over the latent structure and non-terminal variables' values conditional on a given set of observations.

Chomsky Normal Form: In the simplest non-trivial case, an RBN has one latent non-terminal, one observed terminal, and one latent structural template variable, with one non-terminal transition of arity $\eta = 2$ and one terminal transition of arity $\eta = 1$, as illustrated in Figures 2 and 3. It is defined by four distributions

$$p_{\mathrm{P}}(x) : \text{prior/start distribution} \tag{7}$$

$$p_{\mathrm{N}}(x', x'' \mid x) : \text{non-terminal transition} \tag{8}$$

$$p_{\mathrm{T}}(y \mid x) : \text{terminal transition} \tag{9}$$

$$p_{\mathrm{S}}(z \mid x) : \text{termination probability .} \tag{10}$$

In analogy to PCFGs, we call this the Chomsky normal form (CNF). Any RBN may be rewritten in CNF (see Appendix A.1 for details).

RBN Chart: During inference, we will make use of an RBN *chart*, similar to the parse chart for PCFGs [6]. Each non-terminal variable is associated to a layer in the chart. For discrete variables, they store the actual distributions, while for continuous variables they either hold the point estimate (for MAP inference) or the parameters of the approximate distributions (for inference in Gaussian RBN). Different instances of the same template variable are identified by a subscript indicating the span of data generated from them, which also corresponds to their position in the chart (see Figure 3). Sets of variables that are generated from a specific latent non-terminal variable $x_{i:k}$ are denoted by a bold capital letter with a corresponding subscript ($\mathbf{X}_{i:k}$, $\mathbf{Y}_{i:k}$, $\mathbf{Z}_{i:k}$); omitting the subscript refers to *all* variables ($\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$); for $\mathbf{X}$ and $\mathbf{Z}$ this also includes the root variables $x_{0:n}$ and $z_{0:n}$, respectively. The subscripts are to be interpreted as time intervals, that is, $\mathbf{Y}_{i:i}$ is empty, $\mathbf{Y}_{0:1} = y_1$ is the first observation, $\mathbf{Y}_{n\text{-}2:n} = (y_{n\text{-}1}, y_n)$ are the last two observations etc.

Comparison to PCFGs: Any PCFG can be rewritten as an RBN in two different ways, which we call *abstraction* and *expansion* (see Appendix A.2 for details). Abstraction of a PCFG produces a discrete RBN with one latent non-terminal and one observed terminal variable. The resulting RBN is exactly equivalent to the original PCFG but describes the same relations in a more abstract and compact way. In contrast, *expansion* of a PCFG considers the symbols of the grammar as random variables in their own right, thereby endowing them with additional (possibly continuous) degrees of freedom. The resulting RBN is therefore more powerful than the original PCFG. A PCFG is abstracted to a discrete RBN by defining the start/prior, transition, and structural distributions (7–10) as

$$p_{\mathrm{P}}(x{=}A) = \frac{W_{S\to A}}{\sum_{A'} W_{S\to A'}} \tag{11}$$

$$p_{\mathrm{N}}(x'{=}B, x''{=}C \,|\, x{=}A) = \frac{W_{A\to BC}}{\sum_{B',C'} W_{A\to B'C'}} \tag{12}$$

$$p_{\mathrm{T}}(y{=}b \,|\, x{=}A) = \frac{W_{A\to b}}{\sum_{b'} W_{A\to b'}} \tag{13}$$

$$p_{\mathrm{S}}(z \,|\, x{=}A) = \begin{cases} \frac{\sum_{B,C} W_{A\to BC}}{\sum_{X} W_{A\to X}} & \text{if } z{=}\mathrm{N} \\ \frac{\sum_{b} W_{A\to b}}{\sum_{X} W_{A\to X}} & \text{if } z{=}\mathrm{T}\ , \end{cases} \tag{14}$$

where S is the grammar's start symbol, A, B, C are non-terminal symbols, b is a terminal symbol, X is any right-hand side of a rule, $z{=}\mathrm{N}$ and $z{=}\mathrm{T}$ indicate a non-terminal and terminal transition, respectively, $W_{\cdot\to\cdot}$ is the weight of the corresponding rule, and rules that do not exist in the original PCFG are taken to have zero weight. In *expansion*, the PCFG is only used to define a "skeleton" for the RBN, while the specific random variables and the concrete transition distributions need to be additionally specified. This means that the resulting RBN model is more powerful than the original PCFG, as the symbols may, for instance, be expanded to continuous random variables.

2.2 Inference

The two main goals of inference in RBNs are to 1) train model parameters by maximising the marginal data likelihood and to 2) compute posterior distributions or maximum posterior (MAP) estimates of the network structure and non-terminal variables. In PCFGs, both is achieved by computing inside and outside probabilities [14], which will be the starting point for our generalisation to continuous variables.

Inside and Outside Probabilities: We define inside and outside probabilities, β and α, for RBNs in analogy to how they are defined for PCFGs, the only difference being that the variables may be continuous. We thus have

$$\beta(x_{i:k}) := p(\mathbf{Y}_{i:k} \,|\, x_{i:k}) \tag{15}$$

and

$$\alpha(x_{i:k}) := p(\mathbf{Y}_{0:i}, x_{i:k}, \mathbf{Y}_{k:n})\ , \tag{16}$$

where n is the length of the sequence and $\mathbf{Y}$ is fixed (and therefore omitted as argument on the left-hand side). That is, $\beta(x_{i:k})$ is the marginal likelihood of generating the sub-sequence $\mathbf{Y}_{i:k}$ conditional on the respective non-terminal variable $x_{i:k}$, while $\alpha(x_{i:k})$ is the marginal likelihood of generating the two sub-sequences $\mathbf{Y}_{0:i}$ and $\mathbf{Y}_{k:n}$ as well as the non-terminal variable $x_{i:k}$. In both cases, β and α are functions of the corresponding non-terminal variable with the structure and the remaining variables being marginalised out. Based on the inside and outside probabilities, the marginal data likelihood and the marginal posterior distributions over non-terminal variables are

