Models for music analysis from a Markov logic networks perspective

H Papadopoulos, G Tzanetakis

Models for Music Analysis from a Markov Logic Networks Perspective

H. Papadopoulos and G. Tzanetakis

***Abstract*—Analyzing and formalizing the intricate mechanisms of music is a very challenging goal for Artificial Intelligence. Dealing with real audio recordings requires the ability to handle both uncertainty and complex relational structure at multiple levels of representation. Until now, these two aspects have been generally treated separately, probability being the standard way to represent uncertainty in knowledge, while logical representation being the standard way to represent knowledge and complex relational information. Several approaches attempting a unification of logic and probability have recently been proposed. In particular Markov Logic Networks (MLNs) which combine first-order logic and probabilistic graphical models have attracted increasing attention in recent years in many domains.**

This paper introduces MLNs as a highly flexible and expressive formalism for the analysis of music that encompasses most of the commonly used probabilistic and logic-based models. We first review and discuss existing approaches for music analysis. We then introduce MLNs in the context of music signal processing by providing a deep understanding of how they specifically relate to traditional models, specifically Hidden Markov Models and Conditional Random Fields. We then present a detailed application of MLNs for tonal harmony music analysis that illustrates the potential of this framework for music processing.

***Index Terms*—Statistical Relational Learning, Markov Logic Networks, Hidden Markov Models, Conditional Random Fields, Music Information Retrieval, Tonal Harmony, Chord, Key, Musical Structure**

I. Introduction

THE fascinating task of understanding how human beings create and listen to music has attracted attention throughout history. Nowadays, many research fields have converged to the particular goal of analyzing and formalizing the complex mechanisms of music. The development of computer hardware technology has made possible the development of Artificial Intelligence (AI) techniques for musical research in several directions such as composition, performance, music theory, and digital sound processing. The recent explosion of online audio music collections and the growing demand of listening to music in a personalized way have motivated the development of advanced techniques for interacting with these huge digital music libraries at the song level. Using computers to model human analysis of music and to get insight into the intellectual process of music is a challenge that is faced by many research communities under various names such as Intelligent Audio Analysis [1], Machine Listening [2], or Music Artificial Intelligence.

Part of this research was supported by a Marie Curie International Outgoing Fellowship within the 7th European Community Framework Program.

H. Papadopoulos is with CNRS, Laboratoire des Signaux et Systèmes, 3 rue Joliot-Curie, 91192 Gif-sur-Yvette, France (corresponding author to provide; e-mail papadopoulos@lss.supelec.fr)

George Tzanetakis is with the Computer Science Department, University of Victoria, Victoria, B.C., V8P 5C2, Canada (e-mail gtzan@cs.uvic.ca)

A. Towards a Unified Musical Analysis

In the growing field of Music Information Retrieval (MIR), a fundamental problem is to develop content-based methods to enable or improve multimedia retrieval [3]. The exploration of a large music corpus can be based on several cues such as the audio signal, the score or textual annotations, depending on what the user is looking for. Metadata and textual annotations of the audio content allow for searching based on specific requests such as the title of the piece or the name of the composer. When not looking for a specific request, but more generally for some music pieces that exhibit certain musical properties, search engines are based on annotations that describe the actual music content of the audio, such as the genre, the tempo, the musical key, and the chord progression. Manual annotation of the content of musical pieces is a very difficult, time-consuming and tedious process that requires a huge amount of effort. It is thus essential to develop techniques for automatically extracting musical information from audio.

Although there have been considerable advances in music storage, distribution, indexation and many other directions in the last decades, there are still some bottlenecks for the analysis and extraction of content information. Music audio signals are very complex, both because of the intrinsic nature of audio, and because of the information they convey. Often regarded as an innate human ability, the automatic estimation of music content information proves to be a highly complex task, for at least two reasons.

On the one hand, music signals are extremely rich and complex from a physical point of view, in particular because of the many modes of sound production, of the wide range of possible combinations between acoustic events, and also because signal observations are generally incomplete and noisy. On the other hand, music audio signals are also complex from a semantic point of view: they convey multi-faceted and strongly interrelated information such as harmony, melody, metric, and structure. For instance, chords change more often on strong beats than on other positions of the metrical structure [4].

Recent work has shown that the estimation of musical attributes would benefit from a unified musical analysis that considers the complex relational structure of music as well as the context[1] [5], [6]. Although there is a number of approaches that take into account interrelations between several dimensions in music (e.g. [7]), most existing computational models for music analysis tend to focus on a single music attribute. This is contrary to the human understanding and

[1]For instance, the use of specific instruments can be established based on knowledge of the composition period.

perception of music that is known to process holistically the global musical context [8]. In practice, most existing MIR systems have a relatively simple probabilistic structure and are constrained by limited hypotheses that do not model the underlying complexity of music. Dealing with real audio recordings requires the ability to handle both uncertainty and complex relational structure at multiple levels of representation. Existing approaches for music retrieval tasks typically fail to capture these two aspects simultaneously.

B. Statistical Relational Learning and Markov Logic

Real data such as music signals exhibit both uncertainty and rich relational structure. Until recent years, these two aspects have been generally treated separately, probability being the standard way to represent uncertainty in knowledge, while logical formalisms are the standard way to represent knowledge and complex relational information. Music retrieval tasks would benefit from a unification of both representations. As reflected by previous works, both aspects are important in music, and should be fully considered. However, traditional probabilistic graphical models and machine learning approaches are not able to cope with rich relational structure, while logic-based approaches are not able to cope with the uncertainty of audio and need a transcription step to apply logical inference on a symbolic representation.

Appealing approaches towards a unification between logic and probability have emerged within the field of Statistical Relational Learning (SRL) [9]–[11]. They combine first order logic, relational representations and logical inference, with concepts of probability theory and machine learning [12]. Ideas from probability theory and statistics are used to address uncertainty while tools from logic, databases, and programming languages are introduced to represent structure. Many representations in which statistical and relational knowledge are unified within a single representation formalism have been proposed. They include relational Markov networks [13], probabilistic relational models [14], probabilistic inductive logic programming [15] or Bayesian logic programs [16]. We refer the reader to [11], [12] for a survey of SRL.

Among these approaches, Markov Logic Networks (MLNs) [17]–[19], which combine First-Order Logic (FOL) and probabilistic graphical models (Markov networks) have received considerable attention in recent years. Their popularity is due to their expressiveness and simplicity for compactly representing a wide variety of knowledge and reasoning about data with complex dependencies. Multiple learning and inference algorithms for MLNs have been proposed, for which open-source implementations are available (e.g. the *Alchemy*[2] and *ProbCog* [3] software packages). MLNs have been successfully applied in many domains and used for various tasks in AI, such as collective classification [20] and natural language processing [21].

C. Goal of the Paper

A MLN is a statistical relational learning framework that combines probabilistic inference with first-order logical reasoning. In this paper, we examine the current existing models for music processing and discuss their limitations from an artificial intelligence perspective. We then present MLNs as a highly flexible and expressive formalism for the analysis of music audio signals that encompasses most currently used probabilistic and logic-based models. Our research to date on the use of MLNs for music analysis has shown that they offer a very interesting alternative to the most commonly used hidden Markov models as a more expressive and flexible, yet concise model for content information extraction. We have proposed a single unified MLN model for the joint estimation of chords and global key [22] and we have explored the use of MLNs to integrate structural information to enhance chord estimation [23]. Here, we aim to provide a deeper understanding of the potential of MLNs for music analysis. Very few papers try to explain the deep-seated reasons why MLNs work. To be of real interest to the MIR community, we believe that an understanding of how they specifically relate to commonly used models is needed. To this purpose, we first focus on the theoretical foundations of Hidden Markov Models (HMMs) and Conditional Random Fields (CRFs) and compare the relative capabilities of these models in terms of formalism. This allows us to show how they can be elegantly and flexibly embedded in a more general multilevel architecture with MLNs, which offers new perspectives for music analysis.

Within the music analysis area, we present an application for tonal harmony music analysis [24]. Here tonal harmony analysis is understood as segmenting and labeling an audio signal according to its underlying harmony [25]. In traditional computational models, it is not easy to express dependencies between different semantic and temporal levels. We design in the MLN framework a multi-level harmony description of music, at the beat (chords), bar/phrase (local key, including modulations) and global semantic structure time scales, in which information specific to the various strata interact.

II. Background

Previous work on music content estimation can be classified into two main categories, probabilistic and logic-based models. In the following section, specific emphasis will be given to applications related to tonal harmony analysis.