$$p(\mathbf{Y}) = \int \beta(x_{0:n})\, p_{\mathrm{P}}(x_{0:n})\, dx_{0:n} \tag{17}$$

and

$$\widetilde{p}(x_{i:k} \,|\, \mathbf{Y}) = \frac{\alpha(x_{i:k})\, \beta(x_{i:k})}{p(\mathbf{Y})}\ , \tag{18}$$

respectively. $\widetilde{p}(x_{i:k} \,|\, \mathbf{Y})$ is an *unnormalised* probability distribution that specifies the probability of $x_{i:k}$ to exist via the normalisation constant $\int \widetilde{p}(x_{i:k} \,|\, \mathbf{Y})\, dx_{i:k}$, while the normalised version corresponds to the marginal posterior distribution of $x_{i:k}$ for the case that it *does* exist.

Inside probabilities are recursively computed bottom-up. For an RBN in CNF we start with the base case (19) for single observations and then iterate (20) to the top of the RBN chart

$$\beta(x_{i:i+1}) = p_{\mathrm{S}}(z_{i:i+1}{=}\mathrm{T} \,|\, x_{i:i+1})\, p_{\mathrm{T}}(y_{i+1} \,|\, x_{i:i+1}) \tag{19}$$

$$\beta(x_{i:k}) = p_{\mathrm{S}}(z_{i:k}{=}\mathrm{N} \,|\, x_{i:k}) \sum_{j=i+1}^{k-1} \iint p_{\mathrm{N}}(x_{i:j}, x_{j:k} \,|\, x_{i:k})\, \beta(x_{i:j})\, \beta(x_{j:k})\, dx_{i:j}\, dx_{j:k}\ . \tag{20}$$

Outside probabilities are recursively computed top-down, while making use of the inside probabilities

$$\alpha(x_{0:n}) = p_{\mathrm{P}}(x_{0:n}) \tag{21}$$

$$\begin{aligned} \alpha(x_{j:k}) = {} & \Bigg[\sum_{i=0}^{j-1} \iint p_{\mathrm{S}}(z_{i:k}{=}\mathrm{N} \,|\, x_{i:k})\, p_{\mathrm{N}}(x_{i:j}, x_{j:k} \,|\, x_{i:k})\, \alpha(x_{i:k})\, \beta(x_{i:j})\, dx_{i:j}\, dx_{i:k}\Bigg] + \\ & \Bigg[\sum_{l=k+1}^{n} \iint p_{\mathrm{S}}(z_{j:l}{=}\mathrm{N} \,|\, x_{j:l})\, p_{\mathrm{N}}(x_{j:k}, x_{k:l} \,|\, x_{j:l})\, \alpha(x_{j:l})\, \beta(x_{k:l})\, dx_{j:l}\, dx_{k:l}\Bigg]. \end{aligned} \tag{22}$$

As for PCFGs, the two terms in (22) correspond to the possibility of $x_{j:k}$ being generated as the right or the left child, respectively. The main conceptual difference to PCFGs is that we treat the discrete structural part (marginalised out by the sums) separately from the potentially continuous variables (marginalised out by the integrals). For RBNs that are not in CNF, the equations have to be adapted accordingly (see Appendix A.3 for the general case).

Marginalisation: Computing the marginal data likelihood (17) and the marginal posterior distributions over non-terminal variables (18) requires to solve an exponential (w.r.t. the length n of the sequence) number of nested integrals in (19–22), which is generally intractable. However, for the special case of Gaussian RBNs, we provide an adaptive closed-form approximation in Section 2.3. Moreover, marginalising *only* over the network structure for a fixed assignment of the non-terminal variables $\mathbf{X}$ is straight forward and allows for maximum posterior (MAP) inference in general RBNs.

Maximum Posterior Inference: For a fixed assignment of all non-terminal variables $\mathbf{X}$, we can compute the joint marginal likelihood $p(\mathbf{X}, \mathbf{Y})$ over observed terminal and latent non-terminal variables by only marginalising over the structure. This follows the same principle as above but uses the modified *joint* inside and outside probabilities

$$\widehat{\beta}_{i:k} := p(\mathbf{X}_{i:k}, \mathbf{Y}_{i:k} \,|\, x_{i:k}) \qquad (23) \qquad \text{and} \qquad \widehat{\alpha}_{j:k} := p(\mathbf{X}_{0:j}, \mathbf{Y}_{0:j}, x_{j:k}, \mathbf{X}_{k:n}, \mathbf{Y}_{k:n}) \;, \qquad (24)$$

where all variables are fixed (and therefore omitted as arguments on the left-hand side). Analogously, the joint marginal likelihood and the marginal posterior probability of $x_{i:k}$ to exist then are

$$p(\mathbf{X}, \mathbf{Y}) = \widehat{\beta}_{0:n} \, p_{\mathrm{P}}(x_{0:n}) \qquad (25) \qquad \text{and} \qquad \widetilde{p}_{i:k} = \frac{\widehat{\alpha}_{i:k} \, \widehat{\beta}_{i:k}}{p(\mathbf{X}, \mathbf{Y})} \;, \qquad (26)$$

where $\widetilde{p}_{i:k}$ is the probability of $x_{i:k}$ to exist for *this specific* assignment of $\mathbf{X}$. The corresponding equations for the recursion differ from (19–22) only in that they do not integrate out the latent non-terminal variables (see Appendix A.3.3). As before, all computations can be efficiently performed via dynamic programming. Gradients w.r.t. the variables and/or parameters are readily obtained from libraries such as `PyTorch` [26]. Optimising the values of the latent non-terminal variables $\mathbf{X}$ via gradient descent yields maximum posterior (MAP) estimates, while the structure is marginalised out. MAP estimates for the structure (i.e. the best tree) conditional on an assignment for $\mathbf{X}$ can be computed (as for PCFGs) by replacing summation with maximisation [13, 60].