A. Probabilistic vs. Logic for Music Processing

1) Probabilistic Approaches for Music Content Analysis: Probabilistic graphical models form a large class of *structured prediction models* and are popular for MIR tasks that involve predicting structured objects. In particular hidden Markov models [26] have been quite successful in modeling various tasks where objects can be represented as sequential phenomena, such as chord [27] and local key [28] estimation, beat tracking [29], note segmentation [30] and melody transcription [31]. The objects of interest are modeled as hidden variables that are inferred from some observations. For instance, in a typical chord estimation HMM, the unknown chord progression is inferred from the observation of chroma vectors. An important limitation of HMMs is that it is hard to express dependencies in the data. Strong independence assumptions between the observation variables are made (e.g. each chroma observation is independent from the other etc.). A relevant musical description of audio would ideally consider multiple and typically interdependent observations.

[2] http://alchemy.cs.washington.edu
[3] http://ias.cs.tum.edu/research/probcog

Other formalisms that allow considering more complex dependencies between data have been explored. Variants of HMMs, such as semi-Markov models can better model the duration distributions of the underlying events [32]. N-grams can model long-range chord sequences without making the simplifying Markovian assumption, as in HMM-based approaches, that each chord symbol depends only on the preceding one [33]. The tree structure presented in Paiement *et al.* [7] allows building a graphical probabilistic model where contextual information related to the meter is used to model the chord progression in order to generate chords. Dynamic Bayesian networks (DBN) allow the joint modeling of several musical attributes [5].

However, the use of graphical models that allow more complex dependencies than HMMs for music content estimation remains limited in the MIR field. HMMs belong to the class of Bayesian network models [34] that are used to represent the joint probability distribution $p(y, x)$ between the hidden states y and the observations x, where both x and y are random variables. HMMs are *generative models* in the sense that they describe how the output (the hidden states y) probabilistically generates the input (the observations x), the outputs topologically preceding the inputs. According to Baye's rule, the calculation of the conditional distribution $p(y|x)$ from $p(y, x)$ requires to compute the marginal distribution $p(x)$. This requires enumerating all possible observation sequences, which is difficult when using multiple interdependent input features that result in a complex distribution. This generally leads to intractable models, unless the observation elements are considered as independent from each other. But ignoring these dependencies may impair the performances of the model.

In fact, in all the previously mentioned applications, the observation sequence x is already known and visible in both training and testing datasets. We are only interested in predicting the values of the hidden variables, given the observations. A discriminative framework that directly models the conditional distribution $p(y|x)$ is thus sufficient. In *discriminative models*, the assumptions of conditional independence between the observations and the current state that are posed for the HMMs are relaxed and there is no need to model the prior distribution over the input, $p(x)$. This is in particular the case of Conditional Random Fields (CRFs) [35]. Many works have demonstrated that CRFs overcome several of the limitations of HMMs and offer lot of potential for modeling linear sequence structures. In particular they offer attractive properties in terms of designing flexible observation functions, with multiple interacting features, and modeling complex dependencies or long-range dependencies of the observations.

CRFs have been successfully applied in various fields other than music audio signal processing, including natural language processing, bioinformatics, computer vision and speech processing. There has been recently an increasing interest in using CRFs for modeling music-related tasks, and we review here these works. A tutorial on CRF in the context of MIR research can be found in Essid (2013) [36].

In the context of audio music content estimation, a first attempt to use CRFs is presented in Burgoyne *et al.* (2007) [37] for the purpose of chord progression estimation. Only the property of discriminative learning with few parameters versus generative learning with HMM is exploited. The other possibilities of the framework, such as using richer features or modeling complex dependencies are not considered and the implemented model does not yield to results that outperform a classic HMM. Very recently, CRFs have been applied to beat tracking [38], and to singing voice separation [39].

Other audio tasks that can be seen as a labeling sequential data problem have been modeled in a CRF framework. Audio-to-score alignment has been the most extensive application of CRFs in MIR [40], [41]. It has been shown that existing models for this task can be reformulated with CRF of different structures (semi-Markov CRF, Hidden CRF). The use of CRFs allows designing flexible observation functions that incorporate several features characterizing different aspects of the musical content. The calculation of each state conditional probability is based on audio frames from an arbitrary past or future, improving the matching of a frame with a score position. CRFs have been employed in automatic music transcription [42] in a post-processing step to reduce single-frame errors in a multiple-F0 transcription. They have also been used in the context of audio-tagging [43], and musical emotion evolution prediction [44]. Finally, the ability of CRFs to use multiple dependent features has also been exploited in the symbolic domain, as for symbolic music retrieval [45] and for the automatic generation of lyrics [46].

2) Logic-Based Approaches for Music Content Analysis: A major advantage of the logic framework is that its expressiveness allows modeling music rules in a compact and human-readable way, thus providing an intuitive description of music. For instance, background knowledge, such as music theory, can be introduced to construct rules that reflect the human understanding of music [47]. Another advantage is that logical inference of rules allows taking into account all events including those which are rare [48]. Inductive Logic Programming (ILP) [49] refers to logical inference techniques that are a subset of FOL. These approaches combine logic programming with machine learning. They have been widely used to model and learn music rules, especially in the context of harmony characterization and in the context of expressive music performance. Approaches based on logic have focused on symbolic representations, rather than on audio.

In the context of harmony characterization, pattern-based first-order inductive systems capable of learning new concepts from examples and background knowledge [50], or counterpoint rules for two-voice musical pieces in symbolic format [51] have been proposed. An inductive approach for learning generic rules from a set of popular music harmonization examples to capture common chord patterns is described in [52]. Some ILP-based approaches for the automatic characterization of harmony in symbolic representations [53] and classification of musical genres [54] have been extended to audio [55]. However, they require a transcription step, the harmony characterization being induced from the output of an audio chord transcription algorithm and not directly from audio.

In the context of expressive music performance, algorithms for discovering general rules that can describe fundamental principles of expressive music performance, such as rules

about tempo and expressive timing, dynamics and articulation, have also been proposed [47], [56]–[58]. The inductive logic programming approaches are not directly applied to audio, but on symbolic representations. This generally requires a transcription step, such as melody transcription [47].

B. Tonal and Harmony Music Content Estimation

Since we present an application of MLNs for tonal harmony music analysis, we briefly review in this section existing work on chord, key and structure estimation. The automatic estimation of each of these musical attributes by itself is an important topic of study in the area of content estimation of music audio signals. We review below only works that are directly related to the proposed model. We refer the reader to [59]–[61] for recent reviews on each of these topics.

1) Chord and Local Key Estimation: Harmony together with rhythm are two of the faces of Western tonal music that have been investigated for hundreds of years [62]. Harmony is structured at different time-scales (beat, bar, phrase-level, sections, etc.). Pitches are governed by structural principles and music is organized around one or more stable reference pitches. A *chord* is defined as combination of pitches, and the system of relationships between pitches corresponds to a *key*. A key, as a theoretical concept, implies a tonal center that is the most stable pitch called the tonic, and a mode (usually major or minor). A piece of music generally starts and ends in a particular key referred to as the main or *global key* of the piece. However, it is common that the composer moves between keys. A change between different keys is called a *modulation*. Western tonal music can be conceived of as a progression of a sequence of key regions in which pitches are organized around one stable tonal center. Such a region is defined here as a *local key*, as opposed to the global key. Tonality analysis describes the relationships between the various keys in a piece of music. Tonality and key are complex perceptual attributes, whose perception depends on the listener's level of music training. Moreover, numerous phenoma in a music piece (ambiguous key, apparent tonality, no tonality etc.) contribute to make the problem of local key estimation challenging, and little work has been conducted on this topic (see [28], [60] for more details).

Chords and (local) keys reflect the pitch content of an audio signal at different time-scales. They are intimately related, specific chords indicating a stable tonal center while a given key implying the use of particular chords. Previous works have explored the idea of using chords to find the main key of a musical excerpt [63]–[66]. But the question of how the chord and the key progressions can be jointly modeled and estimated remains scarcely addressed. The few existing works on the topic present serious limitations, as the analysis window size for key estimation is empirically set to a fixed value [67], [68] (resulting in undetected key changes for pieces with a fast tempo and chord rather than key estimation for pieces with a slow tempo), or they do not fully exploit the mutual dependency between chords and keys [28] (the local key is estimated from a fixed chord progression).

2) A Structurally Consistent Description of Muisc: Music structure appears at several time scales, from musical phrases to longer semantically meaningful sections that generally have multiple occurrences (with possible variations) within the same musical piece. Previous works have revealed that the semantic structure can be used as a cue to obtain a "structurally consistent" mid-level representation of music. In the work of Dannenberg [69], music structure is used to constrain a beat tracking program based on the idea that similar segments of music should have corresponding beats and tempo variation. A work more closely related to this article is [70], in which the repetitive structure of songs is used to enhance chord extraction. A chromagram [71], [72] is extracted from the signal, and segments corresponding to a given type of section are replaced by the average of the chromagram over all the instance of the same segment type over the whole song, so that similar structural segments are labeled with the exact same chord progression. A limitation of this work is that it relies on the hypothesis that the chord sequence is the same in all sections of the same type. However, repeated segments are often transformed up to a certain extent and present variations between several occurrences. Moreover, in the case that one segment of the chromagram is blurred (e.g. because of noise or percussive sounds), this will automatically affect all same segments, and thus degrade the chord estimation.

III. MLNs and their relationship to Probabilistic Graphical Models

In this section, we introduce Markov logic networks for music signal processing and clarify the relationship of MLNs to both HMMs and CRFs. As examined in Sec. II, HMMs are the most common models used for music processing. In a labeling context, a HMM can be viewed as a particular case of CRF, which itself is a special case of Markov network. CRFs serve here as a bridge between HMMs and MLNs.

Notations: We will use the following notations. We consider probability distributions over sets of random variables $V = X \cup Y$, where X is a set of input variables that we assume are observed (X is a sequence of observations), and Y is a set of output variables that we wish to predict (the "hidden states"). Every variable takes outcomes from a set $\mathcal{V} = \mathcal{X} \cup \mathcal{Y}$ that can be either continuous or discrete. We focus on the discrete case in this paper. An arbitrary assignment to X is denoted by a vector $x = (x_1, \ldots, x_N)$. Given a variable $X_n \in X$, the notation x_n denotes the value assigned to X_n by x.When there is no ambiguity, we will use the notations $p(y, x)$ and $p(y|x)$ instead of $p(Y = y, X = x)$ and $p(Y = y|X = x)$.

The extraction of music content information can be often seen as a classification problem, in the sense that we wish to assign a class or a label $y \in Y$ (e.g. a chord label) that is not directly observed to an observation $x \in X$ (e.g. a chroma vector). Note that the x are generally fixed observations, rather than treated as random variables. We can approach this problem by specifying a probability distribution to select the most likely class $y \in Y$ we wish to predict for a given observation $x \in X$. In general, the set of variables $X \cup Y$ have a complex structure. A popular approach is to use a *(probabilistic) graphical model* that allows representing the manner in which the variables depend on each other. A graphical model is a family of probability distributions that factorize according to an underlying graph. The idea is to represent

a distribution over a large number of random variables by a product of *potential functions*[4] that each depend on only a smaller subset of variables.

In a probabilistic graphical model, there is a node for each random variable. The absence of an edge between two variables a and b means that they are *conditionally independent* given all other random variables in the model[5]. The concept of conditional independence allows decomposing complex probability distributions into a product of independent factors (see Fig. 1 for an example).

Graphical models include many model families. There are *directed* and *undirected* graphical models, depending on the way the original probability distribution is factorized. Many concepts of the theory of graphical models have been developed independently in different areas and thus have different names. Directed graphical models are also commonly known a *Bayesian networks* and undirected models are also referred to as *Markov random fields* or *Markov networks*.

A. Hidden Markov Models

Hidden Markov models [26], belong to the class of directed graphs, and are standard models for estimating a sequential phenomenon in music. They make strong independence assumptions between the observation variables to reduce complexity. A HMM computes a joint probability $p(y, x)$ between an underlying sequence of N hidden states $y = (y_1, y_2, \ldots, y_N)$ and a sequence of N observations $x = (x_1, x_2 \ldots, x_N)$. A HMM makes two independence assumptions. First, each observation variable x_n is assumed to depend only on the current state y_n. Second it makes the Markov assumption that each state depends only on its immediate predecessor[6]. A HMM is specified using 3 probability distributions:

- The distribution over initial states $p(y_1)$;
- The state transition distribution $p(y_n|y_{n-1})$ to transit from a state y_{n-1} to a state y_n ;
- The observation distribution $p(x_n|y_n)$ of an observation x_n to be generated by a state y_n.

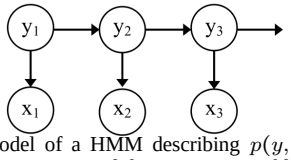

Fig. 1. Graphical model of a HMM describing $p(y, x)$ for a sequence of three input variables x_1, x_2, x_3 and three output variables y_1, y_2, y_3. Because of the conditional independence between variables, the model simplifies in: $p(x_1, x_2, x_3, y_1, y_2, y_3) = p(y_3|y_2) \cdot p(y_3|x_3) \cdot p(y_2|y_1) \cdot p(y_2|x_2) \cdot p(y_1) \cdot p(y_1|x_1)$. Both the observations and the hidden state are random variables and thus represented as unshaded nodes.

The joint probability of a state sequence y and an observation sequence x factorizes as the product of conditional distributions. In this directed graph model, each observation has a "parent label" and the joint probability of the sequence factorizes into pairs of terms, one term corresponding to pairs of labels and a second term corresponding to each observation with its parent label (see Fig 1 for an example):

$$p(y, x) = \prod_{n=1}^{N} p(y_n|y_{n-1}) p(x_n|y_n) \quad (1)$$

where we assume an unconditional prior distribution over the starting state and for time $n = 1$ we write the initial state distribution $p(y_1)$ as $p(y_1|y_0)$.

B. Conditional Random Fields

In many real-word schemes that involve relational data, in particular in music, the entities to be classified are related to each other in complex ways and their labels are not independent. Moreover, any successful classification would need to rely on multiple highly interdependent features that describe the objects of interest. CRFs are generally better suited than HMMs to including rich, overlapping features and thus to represent much additional knowledge in the model[7]. A CRF is a probabilistic model for computing the conditional probability $p(y|x)$ of the output y given the sequence of observations x. A CRF may be viewed as a Markov network globally conditioned on the observations.

A *Markov network* is a model for the joint distribution of a set of variables $V = (V_1, V_2, \ldots, V_n) \in \mathcal{V}$ [75]. It is composed of an undirected graph G and a set of *potential functions* ϕ_k. The graph has a node for each variable and there is a potential function for each clique in the graph[8]. A potential function is a non-negative real-valued function of the state of the corresponding clique. A potential between connected nodes can be viewed as some correlation measure, but it does not have a direct probabilistic interpretation and its value is not restricted to be between 0 and 1. The joint distribution of the variables represented by a Markov network can be factorized over the cliques of the network by:

$$p(V = v) = \frac{1}{Z} \prod_{k} \phi_k(v_{\{k\}}) \quad (2)$$

where, v is an assignment to the random variables V and $v_{\{k\}}$ is the state of the k[th] clique (i.e., the state of the variables that appear in that clique). Z, known as the *partition function*, is given by $Z = \sum_{v \in \mathcal{V}} \prod_k \phi_k(v_{\{k\}})$.

CRFs can be arbitrarily structured (e.g. skip-chain CRFs [76],semi-Markov CRFs [77], segmental CRF [78]). Here, we focus on the canonical linear-chain model introduced in Lafferty *et al.* (2001) [35], that assumes a first-order Markov dependency between the hidden variables y (see Fig. 2).

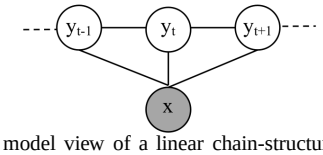

Fig. 2. Graphical model view of a linear chain-structured CRF. The single, large shaded node corresponds to the entire fixed observation sequence x, and the clear nodes correspond to the label variables of the sequence y.

For a linear-chain CRF, the cliques consist of an edge between y_{n-1} and y_n as well as the edges from those two

[4]Also referred to as *factors function* or *local functions* in the literature.

[5]Formally, given a third random variable c, two random variables a and b are *conditionally independent* if and only if $p(a, b|c) = p(a|c)p(b|c)$. Note that in contrast two random variables a and b are *statistically independent* if and only if $p(a, b) = p(a)p(b)$.

[6]This actually corresponds to the standard case of first-order HMMs. Higher-order HMMs called *k-order HMMs* exist where the next state may depend on past k states.

[7]For a discussion about the advantages and disadvantages of CRF vs. HMM, see for instance Murphy (2012) [34], chapter 19. For further reading on CRFs, we recommend the tutorials [73] and [74].

[8]In an undirected graph, a *clique* is a set of nodes Ω forming a fully connected subgraph: for every two nodes in Ω, there exists an edge connecting the two.

labels to the set of observations x (see Fig. 2). The probability of a particular label sequence y given observation sequence x can be factorized into a normalized product of strictly positive, real-valued potential functions, globally normalized over the entire sequence structure:

$$p(y|x) = \frac{1}{Z(x)} \prod_{n=1}^{N} F_n(x,y) \tag{3}$$

The normalization factor $Z(x)$ sums over all possible state sequences so that the distribution sums to 1. $F_n(x,y)$ is a set of feature functions designed to capture useful domain information that have the form $F_n(x,y) = \exp(\sum_{k=1}^{K} \lambda_k f_k(y_{n-1}, y_n, x))$, where f_k are real-valued functions of the state, and λ_k is a set of weights. Eq. (3) writes:

$$p(y|x) = \frac{1}{Z(x)} \prod_{n=1}^{N} \exp\Big(\sum_{k=1}^{K} \lambda_k f_k(y_{n-1}, y_n, x) \Big) \tag{4}$$

where $Z(x) = \sum_{y} \prod_{n=1}^{N} \exp\Big(\sum_{k=1}^{K} \lambda_k f_k(y_{n-1}, y_n, x) \Big)$.