There are two caveats: First, due to marginalising over multiple (exponentially many) network structures, $p(\mathbf{X}, \mathbf{Y})$ may be highly non-convex and optimising $\mathbf{X}$ via gradient descent is not guaranteed to find the global optimum. This is even the case for purely Gaussian RBNs, for which $p(\mathbf{X}, \mathbf{Y})$ is a mixture of Gaussians (one for each structure). Second, we can optimise $\mathbf{X}$ while marginalising out the structure and we can optimise the structure for a fixed assignment of $\mathbf{X}$. However, successively optimising $\mathbf{X}$ and the structure is not equivalent to *jointly* optimising both and the maximum of $p(\mathbf{X}, \mathbf{Y})$ may be unrelated to the maximum of the best structure (also see Figure 4). This means that generally, *exact joint* MAP inference over the latent variables *and* the structure is hard. For Gaussian RBNs, we provide an approximate solution below.

2.3 Gaussian RBNs

In a Gaussian RBN (GRBN), the prior, non-terminal, and terminal distributions are linear Gaussians and the termination probability (structural distribution) is constant

$$\begin{aligned}
p_{\mathrm{P}}(x) &:= \mathcal{N}(x; \mu_{\mathrm{P}}, \Sigma_{\mathrm{P}}) && \text{[prior]} && (27)\\
p_{\mathrm{N}}(x', x'' \,|\, x) &:= \mathcal{N}(x'; x, \Sigma_{\mathrm{NL}})\, \mathcal{N}(x''; x, \Sigma_{\mathrm{NR}}) && \text{[non-terminal]} && (28)\\
p_{\mathrm{T}}(y \,|\, x) &:= \mathcal{N}(y; x, \Sigma_{\mathrm{T}}) && \text{[terminal]} && (29)\\
p_{\mathrm{S}}(z{=}\mathrm{T} \,|\, x) &:= p_{\text{term}} \;. && \text{[termination/structural]} && (30)
\end{aligned}$$

For clarity, we will show all derivations for GRBNs in this basic form. For our evaluations and the application to music, we use a slightly extended version that includes linear transformations, mixtures of Gaussians, and multi-terminal transitions (Section 2.3.1). The derivations do not fundamentally change for the extended case (see Appendix A.4). In Appendix B, we show all calculations on a simple example.

Adaptive Approximation: If the structure of a GRBN was fixed, all variables would be jointly Gaussian distributed as in a conventional Gaussian Bayesian network [16]. However, due to the

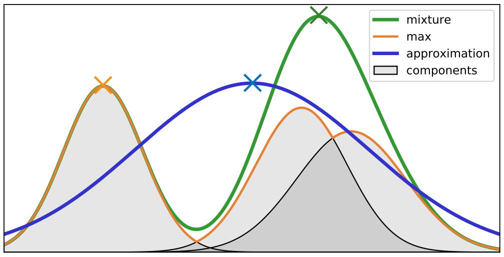

Figure 4: Three Gaussians components, the resulting mixture (green), maximum (orange), and moment-matching single Gaussian approximation (blue). Note that the maximum of the mixture (×), the best component (×), and the approximation (×) may be unrelated.

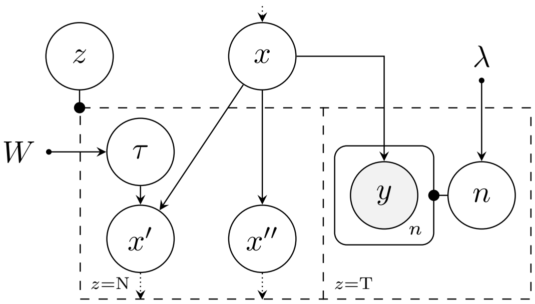

Figure 5: Graphical model of the Gaussian RBN for modelling music. The additional transposition variable τ is marginalised out during inference; the number of jointly generated observations n is uniquely determined by the location in the parse chart.

unknown structure, we effectively have a mixture of exponentially many Gaussians, one for each possible structure. While in principle all integrals can be solved analytically, the exponential growth makes exact inference intractable. Therefore, our goal is to derive a parsing strategy that retains tractability by adaptively applying local approximations to the Gaussian mixtures occurring in each recursion step. We will here focus on the simplest case of approximating the mixtures with a single Gaussian (illustrated in Figure 4, details in Appendix A.4.2), which can be efficiently computed in closed form [18, 61]. The inside and outside probabilities are thus represented by a simple Gaussian

$$\beta(x_{i:k}) \approx c^{(\beta)}_{i:k}\,\mathcal{N}(x_{i:k};\mu^{(\beta)}_{i:k},\Sigma^{(\beta)}_{i:k}) \tag{31}$$

$$\alpha(x_{j:k}) \approx c^{(\alpha)}_{j:k}\,\mathcal{N}(x_{j:k};\mu^{(\alpha)}_{j:k},\Sigma^{(\alpha)}_{j:k}) \tag{32}$$

and this form is reestablished in each iteration by approximating the occurring mixtures. Consequently, the marginal posterior distributions over latent variables (18) are also simple Gaussians and the marginal data likelihood (17) can be computed in closed form. This approximation scheme can be extended and refined by using existing methods for approximating each Gaussian mixture by one with fewer components [62, 63].

Marginalisation: In (20) and (22), we have to integrate over products of Gaussian distributions to marginalise out the latent variables. To solve these integrals, we make use of the fact that the product of two Gaussians over a variable x can be rewritten as [see e.g. 64]

$$\mathcal{N}(x;\mu_1,\Sigma_1)\,\mathcal{N}(x;\mu_2,\Sigma_2) = \mathcal{N}(\mu_1;\mu_2,\Sigma_1+\Sigma_2)\,\mathcal{N}(x;\bar{\mu},\bar{\Sigma}) \tag{33}$$

with

$$\bar{\Sigma} := \left(\Sigma_1^{-1}+\Sigma_2^{-1}\right)^{-1} \qquad \text{and} \qquad \bar{\mu} := \bar{\Sigma}\left(\Sigma_1^{-1}\mu_1+\Sigma_2^{-1}\mu_2\right)\ . \tag{34}$$

Hence, when integrating over x, only the first term on the rhs. of (33) remains. A detailed step-by-step derivation of all results can be found in Appendix A.4.1. With the latent variables being marginalised out, (20) and (22) become simple mixtures of Gaussians that can be easily approximated to retain the simple analytic form of the inside and outside probabilities.