In contrast to HMMs, the feature functions of CRFs can not only depend on the current observation but on observations from an arbitrary past or future for the calculation of each state probability. Feature functions can belong to any family of real-valued functions, but in general they are binary functions, and we will focus on this case here.

We also write the model in the case where the observations are restricted to a single frame x_n, for convenience of future comparison to HMMs (see Fig. 3):

$$\begin{aligned} p(y|x) &= \frac{1}{Z(x)} \prod_{n=1}^{N} \exp\Big(\sum_{k=1}^{K} \lambda_k f_k(y_{n-1}, y_n, x_n) \Big) \\ &= \frac{1}{Z(x)} \exp\Big(\sum_{n=1}^{N} \sum_{k=1}^{K} \lambda_k f_k(y_{n-1}, y_n, x_n) \Big) \end{aligned} \tag{5}$$

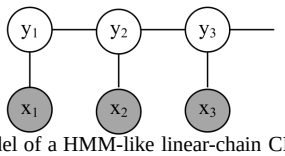

Fig. 3. Graphical model of a HMM-like linear-chain CRF describing $p(y|x)$. Here, it is an undirected graphical model: compared to the HMM in Fig. 1, the arrowheads of the edges have disappeared. The shaded nodes indicate that the corresponding variables are observed and not generated by the model.

C. HMM vs. CRF

The joint distribution $p(y,x)$ of a HMM can be viewed as a CRF with a particular choice of feature functions. We now describe how it is possible to translate a HMM into the feature functions and weights of a CRF.

For each state transition pair $(i,j), i,j \in S$ (where S represents a set of hidden states) and each state-observation pair $(i,o), i \in S, o \in O$ (where O represents the set of the observations), let define a binary feature function of the form:

$$f_{i,j}^{\text{trans}}(y_{n-1}, y_n, x_n) = \mathbb{1}(y_{n-1} = i) \cdot \mathbb{1}(y_n = j)$$

$$f_{i,o}^{\text{obs}}(y_{n-1}, y_n, x_n) = \mathbb{1}(y_n = i) \cdot \mathbb{1}(x_n = o)$$

where $\mathbb{1}(x = i)$ denotes an indicator function of x that takes the value 1 when $x = i$ and 0 otherwise. In other words, $f_{i,j}^{\text{trans}}(y_{n-1}, y_n, x_n)$ returns 1 when $y_{n-1} = i$ and $y_n = j$, and 0 otherwise.

Let also define the set of weights $w_{i,j}^{\text{trans}} = \log p(y_n = j | y_{n-1} = i)$. We have:

$$\sum_{i,j \in S} w_{i,j}^{\text{trans}} \cdot f_{i,j}^{\text{trans}}(y_{n-1}, y_n, x_n) = \log p(y_n = j | y_{n-1} = i)$$

Similar equations are obtained with $w_{i,o}^{\text{obs}} = \log p(x_n = o | y_n = i)$[9].We can then rewrite Eq. (1) as:

$$\begin{aligned} p(y,x) &= \prod_{n=1}^{N} p(y_n|y_{n-1}) p(x_n|y_n) \\ &= \prod_{n=1}^{N} \exp\Big(\log p(y_n|y_{n-1}) + \log p(x_n|y_n) \Big) \\ &= \prod_{n=1}^{N} \exp\Big(\sum_{i,j \in S} w_{i,j}^{\text{trans}} \cdot f_{i,j}^{\text{trans}}(y_{n-1}, y_n, x_n) \\ &\qquad + \sum_{i \in S} \sum_{o \in O} w_{i,o}^{\text{obs}} \cdot f_{i,o}^{\text{obs}}(y_{n-1}, y_n, x_n) \Big) \end{aligned} \tag{7}$$

We refer to a feature function generically as f_k, where f_k ranges over both all of the $f_{i,j}^{\text{trans}}$ and all of the $f_{i,o}^{\text{obs}}$, and similarly refer to a weight generically as w_k. The previous equation writes:

$$p(y,x) = \prod_{n=1}^{N} \exp\Big(\sum_{k=1}^{K} w_k \cdot f_k(y_{n-1}, y_n, x_n) \Big)$$

From the definition of conditional probability, we have:

$$p(y|x) = \frac{p(y,x)}{p(x)} = \frac{p(y,x)}{\sum_{y} p(y,x)}$$

Denoting $Z(x) = \sum_{y} p(y,x)$, we finally obtain:

$$\begin{aligned} p(y|x) &= \frac{1}{Z(x)} \prod_{n=1}^{N} \exp\Big(\sum_{k=1}^{K} w_k \cdot f_k(y_{n-1}, y_n, x_n) \Big) \\ &= \frac{1}{Z(x)} \exp\Big(\sum_{n=1}^{N} \sum_{k=1}^{K} w_k \cdot f_k(y_{n-1}, y_n, x_n) \Big) \end{aligned} \tag{10}$$

Eq. (10) defines the same family of distributions as Eq. (5). For a labeling task, a HMM is thus a particular case of linear-chain CRF for a suitable choice of clique potentials, where each potential feature is either a state feature function or a transition feature function. In practice, the main difference between using HMMs and HMM-like CRFs lies in the way the model parameters are learnt. In the case of HMMs they are learned by maximizing the joint probability distribution $p(x,y)$, while in the case of CRF they are learned by maximizing the conditional probability distribution $p(y|x)$, which avoids modeling the observations distribution $p(x)$.

For convenience (see Sec. IV-B3), we also write the conditional probability of the CRF in its direct translation from a HMM from Eq. (7) as a product of factors:

$$p(y|x) = \frac{1}{Z(x)} \prod_{n=1}^{N} \Phi_{\text{trans}}(y_{n-1}, y_n, x_n) \Phi_{\text{obs}}(y_n, x_n) \tag{11}$$

where $\Phi_{\text{trans}}(y_{n-1}, y_n, x_n)$ and $\Phi_{\text{obs}}(y_n, x_n)$ are exponential family potential functions, respectively over state and observation configurations, that are derived from the transition and observation probabilities of the HMM. Here we have removed the unused variable y_{n-1} in the state-observation pairs.

D. Markov Logic Networks

1) MLN Intuitive Idea: MLNs have been introduced by Richardson & Domingos [17] and are a combination of

[9]For state-observation pairs, the variable y_{n-1} could be removed but we keep it to stay consistent with the definition of linear-chain CRF (Eq. (3)).

Markov networks and first-order logic (FOL). A MLN is a set of weighted FOL formulas[10], that can be seen as a template for the construction of probabilistic graphical models. We present in this section a short overview of the main concepts of Markov logic, with specific examples from the modeling of musical concepts. We refer the reader to Domingos & Lowd (2009) [19] for a thorough review.

MLNs are meant to be intuitive representations of real-world scenarios. In general, FOL formulas are first used to express knowledge. Then a Markov network is constructed from the instantiation of these formulas. The knowledge base is transformed into a probabilistic model simply by assigning weights to the formulas, manually or by learning them from data. Inference is then performed on the Markov network.

2) Definitions and Vocabulary: A *Markov network*, as presented in Sec. III-B, is a model for the joint distribution of a set of variables $V = (V_1, V_2, \dots, V_n) \in \mathcal{V}$, often represented as a log-linear model with each clique potential replaced by an exponentiated weighted sum of features of the state:

$$p(V = v) = \frac{1}{Z} \exp\Big(\sum_j w_j f_j(v) \Big) \tag{12}$$

where Z is a normalization factor, and $f_j(v)$ are features of the state v. A feature may be any real-valued function of the state, but here (and in the literature of Markov logic)[11], we focus on binary features, $f_j(v) \in \{0, 1\}$. The most direct translation from the potential function form Eq. (2) to the log-linear form Eq. (12) is obtained with one feature corresponding to each possible state $v_{\{k\}}$ of each clique, with its weight being $\log\Big(\phi_k(v_{\{k\}})\Big)$.

In first-order logic, the domain of discourse is defined by a set of four types of symbols. *Constants* (e.g. `CMchord` ("C major chord"), `GMchord`) represent objects in the domain; the set of constants is here assumed finite[12]. Variables (e.g. `x`,`y`) take objects in the domain as values. *Predicates* represent properties of objects (e.g. `IsMajor(x)`, `IsHappyMood(x)`) and relations between them (e.g. `AreNeighbors(x,y)`). *Functions* represent mappings from tuples of objects to objects.