Tree Induction: As described above, exact joint MAP inference over the continuous latent variables and the structure is generally intractable. Moreover, the maximum of the approximate posterior does not necessarily coincide with the maximum of the exact posterior or that of a particular structure (see Figure 4). Thus, first optimising $\mathbf{X}$ (based on the approximation) and then estimating the structure (conditional on the picked value of $\mathbf{X}$) may lead to arbitrarily bad results for tree induction. Therefore, we leverage the adaptive character of our approximation scheme to compute local structure estimates in each step, before loosing relevant information due to further approximations. Specifically, during the bottom-up pass for computing inside probabilities, all structures are scored by the maximum of their marginal likelihood, based on its current approximation (31). The best overall structure is then selected (as usual) in a top-down pass (see Appendix A.4.3 and our example in Appendix B).

2.3.1 Gaussian RBNs for Music

For the application to music, we slightly extend the basic GRBN discussed so far by introducing *transpositions* and *multi-terminal transitions* (changes in the equations highlighted in blue). The corresponding graphical model of the RBN cell is shown in Figure 5. Furthermore, we describe how GRBNs can be applied to *categorical* data.

Transpositions: A transposition rotates the dimensions of the latent variable by a number of steps τ before generating the child. This is achieved by multiplying with an orthonormal transposition matrix T_τ that corresponds to the identity matrix with cyclicly rearranged columns. For the prior distribution, we assume a uniform weighting of all possible transpositions

$$p_{\mathrm{P}}(x) := \sum_{\tau=0}^{D-1} \frac{1}{D} \mathcal{N}(x; T_\tau\, \mu_{\mathrm{p}}, \Sigma_{\mathrm{P}})\ , \qquad \text{[prior]} \tag{35}$$

where D is the dimensionality of the data ($D = 12$ for music in 12-tone equal temperament). For the non-terminal transitions, the probability for a specific transposition is determined by the weight parameter W

$$p_{\mathrm{N}}(x', x'' \,|\, x) := \sum_{\tau=0}^{D-1} p(\tau \,|\, W)\, \mathcal{N}(x'; T_\tau\, x, \Sigma_{\mathrm{NL}}) \mathcal{N}(x''; x, \Sigma_{\mathrm{NR}})\ . \tag{36}$$

Note that transpositions are only applied to the left child, because Western classical music is thought to be fundamentally goal directed [2, 8, 65]. This means that the character of a section is largely determined by how it ends (the right child), which should also be reflected in the value of the parent node. In contrast, the role of the left child is to harmonically prepare the ending (or prepare a preparation to the ending etc). We therefore allow for arbitrary transpositions in the left child and we will see below that our model indeed captures the most important type of preparation in Western classical music: the cadential dominant-tonic progression.

Multi-Terminal Transitions: A multi-terminal transition generates multiple observed variables from a single latent variable. The variables are generated i.i.d. and their number is governed by a Poisson distribution with rate parameter λ

$$p_{\mathrm{T}}(y_{i:k} \,|\, x_{i:k}) := \mathrm{Pois}(k - i - 1 \,|\, \lambda) \prod_{j=i+1}^{k} \mathcal{N}(y_j; x_{i:k}, \Sigma_{\mathrm{T}})\ . \qquad \text{[multi-terminal]} \tag{37}$$

Multi-terminal transitions do not conform to the CNF assumed so far and we need to add the term

$$\beta(x_{i:k}) = \cdots + p_{\mathrm{S}}(z_{i:k}{=}\mathrm{T} \,|\, x_{i:k})\, p_{\mathrm{T}}(y_{i:k} \,|\, x_{i:k}) \tag{38}$$

$$= \cdots + p_{\text{term}}\, p_{\mathrm{T}}(y_{i:k} \,|\, x_{i:k}) \qquad \text{[for GRBNs, see (30)]} \tag{39}$$

to (20) in order to account for the possibility to terminate from a higher-level variable. For $k = i + 1$, this term becomes the base case (19) of an RBN in CNF.

Multi-terminal transitions account for the situation where changes in the hierarchical structure occur at a lower rate than the time series is sampled. In between the structural changes, the data is assumed to be generated from the same model, which could also be more elaborate than i.i.d. samples, as long as the relevant model parameters are captured by the RBN's latent variables.

Categorical Data: The observed variables of a GRBN are unconstrained real-valued, which poses a problem if the data are categorical. This situation is comparable to using Gaussian processes (GPs) [66] for classification and can be approached with similar methods. In our application to musical data, we observe one or more notes being played at any particular time and normalise these counts to obtain observations that correspond to the parameter of a categorical distribution. The natural likelihood function for this type of observations is a Dirichlet distribution. Therefore, we adapt the approach suggested in [67] for GPs, who assume a Dirichlet likelihood, which is then approximated by a Gaussian likelihood in log-space. Since an observation from a Dirichlet distribution corresponds to a normalised sample from independent Gamma distributions, each Gamma distribution can be separately approximated by a log-normal distribution, which results in a diagonal covariance matrix for the Gaussian likelihood in log-space. Matching the first and second moment yields [67]

$$\widetilde{y}_j^{(l)} = \log y_j^{(l)} - \widetilde{\Sigma}_{ll}^{(j)}/2 \qquad \text{and} \qquad \widetilde{\Sigma}_{ll}^{(j)} = \log(1/y_j^{(l)} + 1)\ , \tag{40}$$

where $0 < y_j^{(l)} < 1$ is the l^{th} element (normalised count) of the j^{th} observation, $\widetilde{y}_j^{(l)}$ is the corresponding mean of the approximate Gaussian likelihood in log-space, and $\widetilde{\Sigma}_{ll}^{(j)}$ is the l^{th} element on the diagonal of the covariance matrix for the j^{th} observation. We thus have to replace $\widetilde{y}_j$ and $\widetilde{\Sigma}^{(j)}$ for y_j and Σ_{T} in (37).