A predicate can be *grounded* by replacing its variables with constants (e.g. `IsMajor(CMchord)`, `AreNeighbors(CMchord,GMchord)`). A predicate takes as outputs either True (synonymous with 1) or False (synonymous with 0). A *ground predicate* is called an *atomic formula* or an *atom*. A *positive literal* is an atomic formula and a *negative literal* is the negation of an atomic formula. A *world* is an assignment of a truth value (0 or 1) to each possible ground predicate.

A *first-order knowledge base* (KB) is a set of formulas in first-order logic, constructed from predicates using logical connectives ($\Rightarrow$: "if... then" (implication); $\Leftrightarrow$: "if and only if" (equivalence); $\wedge$: "and" (conjunction); $\vee$: "or" (disjunction); $\neg$: (negation)) and quantifiers (the universals $\forall$: "for all"; $\exists$: "there exists").

In general, in Markov logic implementations, formulas are converted to *clausal form*, also known as *conjunctive normal form* (CNF) for automated inference. Every KB in FOL can be converted to clausal form [82]. A clausal form is a conjunction of clauses, a *clause* being a disjunction of literals.

3) MLN formal definition: A first-order KB can be seen as a set of hard constraints on the set of possible worlds: if a world violates even one formula, it has zero probability. In a real world scheme, logic formulas are *generally*, but not *always* true. The basic idea in Markov logic is to soften these constraints to handle uncertainty: when a world violates one formula in the KB, it is less probable than one that does not violate any formulas, but not impossible. Markov logic allows contradictions between formulas by weighting the evidence on both sides. The weight associated with each formula reflects how strong a constraint is, i.e. how unlikely a world is in which that formula is violated. The more formula a possible world satisfies, the more likely it is. Tab. I shows a KB and its conversion to clausal form, with corresponding weights in the MLN.

Definition 1 *Formally, a* Markov logic network L *[17] is defined as a set of pairs* (F_i, w_i)*, where* F_i *is a formula in first-order logic and* w_i *is a real number associated with the formula. Applied to a finite set of constants* C *(to which the predicates appearing in the formulas can be applied), it defines a ground Markov network* $M_{L,C}$ *as follows:*

1) $M_{L,C}$ *contains one binary node for each possible grounding of each predicate (i.e. each atom) appearing in* L*. The value of the node is 1 if the ground predicate is true, and 0 otherwise.*

2) $M_{L,C}$ *contains one feature* f_j *for each possible grounding of each formula* F_i *in* L*. The feature value is 1 if the ground formula is true, and 0 otherwise. The weight* w_j *of the feature is the weight* w_i *associated with the formula* F_i *in* L*.*

A MLN can be viewed as a *template* for constructing Markov networks: given different set of constants, it will produce different networks. Each of these networks is called a *ground Markov network*. A ground Markov network $M_{L,C}$ specifies a joint probability distribution over the set $\mathcal{V}$ of possible worlds, i.e. the set of possible assignments of truth values to each of the ground atoms in V[13]. From Def. (1) and Eq. (12), the joint distribution of a possible world V given by $M_{L,C}$ is:

$$p(V = v) = \frac{1}{Z} \exp\Big(\sum_i w_i n_i(v) \Big) \tag{13}$$

where the sum is over indices of MLN formulas and $n_i(v)$ is the number of true groundings of formula F_i in v (i.e. $n_i(v)$ is the number of times the i^{th} formula is satisfied by possible world V), and $Z = \sum_{v' \in \mathcal{V}} \exp\Big(\sum_i w_i n_i(v') \Big)$.

[10]First-order logic is also known as predicate logic because it uses predicates and quantifiers, as opposed to propositional logic that deals with simple declarative propositions and is less expressive. The adjective "first-order" distinguishes first-order logic, in which quantification is applied only to variables, from higher-order logic in which quantification can be applied to predicate and function symbols. For more details, see e.g. [79], [80].

[11]Also as in the case of CRFs presented in Sec. III-B.

[12]Markov logic have originally been defined only for finite domains [17] but have since been extended to infinite domains [81]. In this paper we are only concerned with finite domains.

[13]The ground Markov network consists of one binary node for each possible grounding of each predicate. A world $V \in \mathcal{V}$ is a particular assignment of truth value (0 or 1) to each of these ground predicates. If $|V|$ is the number of nodes in the network, there are $2^{|V|}$ possible worlds.

TABLE I
EXAMPLE OF A FIRST-ORDER KB AND CORRESPONDING WEIGHTS IN THE MLN.

Knowledge	*First-order Logic formula*	*Clausal Form*	*Weight*
A major chord implies a happy mood.	∀x IsMajor(x) ⇒ IsHappyMood(x)	¬IsMajor(x) ∨ IsHappyMood(x)	$w_1 = 0.5$
If 2 chords are neighbors on the circle of fifths, either both are major chords or neither are.	∀x ∀y AreNeighbors(x, y) ⇒ (IsMajor(x) ⇔ IsMajor(y))	¬AreNeighbors(x, y) ∨ IsMajor(x) ∨ ¬ IsMajor(y), ¬AreNeighbors(x, y) ∨ ¬ IsMajor(x) ∨ IsMajor(y)	$w_2 = 1.1$ $w_2 = 1.1$

Assumptions in practical applications: To ensure that the number of possible worlds for $M_{L,C}$ is finite, and that the MLN will give a well-defined probability distribution over those worlds, three assumptions about the logical representation are typically made: different constants refer to different objects (unique names), the only objects in the domain are those representable using the constant and function symbols (domain closure), and the value of each function for each tuple of object is always a known constant (known functions). For more details, see [17].

Remark: MLNs are usually defined as log-linear models. However, Eq. (13) can be rewritten as a product of potential functions:

$$p(V = v) = \frac{1}{Z}\exp\Big(\sum_i w_i n_i(v)\Big) = \frac{1}{Z}\prod_i \phi_i(v_{\{i\}})^{n_i(v)} \quad (14)$$

with $\phi_i(v_{\{i\}}) = \mathrm{e}^{w_i}$. This shows that any discrete probabilistic model expressible as products of potentials can be expressed with a MLN. This includes Markov and Bayesian networks.

4) Example: Fig. 4 shows the graph of the ground Markov network defined by the two formulas in Tab. I and the constants `CMchord` (`CM`) and `GMchord` (`GM`). The grounding process is illustrated in Fig. 5. There are 3 predicates and 2 constants. They result in 8 nodes that are binary random variables denoted by V, and that each represent a grounded atom.

The graphical structure of $M_{L,C}$ follows from Def. (1):

- Each possible grounding of each predicate in F_i becomes a node in the Markov network. Each node has a binary value: 1("True") or 0 ("False").
- Each possible grounding of each formula becomes a feature in the Markov network.
- All nodes whose corresponding predicates appear in the same formula form a clique in the Markov network. Each clique is associated with a feature.

The Markov network grows as the number of constants and formula groundings increases, but the number of the formulas (or the templates) stays the same.

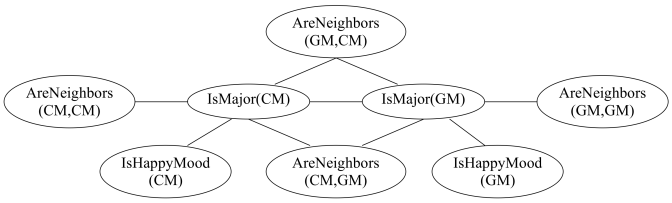

Fig. 4. Ground Markov network obtained by applying the formulas in Tab. I to the constants `CMchord` (`CM`) and `GMchord` (`GM`).

For the Markov network in Fig. 4, a world is an assignment of a truth value to each possible ground predicate in V = (`IsMajor(CM)`,`IsMajor(GM)`,`IsHappyMood(CM)`,`IsHappyMood(GM)`,`AreNeighbors(CM,CM)`,`AreNeighbors(CM,GM)`,`AreNeighbors(GM,CM)`,`AreNeighbors(GM,GM)`). Some elements in V may correspond to the same template formula F_i with different truth assignments, and $n_i(x)$ only counts the assignments which make F_i true. For instance, there are two groundings for formula: ∀ `x IsMajor(x)` ⇒ `IsHappyMood(x)`. For $v = (1,1,1,0,1,1,1,1)$ where 1 is true and 0 is false, $n_1(x) = 1$ because only `IsMajor(CM)`⇒ `IsHappyMood(CM)` gives true value while `IsMajor(GM)` ⇒ `IsHappyMood(GM)` does not. For detailed examples of the computation of joint distribution of a possible world V from Eq. (13) in Markov logic, we refer the reader to Cheng *et al.* (2014) [84].

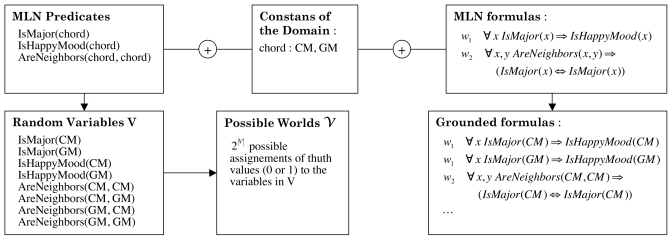

Fig. 5. Illustration of the grounding process of the ground Markov network in Fig. 4. Adapted from [83].

E. MLNs vs. HMM and CRF

In a labeling task context, we know *a priori* which predicates are evidence and which ones will be queried. The ground atoms in the domain can be partitioned into a set of evidence atoms (observations) x and a set of query atoms y. The conditional probability distribution of y given x is [18]:

$$p(y|x) = \frac{1}{Z(x)}\exp\Big(\sum_{i\in F_Y} w_i n_i(x,y)\Big) \quad (15)$$

where F_Y is the set of all MLN clauses with at least one grounding involving a query atom and $n_i(x,y)$ is the number of true groundings of the i^{th} clause involving query atoms.

This can also be written:

$$p(y|x) = \frac{1}{Z(x)}\exp\Big(\sum_{i\in G_Y} w_i g_i(x,y)\Big) \quad (16)$$

where G_Y is the set of ground clauses in $M_{L,C}$ involving query atoms, and $g_i(x,y) = 1$ if the i^{th} ground clause is true in the data and 0 otherwise.

Comparing Eq. (16) and Eq. (10), we can see that for a labeling task, a HMM can be expressed in Markov logic by producing a clause for each state-transition pair $(i,j), i,j \in S$ and each state-observation pair $(i,o), i \in S, o \in O$, and giving a weigh $w_{i,j} = \log p(y_n = j|y_{n-1} = i)$ and $w_{i,o} = \log p(x_n = o|y_n = i)$ respectively. Eqs. (16) and (5) show that a linear-chain CRF can be expressed in Markov logic by producing a clause corresponding to each feature function of the CRF, with the same weight as the CRF feature. These graphical models can be specified very compactly in Markov logic using a few generic formulas (see Section IV-C1).

F. MLNs, First-order Logic & Probabilistic Graphical Models

Markov Logic generalizes both first-order logic and most commonly-used statistical models. FOL is the special case of MLNs obtained when all weights are equal and tend to infinity. In addition to add flexibility in knowledge bases using weights, Markov logic allows contradictions between formulas. Other interesting features include building a MLN by merging several KBs, even if they are partly incompatible [19].

From a probabilistic point of view, Markov logic allows very complex models to be represented very compactly (e.g. only three formulas for a hidden Markov model). Any discrete probabilistic models expressible as products of potentials can be expressed with a Markov logic network. MLNs thus generalize most commonly-used probabilistic graphical models, which includes Markov networks and Bayesian networks[14]. They also facilitate the incorporation of rich domain knowledge that can be combined with purely empirical learning, and allows reasoning with incomplete data [86], [87].

Several implementations of Markov logic exist that comprise a series of efficient algorithms for inference, as well as weight and structure learning. In practical applications, MLNs distributions are typically unique in the sense that they represent a large number of variables that have implicit/explicit dependencies between each other. In this context, algorithms that combine probabilistic methods with ideas from logical inference have been developed. This is out of the scope of this article, but we refer the reader to Domingos & Lowd (2009) [19], Chapters 3 and 4 for a detailed description of these algorithms, and also to [88]–[90] for more recent reviews.

IV. Application: MLN Multi-scale Tonal Harmony Analysis

In this section, we instantiate Markov logic networks for music signal processing, within the context of tonal harmony analysis, and show how they compare with probabilistic graphical model commonly used in MIR. Starting from a classic HMM for chord progression estimation, we then propose a new chord estimation model based on CRF that integrates richer features. We show how these models can be translated into Markov logic that offers further flexibility in terms of modeling the complex relational structure of music. We finally present a MLN that is able to model complex tonal and harmony relational structure at several time scales.

A. Generalities

1) Music Theory Foundations and Hypothesis: Musical elements are highly organized. At the highest level, when listening to a piece of music, we can feel in general a structure and divide the piece into several segments that are semantically meaningful, as for instance verse or chorus sections in a Western popular music song. These segments are in general related to the metrical structure, which itself is a hierarchical structure. The most salient metrical level, called the *beat* level corresponds to the foot-tapping rate. Beats are aggregated in larger time units called *measures* or *bars*.

In the time space, chords and local keys can be viewed respectively as local and more global elements of tonal harmony. In this paper, the chord and (local) key progressions are estimated using a restricted chord lexicon composed of $I = 24$ major (M) and minor (m) triads (CM, ..., BM, Cm, ..., Bm), and considering 24 possible keys (CM key, ..., BM key, Cm key, ..., Bm key) based on the major and harmonic minor scales and 12 pitches that compose an octave range of Western music. We will also reasonably assume that within a given measure, there cannot be any key modulation.

As mentioned in Sec. II-B, chords, keys and the semantic structure are highly interrelated. We propose a model for tonal harmony analysis of audio that takes into account (part of) this complex relational structure. Our model allows a joint estimation of local keys and chords. Moreover, following the idea of designing a "structurally consistent" mid-level representation of music, we show that the MLN framework allows incorporating prior structural information to enhance chord and key estimation in an elegant and flexible way.

Although a long-term goal is to develop a fully automatic model that integrates an automatic segmentation, we follow the previous approach for "structurally consistent" analysis [70] and assume that the metrical and semantic structures are known. The segmentation of the song in beats, downbeats and structure is given as prior information.

2) Signal Processing Front-end: The front-end of all the models described in this section is based on the extraction of chroma feature vectors that describe the signal. The chroma vectors are 12-dimensional vectors that represent the intensity of the twelve semitones of the Western tonal music scale, regardless of octave. We perform a *beat synchronous* analysis and compute one chroma vector per beat[15].

3) About the Model Parameters: In what follows, the parameters of the models are derived from expert knowledge on music theory. All considered models allow training but we left this aspect for future work. In Markov logic, the weights can be learned either generatively or discriminatively. We refer the reader interested in MLNs learning to [19], [92], [93].

Note that in the more general case, the weights of a MLN have no obvious probabilistic interpretation since they are interpreted relative to each other when defining the joint probability function. The weight of a clause specifies the probability of a world in which this clause is satisfied *relative* to a world in which it is not satisfied. According to the heuristic discussed in [17], the weight of a formula F is the log odds between a world where F is true and a world where F is false, other things being equal. However, if F shares variables with other formulas (as it is typically the case) this correspondence does not hold, as the weight of F is influenced not only by its probability, but also by the other formulas that share the same variable. We refer the reader to [83], [94], [95] for more details.

[14]MLNs can also be applied to time-changing domains: dynamic Bayesian networks can be equivalently modeled with MLNs [85]. Approaches such as [5] for music could thus be modeled by a MLN, then possibly enriched, e.g. with longer-term dependencies, as in the application presented in Sec. IV-C3.

[15]This is done by integrating a beat-tracker as a front-end of the system [91]. As a matter of fact we consider half-beat and not beat locations, as it was found to give better results because there are chord changes on half beats. For the sake of simplicity, we will nevertheless use the term "beat-level".

As seen in Sec. III-E, when translating a HMM or a linear-chain CRF into a MLN, there is a one-to-one correspondence between probabilities and weights in the MLN. When making the model more complex by adding new formulas, there may not be any longer a one-to-one correspondence between weights and probabilities of formulas. This is why in general the weights of a MLN are learned from the data. According to [17], a good way to set the weights is to write down the probability with which each formula should hold, treat these as empirical frequencies, and learn the weights from them.

B. HMMs vs. CRFs for Tonal Harmony

1) Baseline HMM for Chord Estimation (HMMChord): We consider here a model for chord estimation that will serve as a baseline for comparison with CFR and MLNs. We utilize the baseline model for chord estimation proposed in [96] that we briefly describe here.

Let $c^i, i \in [1,24]$ denote the 24 chords of the chord lexicon. We observe a succession of $x_n = o_n, n \in [0, N-1]$ 12-dimensional chroma vectors, n being the time index, and N being the total number of beat-synchronous frames of the analyzed song. The chord progression is modeled as an ergodic 24-state HMM with a hidden variable and a single observable variable at each time step. Each hidden state $y_n, n \in [0, N-1]$ is a chord $c^i, i \in [1,24]$ of the lexicon and is observed through a chroma observation, with emission probability $p^{\text{obs}}_{HMM}(x_n|y_n)$. A state-transition matrix based on musical knowledge that reflects chord transition rules is used to model the transition probabilities $p^{\text{trans}}_{HMM}(y_n|y_{n-1})$[16].