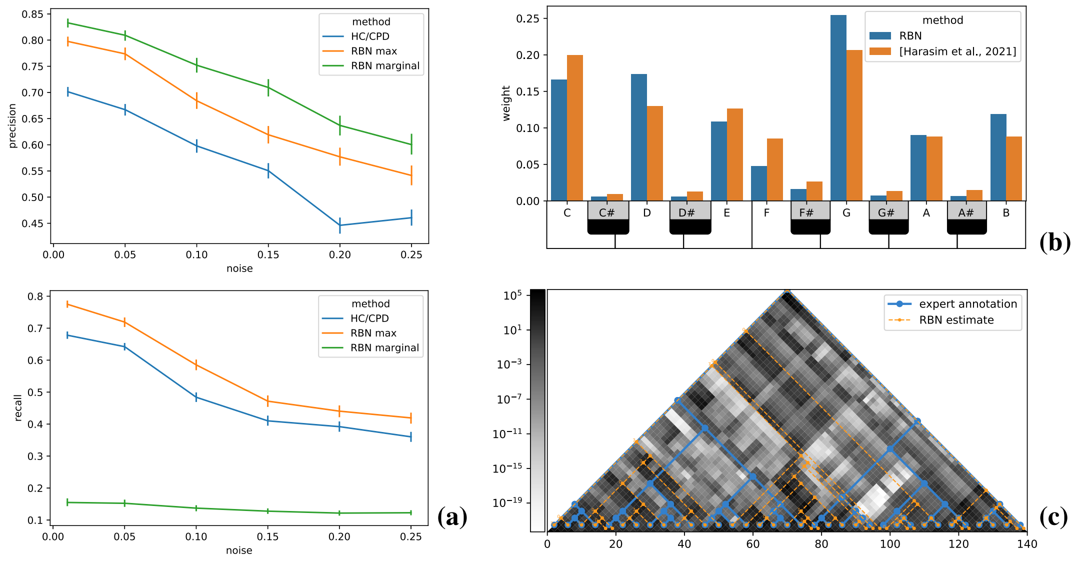

Figure 6: **(a):** Precision and recall w.r.t. the ground-truth trees of 500 sequences for different noise levels for the baseline (blue) and the maximum and marginal RBN estimates (orange, green); error bars indicate 95% confidence intervals from bootstrapping (see Appendix C.1 for technical details). **(b):** Prior mean learned by our GRBN for music in comparison to recent values from the literature [68]. **(c):** Comparison of our model (orange/grey) to an expert annotation (blue) for Johann Sebastian Bach's Prelude No. 1 in C major, BWV 846. The greyscale indicates the marginal probability of a node to exist at that particular location; the small numbers indicate the transposition in semitones for a left child; time is indicated in beats (quarter notes); the piece was divided into two-beat (half note) intervals. The plot follows the idea of *scape plots* [69, 70, 71].

3 Experiments

We performed a quantitative evaluation on synthetic data and applied our model to hierarchical music analysis of Bach preludes. We show that RBNs are superior to change point detection (CPD) and hierarchical clustering (HC) for tree induction and our method is able to infer fundamental harmonic principles of Western classical music. Experiments were run on a 3.6 GHz Quad-Core Intel Core i7 processor with 32GB RAM. The model parameters were trained via gradient descent on the (approximate) marginal neg-log likelihood.

3.1 Quantitative Evaluation on Tree Induction

We performed a quantitative evaluation on synthetic data for the task of segmenting a noisy time series and inferring the underlying tree. For comparison, we used the best-performing change point detection (CPD) method from the `ruptures` library [72] for segmenting the time series, combined with bottom-up hierarchical clustering (HC) for inferring the tree structure ("HC/CPD"). For details of the methodology, see Appendix C.1.

The evaluation results in Figure 6(a) show that the RBN tree estimates (Section 2.3) consistently outperform the one from HC/CPD, in terms of both precision and recall (and thus also in F1 measure). The marginal node probabilities show an interesting performance pattern. They excel in terms of precision, which means that a node with high marginal probability is very likely to actually exist in the tree (low false-positive rate). However, they severely underestimate the overall node probabilities, which leads to recall falling far below the baseline. This means that a node with low marginal probability may in fact occur in the tree (high false-negative rate).

We think that the poor recall measure of the marginal probabilities is primarily due to (and the downside of) a fully Bayesian treatment that quantifies uncertainty. Even if the marginal probabilities have a maximum at the correct node location, probability mass will still spread around it and be allocated to a number of less probable locations. While this is the desired behaviour of a Bayesian method, it inevitably results in a lower recall value. The high precision value confirms that uncertainty is adequately quantified and not underestimated. That being said, the marginal probabilities provide an exceptionally rich basis for qualitative analyses. For instance, all ground-truth nodes are located at local maxima of the marginal probabilities and we can read off a number of other potential node locations, which essentially trace out the grid defined by the piece-wise constant segments (see Figure 8 in Appendix C.1).

3.2 Hierarchical Music Analysis

Harmonies in Western classical music exhibit a nested hierarchical structure that can be modeled by PCFGs operating on abstract chord symbols [8, 73, 10, 3]. While these grammars can be applied to expert annotations of a musical score, hierarchical music analysis from the raw note level is an unsolved problem. We trained a GRBN (Section 2.3.1) on the 24 major preludes of Johann Sebastian Bach's "Wohltemperiertes Klavier I & II" (see Appendix C.2 for technical details and complete results).

Our first major finding is that the prior mean, shown in Figure 6(b), corresponds to a major pitch profile (as could be expected from the training data) and is in excellent agreement with recent Bayesian estimates from the literature [68]. The fact that the major profile appears in the prior (i.e. as the continuous equivalent of a grammar's start symbol) shows that our model picks up fundamentally important structures from the musical data. Our second finding is that only two transpositions have non-zero weights: the identity with a weight of 78% and the fifth scale degree (7 semitones) with a weight of 22%. This corresponds to the left child being generated as the dominant of the parent and realises the most important harmonic preparation in Western classical music: the cadential dominant-tonic relation. A closer inspection of the expert analysis (Figure 10 in Appendix C.2) reveals that when considering the possible surface patterns (raw notes) of the labeled chords, most non-identity transitions can indeed be explained as (noisy) fifth transpositions. The strong weight of fifth transpositions in our model is a highly non-trivial empirical confirmation of the established music theoretical insight that Baroque music is fundamentally driven by dominant-tonic relations. While the estimated tree in Figure 6(c) fails to reproduce the large-scale structure of the expert analysis (e.g. the separation into two main parts), it accurately captures the measure-wise harmonic changes on the bottom level.

On the one hand, we see considerable room for improvement by integrating more advanced concepts, such as different modes (major/minor), diatonic in addition to chromatic transposition, or balancing of trees. On the other hand, our model was able to capture fundamental properties of Western classical music based on only 24 pieces. We therefore think that Gaussian RBNs are a highly promising approach for hierarchical music analysis from the raw note level, which should be further investigated.

4 Conclusion

We introduced Recursive Bayesian Networks (RBNs), a novel class of probabilistic models that unifies the strengths of probabilistic context-free grammars (PCFGs) and dynamic Bayesian networks (DBNs), generalising both model classes. We defined RBNs as a joint distribution over tree-structured Bayesian networks and their (discrete or continuous) variables and described how to perform inference over both the model structure and the variables by leveraging parsing methods for PCFGs. The provided formalisation connects with the methods for formal grammar as well as with the versatile notation for graphical models. On two data sets, we demonstrated the potential of RBNs for modelling nested hierarchical dependencies in real-valued time series and musical data. The class of RBNs represents a substantial contribution to the machine learning toolkit by unifying two of the most important approaches for modelling sequential data and bears a large potential for further development and applications.

Acknowledgments and Disclosure of Funding

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No 760081 – PMSB. This project was conducted at the Latour Chair in Digital and Cognitive Musicology, generously funded by Mr. Claude Latour.

References

[1] Dan Jurafsky. *Speech & Language Processing*. Pearson Education India, 2000.

[2] Fred Lerdahl and Ray Jackendoff. *A Generative Theory of Tonal Music*. MIT press, 1983.

[3] Martin Rohrmeier. The Syntax of Jazz Harmony: Diatonic Tonality, Phrase Structure, and Form. *Music Theory and Analysis (MTA)*, 7(1):1–63, April 2020. doi: 10.11116/MTA.7.1.1.

[4] Andrew G. Barto and Sridhar Mahadevan. Recent Advances in Hierarchical Reinforcement Learning. *Discrete Event Dynamic Systems*, 13(4):341–379, 2003. ISSN 0924-6703.

[5] Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated Planning and Acting*. Cambridge University Press, 2016.

[6] Dick Grune and Ceriel JH Jacobs. Parsing techniques. *Monographs in Computer Science. Springer,*, page 13, 2007.

[7] Christopher W. Geib and Robert P. Goldman. A probabilistic plan recognition algorithm based on plan tree grammars. *Artificial Intelligence*, 173(11):1101–1132, July 2009. ISSN 00043702. doi: 10.1016/j.artint.2009.01.003.

[8] Martin Rohrmeier. Towards a generative syntax of tonal harmony. *Journal of Mathematics and Music*, 5 (1):35–53, March 2011. ISSN 1745-9737, 1745-9745. doi: 10.1080/17459737.2011.573676.

[9] Florent Jacquemard, Pierre Donat-Bouillud, and Jean Bresson. A structural theory of rhythm notation based on tree representations and term rewriting. In *International Conference on Mathematics and Computation in Music*, pages 3–15. Springer, 2015.

[10] Daniel Harasim, Martin Rohrmeier, and Timothy J. O'Donnell. A Generalized Parsing Framework for Generative Models of Harmonic Syntax. In *Proceedings of the 19th International Society for Music Information Retrieval Conference*, pages 152–159, Paris, 2018. doi: 10.5281/zenodo.1492367.

[11] Tadao Kasami. An efficient recognition and syntax-analysis algorithm for context-free languages. *Coordinated Science Laboratory Report no. R-257*, 1966.

[12] Daniel H Younger. Recognition and parsing of context-free languages in time n3. *Information and control*, 10(2):189–208, 1967.

[13] Joshua Goodman. Semiring parsing. *Computational Linguistics*, 25(4):573–605, 1999.

[14] Christopher Manning and Hinrich Schutze. *Foundations of Statistical Natural Language Processing*. MIT press, 1999.

[15] Kevin Patrick Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, University of California, Berkeley, 2002.

[16] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT press, 2009.

[17] Greg Welch and Gary Bishop. An introduction to the Kalman filter. Technical report, University of North Carolina, 1995.

[18] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1st ed. 2006. corr. 2nd printing edition, 2007.

[19] Frank Drewes, H.-J. Kreowski, and Annegret Habel. Hyperedge replacement graph grammars. In *Handbook Of Graph Grammars And Computing By Graph Transformation: Volume 1: Foundations*, pages 95–162. World Scientific, 1997.

[20] Joost Engelfriet and Grzegorz Rozenberg. Node replacement graph grammars. In *Handbook Of Graph Grammars And Computing By Graph Transformation: Volume 1: Foundations*, pages 1–94. World Scientific, 1997.