2) Baseline CRF for Chord Estimation (CRFChord): From Sec. III-C, we equivalently model the previous HMM for chord estimation by a linear-chain CRF where the observations consist of a unique chroma feature, by using a set of transition binary features $f^{\text{trans}}_{i,j}(y_{n-1}, y_n, x_n)$ with weight of $w^{\text{trans}}_{i,j} = \log(p^{\text{trans}}_{HMM}(y_n = c^j | y_{n-1} = c^i))$, and a set of observation features $f^{\text{chroma}}_{i,o}(y_{n-1}, y_n, x_n)$ with weights $w^{\text{chroma}}_{i,o} = \log(p^{\text{obs}}_{HMM}(x_n = o_n | y_n = c^i))$.

3) Enriched CRF for Chord Estimation (CRFPriorKey): Some chords are heard as more stable within an established tonal context [98]. Various *key templates*, which represent the importance of each of the 24 triads within a given key have been proposed in the literature, as the set of 24 24-dimensional *WMCR* key templates $T^l_{\text{key}}, l \in [1,24]$ proposed in [28]. Such prior information about keys and chords relationship can be incorporated in the CRF through additional observation features. At each time instant n an observation is written $x_n = (o_n, q_n)$, where o_n is a chroma feature and q_n is a key feature. For the key features, we assume that, at each time step n, the current local key $q_n = k^l, l \in [1,24]$ is known. We factorize the observation function of Eq. (11) in two terms:

$$\Phi_{\text{obs}}(y_n, x_n) = \Phi_{\text{obs}}(y_n, o_n) \cdot \Phi_{\text{obs}}(y_n, q_n) \quad (17)$$

$\Phi_{\text{obs}}(y_n, o_n)$ is computed as in the previous section. For $\Phi_{\text{obs}}(y_n, q_n)$, we add an observation feature that reflects the correlation measure between the chord being played and the current local key: $f^{\text{key}}_{i,l}(y_{n-1}, y_n, q_n) = 1$ if $q_n = k^l$ and $y_n = c^i$, and 0 otherwise. The key templates values $T^l_{\text{key}}(i)$, can be viewed as a correlation measure that indicates the likelihood of a key q_n is $k^l, l \in [1,24]$, given that the underlying state y_n is chord $c^i, i \in [1,24]$, and are used as weights $w^{\text{key}}_{i,l}, i, l \in [1,24]$ for the key features in the CRF.

4) Inference in HMM and CRF: For the HMM and the linear-chain CRF, the most likely sequence over time can be estimated in a maximum likelihood sense by decoding the underlying sequence of hidden states y from the sequence of observations x using the Viterbi decoding algorithm [26][17]:

$$\hat{y} = \underset{y}{\text{argmax}} \quad p(y|x) \quad (18)$$

TABLE II

MLNChord	*MLNPriorKey*	*MLNLocalKey*	*MLNStruct*	*MLNMusicScale-PriorKey*	*MLNMusicScale*	MLN FOR JOINT CHORDS, LOCAL KEYS AND STRUCTURE DESCRIPTION. THE "X" ON THE LEFT INDICATE THE PREDICATES AND RULES THAT ARE USED FOR EACH MODEL.	
						Predicate declarations	
x	x	x	x	x	x	*// Observed predicates:*	
						Observation(chroma!,time)	
	x			x		LocKey (key!,time)	
x	x	x	x	x	x	$\text{Succ}_{\text{beat}}$(time,time)	
		x			x	Same_{bar}(time,time)	
			x	x	x	$\text{Succ}_{\text{struct}}$(time,time)	
						// Unobserved predicates (querry):	
x	x	x	x	x	x	State(chord! time)	
		x			x	LocKey(key!, time)	
						Weight	*Formula*
						CHORD RULES	
x	x	x	x	x	x	*Prior observation chord probabilities:*	
						w^{chord}_0	State(CM,0)
						...	...
						w^{chord}_0	State (Bm,0)
x	x	x	x	x	x	*Probability that the chroma observation has been emitted by a chord:*	
						$w^{\text{chroma}}_{\text{CM},\text{chroma}_0}$	Observation(Chroma_0,n) $\wedge$ State(CM,n)
						$w^{\text{chroma}}_{\text{C}\sharp\text{M},\text{chroma}_0}$	Observation(Chroma_0,n) $\wedge$ State(C$\sharp$M,n)
						...	...
						$w^{\text{chroma}}_{\text{Bm},\text{chroma}_{N-1}}$	Observation(Chroma_{N-1},n) $\wedge$ State(Bm,n)
x	x	x	x	x	x	*Probability of transition between two successive chords:*	
						$w^{\text{trans}}_{\text{CM},\text{CM}}$	State(CM,n_1) $\wedge$ $\text{Succ}_{\text{beat}}$($n_2$,$n_1$) $\wedge$ State(CM,n_2)
						$w^{\text{trans}}_{\text{CM},\text{C}\sharp\text{M}}$	State(CM,n_1) $\wedge$ $\text{Succ}_{\text{beat}}$($n_2$,$n_1$) $\wedge$ State(C$\sharp$M,n_2)
						...	...
						$w^{\text{trans}}_{\text{Bm},\text{Bm}}$	State(Bm,n_1) $\wedge$ $\text{Succ}_{\text{beat}}$($n_2$,$n_1$) $\wedge$ State(Bm,n_2)
						LOCAL KEY RULES	
	x	x		x	x	*Probability that the key observation has been emitted by a chord::*	
						$w^{\text{key}}_{\text{CM}_k,\text{CM}_k}$	LocKey(CM_k,n) $\wedge$ State(CM,n)
						$w^{\text{key}}_{\text{CM}_k,\text{C}\sharp\text{M}_k}$	LocKey($\text{C}\sharp\text{M}_k$,n) $\wedge$ State(CM,n)
						...	...
						$w^{\text{key}}_{\text{Bm}_k,\text{Bm}_k}$	LocKey(Bm_k,n) $\wedge$ State(Bm,n)
		x			x	*Prior observation key probabilities:*	
						w^{key}_0	LocKey(CM_k,0)
						...	...
						w^{key}_0	LocKey(Bm_k,0)
		x			x	*Probability of transition between two successive keys:*	
						$w^{\text{transKey}}_{\text{CM}_k,\text{CM}_k}$	LocKey(CM_k,n_1) $\wedge$ $\text{Succ}_{\text{beat}}$($n_2$,$n_1$) $\wedge$ LocKey(CM_k,n_2)
						$w^{\text{transKey}}_{\text{CM}_k,\text{C}\sharp\text{M}_k}$	LocKey(CM_k,n_1) $\wedge$ $\text{Succ}_{\text{beat}}$($n_2$,$n_1$) $\wedge$ LocKey($\text{C}\sharp\text{M}_k$,n_2)
						...	...
		x			x	*Minimum local key length:*	
						w_{keyDur}	LocKey(CM_k,n_1) $\wedge$ Same_{bar}(n_2,n_1) $\wedge$ LocKey(CM_k,n_2)
						...	...
						w_{keyDur}	LocKey(Bm_k,n_1) $\wedge$ Same_{bar}(n_2,n_1) $\wedge$ LocKey(Bm_k,n_2)
						SEMANTIC STRUCTURE RULES	
			x	x	x	*Probability that similar segments have the same chord progression:*	
						w^{struct}	State(CM,n_1) $\wedge$ $\text{Succ}_{\text{struct}}$($n_2$,$n_1$) $\wedge$ State(CM,n_2)
						...	...
						w^{struct}	State(Bm,n_1) $\wedge$ $\text{Succ}_{\text{struct}}$($n_2$,$n_1$) $\wedge$ State(Bm,n_2)

C. MLN for Tonal Harmony Analysis

In this section, we start with a basic model for chord recognition that we progressively enrich with additional music

[16]This transition matrix was originally proposed in the context of key estimation [97], but has been used for chords in our previous work [28], [96]. Chords and key are musical attributes related to the harmonic structure and can be modeled in a similar way.

[17]For HMM, we use the HMM Matlab toolbox [99]; for CRF, we use the UGM Matlab toolbox [100].