[21] Eric J. Golin. Parsing visual languages with picture layout grammars. *Journal of Visual Languages & Computing*, 2(4):371–393, December 1991. ISSN 1045-926X. doi: 10.1016/S1045-926X(05)80005-9.

[22] Grzegorz Rozenberg. *Handbook of Graph Grammars and Computing by Graph Transformation*, volume 1. World scientific, 1997.

[23] Yanpeng Zhao, Liwen Zhang, and Kewei Tu. Gaussian Mixture Latent Vector Grammars. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1181–1189, Melbourne, Australia, 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1109.

[24] Richard Socher, John Bauer, Christopher D. Manning, and Andrew Y. Ng. Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 455–465, 2013.

[25] Shay B. Cohen. Latent-Variable PCFGs: Background and Applications. In *Proceedings of the 15th Meeting on the Mathematics of Language*, pages 47–58, London, UK, 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-3405.

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[27] Jason Eisner. Inside-Outside and Forward-Backward Algorithms Are Just Backprop (tutorial paper). In *Proceedings of the Workshop on Structured Prediction for NLP*, pages 1–17, Austin, TX, 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-5901.

[28] Yoon Kim, Chris Dyer, and Alexander Rush. Compound Probabilistic Context-Free Grammars for Grammar Induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2369–2385, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1228.

[29] Yu Zhang, Houquan Zhou, and Zhenghua Li. Fast and accurate neural CRF constituency parsing. In *Proceedings of IJCAI*, pages 4046–4053, 2020. doi: 10.24963/ijcai.2020/560.

[30] Alexander M. Rush. Torch-Struct: Deep Structured Prediction Library. *arXiv:2002.00876 [cs, stat]*, February 2020.

[31] 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, Barcelona, Spain, November 2011. IEEE. ISBN 978-1-4673-0063-6 978-1-4673-0062-9 978-1-4673-0061-2. doi: 10.1109/ICCVW.2011.6130310.

[32] Alejandro Molina, Antonio Vergari, Nicola Di Mauro, Sriraam Natarajan, Floriana Esposito, and Kristian Kersting. Mixed sum-product networks: A deep architecture for hybrid domains. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[33] Xiaoting Shao, Alejandro Molina, Antonio Vergari, Karl Stelzner, Robert Peharz, Thomas Liebig, and Kristian Kersting. Conditional sum-product networks: Imposing structure on deep probabilistic architectures. In *International Conference on Probabilistic Graphical Models*, pages 401–412. PMLR, 2020.

[34] David Chiang and Darcey Riley. Factor Graph Grammars. *Advances in Neural Information Processing Systems*, 33, 2020.

[35] David McAllester, Michael Collins, and Fernando Pereira. Case-factor diagrams for structured probabilistic modeling. *Journal of Computer and System Sciences*, 74(1):84–96, February 2008. doi: 10.1016/j.jcss.2007.04.015.

[36] M. Diligenti, P. Frasconi, and M. Gori. Hidden tree Markov models for document image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(4):519–523, April 2003. ISSN 1939-3539. doi: 10.1109/TPAMI.2003.1190578.

[37] Davide Bacciu, Alessio Micheli, and Alessandro Sperduti. Compositional generative mapping for tree-structured data—Part I: Bottom-up probabilistic modeling of trees. *IEEE transactions on neural networks and learning systems*, 23(12):1987–2002, 2012.

[38] Animashree Anandkumar, Kamalika Chaudhuri, Daniel J Hsu, Sham M Kakade, Le Song, and Tong Zhang. Spectral methods for learning multivariate latent tree structure. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

[39] Myung Jin Choi, Vincent YF Tan, Animashree Anandkumar, and Alan S. Willsky. Learning latent tree graphical models. *Journal of Machine Learning Research*, 12:1771–1812, 2011.

[40] Furong Huang, Niranjan Uma Naresh, Ioakeim Perros, Robert Chen, Jimeng Sun, and Anima Anandkumar. Guaranteed scalable learning of latent tree models. In *Uncertainty in Artificial Intelligence*, pages 883–893. PMLR, 2020.

[41] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, 2013.

[42] David Heckerman, Dan Geiger, and David M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243, 1995.

[43] David Heckerman and Dan Geiger. Learning Bayesian networks: A unification for discrete and Gaussian domains. *arXiv preprint arXiv:1302.4957*, 2013.

[44] Mathias Drton and Marloes H. Maathuis. Structure Learning in Graphical Modeling. *Annual Review of Statistics and Its Application*, 4(1):365–393, March 2017. ISSN 2326-8298, 2326-831X. doi: 10.1146/annurev-statistics-060116-053803.

[45] Robert Gens and Domingos Pedro. Learning the structure of sum-product networks. In *International Conference on Machine Learning*, pages 873–880. PMLR, 2013.

[46] Sang-Woo Lee, Min-Oh Heo, and Byoung-Tak Zhang. Online incremental structure learning of sum–product networks. In *International Conference on Neural Information Processing*, pages 220–227. Springer, 2013.

[47] Amirmohammad Rooshenas and Daniel Lowd. Learning sum-product networks with direct and indirect variable interactions. In *International Conference on Machine Learning*, pages 710–718. PMLR, 2014.

[48] Antonio Vergari, Nicola Di Mauro, and Floriana Esposito. Simplifying, regularizing and strengthening sum-product network structure learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 343–358. Springer, 2015.

[49] Antonio Vergari, Alejandro Molina, Robert Peharz, Zoubin Ghahramani, Kristian Kersting, and Isabel Valera. Automatic Bayesian density analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5207–5215, 2019.

[50] Luca Franceschi, Mathias Niepert, Massimiliano Pontil, and Xiao He. Learning discrete structures for graph neural networks. In *International Conference on Machine Learning*, pages 1972–1982. PMLR, 2019.

[51] Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. Graph structure learning for robust graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 66–74, 2020.

[52] Yue Yu, Jie Chen, Tian Gao, and Mo Yu. Dag-gnn: Dag structure learning with graph neural networks. In *International Conference on Machine Learning*, pages 7154–7163. PMLR, 2019.

[53] Denver Dash and Gregory F. Cooper. Model averaging for prediction with discrete Bayesian networks. *Journal of Machine Learning Research*, 5(Sep):1177–1203, 2004.

[54] Marina Meilă and Tommi Jaakkola. Tractable Bayesian learning of tree belief networks. *Statistics and Computing*, 16(1):77–92, 2006.

[55] Marco Grzegorczyk and Dirk Husmeier. Improving the structure MCMC sampler for Bayesian networks by introducing a new edge reversal move. *Machine Learning*, 71(2-3):265, 2008.

[56] Daniel Eaton and Kevin Murphy. Bayesian structure learning using dynamic programming and MCMC. In *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence (UAI 2007)*, 2007.

[57] Martin Trapp, Robert Peharz, Hong Ge, Franz Pernkopf, and Zoubin Ghahramani. Bayesian Learning of Sum-Product Networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 6344–6355. Curran Associates, Inc., 2019.

[58] Tom Minka and John Winn. Gates. In *Advances in Neural Information Processing Systems*, pages 1073–1080, 2009.

[59] Brendan J. Frey. Extending factor graphs so as to unify directed and undirected graphical models. In Christopher Meek and Uffe Kjærulff, editors, *UAI '03, Proceedings of the 19th Conference in Uncertainty in Artificial Intelligence, Acapulco, Mexico, August 7-10 2003*, pages 257–264. Morgan Kaufmann, 2003. ISBN 0-12-705664-5.

[60] Liang Huang and David Chiang. Better k-best parsing. In *Proceedings of the Ninth International Workshop on Parsing Technology - Parsing '05*, pages 53–64, Vancouver, British Columbia, Canada, 2005. Association for Computational Linguistics. doi: 10.3115/1654494.1654500.

[61] U. Orguner and M. Demırekler. Analysis of single Gaussian approximation of Gaussian mixtures in Bayesian filtering applied to mixed multiple-model estimation. *International Journal of Control*, 80(6): 952–967, June 2007. ISSN 0020-7179. doi: 10.1080/00207170701261952.

[62] Marco F. Huber and Uwe D. Hanebeck. Progressive Gaussian mixture reduction. In *2008 11th International Conference on Information Fusion*, pages 1–8, June 2008.

[63] David F. Crouse, Peter Willett, Krishna Pattipati, and Lennart Svensson. A look at Gaussian mixture reduction algorithms. In *14th International Conference on Information Fusion*, pages 1–8. IEEE, 2011.

[64] Kaare Brandt Petersen, Michael Syskind Pedersen, Jan Larsen, Korbinian Strimmer, Lars Christiansen, Kai Hansen, Liguo He, Loic Thibaut, Miguel Barão, Stephan Hattinger, Vasile Sima, and We The. The matrix cookbook. Technical report, 2006.

[65] Stefan Koelsch, Martin Rohrmeier, R. Torrecuso, and S. Jentschke. Processing of hierarchical syntactic structure in music. *Proceedings of the National Academy of Sciences*, 110(38):15443–15448, September 2013. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1300272110.

[66] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, Cambridge, Massachusetts / London, England, 2006.

[67] Dimitrios Milios, Raffaello Camoriano, Pietro Michiardi, Lorenzo Rosasco, and Maurizio Filippone. Dirichlet-based Gaussian processes for large-scale calibrated classification. In *Advances in Neural Information Processing Systems*, pages 6005–6015, 2018.

[68] Daniel Harasim, Fabian C. Moss, Matthias Ramirez, and Martin Rohrmeier. Exploring the foundations of tonality: Statistical cognitive modeling of modes in the history of Western classical music. *Humanities and Social Sciences Communications*, 8(1):1–11, January 2021. ISSN 2662-9992. doi: 10.1057/s41599-020-00678-6.

[69] Craig Stuart Sapp. Harmonic Visualizations of Tonal Music. In *Proc. International Computer Music Conference (ICMC)*, Havana, Cuba, 2001.

[70] Meinard Müller and Nanzhu Jiang. A Scape Plot Representation for Visualizing Repetitive Structures of Music Recordings. In *ISMIR*, pages 97–102. Citeseer, 2012.

[71] Robert Lieck and Martin Rohrmeier. Modelling Hierarchical Key Structure With Pitch Scapes. In *Proceedings of the 21st International Society for Music Information Retrieval Conference*, pages 811–818, Montréal, Canada, 2020. doi: 10.5281/zenodo.4245558.

[72] Charles Truong, Laurent Oudre, and Nicolas Vayatis. Selective review of offline change point detection methods. *Signal Processing*, 167:107299, February 2020. ISSN 01651684. doi: 10.1016/j.sigpro.2019.107299.

[73] Martin Rohrmeier and Marcus Pearce. Musical Syntax I: Theoretical Perspectives. In Rolf Bader, editor, *Springer Handbook of Systematic Musicology*, pages 473–486. Springer Berlin Heidelberg, Berlin, Heidelberg, 2018. ISBN 978-3-662-55002-1 978-3-662-55004-5. doi: 10.1007/978-3-662-55004-5_25.

[74] Carol L. Krumhansl and Edward J. Kessler. Tracing the dynamic changes in perceived tonal organization in a spatial representation of musical keys. *Psychological review*, 89(4):334, 1982.

[75] David Temperley and Elizabeth West Marvin. Pitch-Class Distribution and the Identification of Key. *Music Perception*, 25(3):193–212, February 2008. ISSN 0730-7829. doi: 10.1525/mp.2008.25.3.193.

[76] Joshua D. Albrecht and David Huron. A Statistical Approach to Tracing the Historical Development of Major and Minor Pitch Distributions, 1400-1750. *Music Perception*, 31(3):223–243, February 2014. ISSN 0730-7829. doi: 10.1525/mp.2014.31.3.223.