TABLE III

MLNChord	*MLNPriorKey*	*MLNLocalKey*	*MLNStruct*	*MLNMusiScale-PriorKey*	*MLNMusiScale*	EVIDENCE FOR JOINT CHORD, LOCAL KEY AND STRUCTURE DESCRIPTION. THE "X" ON THE LEFT INDICATE THE EVIDENCE PREDICATES THAT ARE GIVEN TO EACH MODEL.
						OBSERVATIONS
x	x	x	x	x	x	*// A chroma vector is observed at each time frame:* $\texttt{Observation(Chroma}_0\texttt{,0)}$ ... $\texttt{Observation(Chroma}_{\text{N-1}}\texttt{,N-1)}$
x	x	x	x	x	x	*// The temporal order of the frames is known:* $\texttt{Succ}_{\text{beat}}\texttt{(1,0)}$ ... $\texttt{Succ}_{\text{beat}}\texttt{(N-1,N-2)}$
						ADDITIONAL PRIOR (LOCAL) KEY INFORMATION
	x			x		*// Prior information about the key at each time instant is given* $\texttt{LocKey(CM}_k\texttt{,0)}$ *(If the key is CM at time instant 0)* ... $\texttt{LocKey(GM}_k\texttt{,N-1)}$ *(If the key is GM at time instant N-1)*
		x			x	*// Minimum local key length* *(// Beats [0:3] belong to the same bar and are likely to be in the same key)* $\texttt{Same}_{\text{bar}}\texttt{(1,0)}$ $\texttt{Same}_{\text{bar}}\texttt{(2,1)}$ $\texttt{Same}_{\text{bar}}\texttt{(3,2)}$ *(// Beats [4:7] belong to the same bar and are likely to be in the same key)* $\texttt{Same}_{\text{bar}}\texttt{(5,4)}$ $\texttt{Same}_{\text{bar}}\texttt{(6,5)}$ $\texttt{Same}_{\text{bar}}\texttt{(7,6)}$ ...
						ADDITIONAL SEMANTIC STRUCTURE PRIOR INFORMATION
		x		x	x	*// Prior information about similar segments in the structure:* $\texttt{Succ}_{\text{struct}}\texttt{(1,10)}$ $\texttt{Succ}_{\text{struct}}\texttt{(2,11)}$ ...

dimensions and relational structure links. The structure of the domain is represented by a set of weighted logical formulas. In addition to this set of rules, a set of evidence literals represents the observations and prior information. Given this set of rules with attached weights and the set of evidence literals, Maximum *A Posteriori* (MAP) inference is used to infer the most likely state of the world.

We first describe how the two structures *HMMChord* and *CRFPriorKey* can be expressed in a straightforward way using a MLN. We then build a more complex model that incorporates structural information at various time scales. For this, we propose the use of some *time predicates* that indicate links between time instants and thus that have time as argument. We consider three time scales related to the semantic and metrical structures of a music signal:

- The *micro-scale* corresponds to the beat-level and is related to the chord progression. It is associated to the $\texttt{Succ}_{\text{beat}}\texttt{(time,time)}$ time predicate that indicates two successive beat positions;
- The *meso-scale* corresponds to the bar level and is related to local key progression. It is associated to the $\texttt{Same}_{\text{bar}}\texttt{(time,time)}$ time predicate that indicates frames belonging to a same bar;
- The *macro-scale* corresponds to the global structure level. It is associated to the $\texttt{Succ}_{\text{struct}}\texttt{(time,time)}$ time predicate that indexes structurally similar segments.

1) Beat-Synchronous Level: Chord Estimation:

a) Chord Recognition (MLNChord): The chord progression modeled by *HMMChord*, and consequently *CRFChord* of sections IV-B1 and IV-B2, can be equivalently modeled in the MLN framework considering three generic formulas, given in Eqs. (19), (20), and (21), which reflect the constraints given by the three distributions defining the generative stochastic process of the HMM. The three generic formulas are described in Tab. II in the section "Chord rules".

Description of the predicates: To model the chord progression at the beat-synchronous frame level, we use an unobservable predicate $\texttt{State(c}^i\texttt{,n)}$, meaning that chord c^i (that is hidden) is played at frame n, and two observable ones, the predicate $\texttt{Observation(o}_n\texttt{,n)}$, meaning that we observe chroma o_n at frame n, and the temporal predicate $\texttt{Succ}_{\text{beat}}\texttt{(n}_2\texttt{,n}_1\texttt{)}$, meaning that n_2 and n_1 are successive frames. They are also used for evidence, see in Tab. III.

Choice of the logical formulas: As detailed in [83], conditional probability distributions $p(b|a)$ ("if a holds then b holds with some probability") are well represented using conjunctions of the form $\log(p)\ \texttt{a} \wedge \texttt{b}$ that are mutually exclusive (in any possible world, at most one of the formulas is true). In practice, in MLN implementations, the set of formulas must also be exhaustive (exactly one of the formulas should be true for every given world and every binding of the variables)[18].

For each of the three distributions of the HMM, we use mutually exclusive and exhaustive sets of formulas. This is achieved in Tab. II using the symbol !. When the predicate $\texttt{State(chord!,time)}$ is declared, this means there is one and only one possible chord per time instant. In the same way, because the observation predicate $\texttt{Observation}$ is declared as functional, if $\texttt{Observation(Chroma}_1\texttt{,n)}$ is true at time instant n, $\texttt{Observation(Chroma}_0\texttt{,n)}$, $\texttt{Observation(Chroma}_2\texttt{,n)}$, etc. is automatically false.

The prior observation probabilities are described using:

$$w_0^{\text{chord}} \quad \texttt{State(c}^i\texttt{,0)} \tag{19}$$

for each chord $c^i, i \in [1,24]$, and with $w_0^{\text{chord}} = \log p(y_0 = c^i)$ denoting a uniform prior distribution of chord symbols.

The conditional observation probabilities are described using a set of conjunctions of the form:

$$w_{i,o}^{\text{chroma}} \quad \texttt{Observation(o}_n\texttt{,n)} \wedge \texttt{State(c}^i\texttt{,n)} \tag{20}$$

for each combination of chroma observation o_n and chord c^i, and with the weights $w_{i,o}^{\text{chroma}}$ defined in Sec. IV-B2.

The transition probabilities are described using:

$$w_{i,j}^{\text{trans}} \quad \texttt{State(c}^i\texttt{,n}_1\texttt{)} \wedge \texttt{Succ}_{\text{beat}}\texttt{(n}_2\texttt{,n}_1\texttt{)} \wedge \texttt{State(c}^j\texttt{,n}_2\texttt{)} \tag{21}$$

for all pairs of chords $(c^i, c^j), i, j \in [1,24]$, and with the weights $w_{i,j}^{\text{trans}}$ defined in Sec. IV-B2.

Evidence consists of a set of ground atoms that give chroma observations corresponding to each frame, and the temporal succession of frames over time using the beat-level temporal predicate $Succ_{\text{beat}}$. Evidence is described in Tab. III.

b) Incorporating Prior Information About Key (MLNPriorKey): Prior key information can be incorporated in the MLN model, equivalently than in the case of the model

[18]Indeed cases not mentioned in the set of formulas (e.g. not writing down the formula $Observation(\text{Chroma}_1, n) \wedge State(\text{CM}, n)$ with weight $w_{\text{CM}}^{\text{chroma}}$) obtain by default an implicit weight of 0. As a result ground formulas not mentioned have a higher probability, since for $p_i \in [0,1]$, $0 \geq \log p_i$.

CRFPriorKey described in Sec. IV-B3 by simply adding a new template formula that reflects the impact of key features.

Assuming that, at each time instant, the current local key $k^l, k \in [1,24]$ is known, `LocKey` is added as a functional predicate in Tab. II (`LocKey(key!,time)`) and given as evidence in the MLN by adding in Tab. III the evidence predicates:

$$\texttt{LocKey(k}^l\texttt{,0)}, \texttt{LocKey(k}^l\texttt{,1)}, \cdots, \texttt{LocKey(k}^l\texttt{,N-1)} \quad (22)$$

An additional rule about key and chord relationship is incorporated in the model. For each pair (k^l, c^i) of key $k^l, l \in [1,24]$ and chord $c^i, i \in [1,24]$, we add the rule:

$$w^{\text{key}}_{i,l} \quad \texttt{LocKey(k}^l\texttt{,n)} \wedge \texttt{State(c}^i\texttt{,n)} \quad (23)$$

with values of the weights $w^{\text{key}}_{i,l}, i,l \in [1,24]$ defined in Sec. IV-B3. This rule "translates" the CRF key observation features.

2) Bar level: Joint Estimation of Chords and Local Key (MLNLocalKey): Using MLNs, the key can be estimated jointly with the chord progression by simply removing the evidence predicates about key listed in Eq. (22), and by considering the predicate `LocKey` as a query along with the predicate `State`. `LocKey(key!,time)` becomes an unobservable predicate and local keys are estimated from the chords, under the assumption that a chord implies a tonal context.

In addition to the template formula reflecting rules about chords and key relationships, we add rules to model key modulations in the same way that we add chord transition rules (see Eq. (21)). For this, we use the following set of generic formulas, see Tab. II, Sec. "Local key rules":

$$w^{\text{transKey}}_{ij} \quad \texttt{LocKey(k}^i\texttt{,n}_1\texttt{)} \wedge \texttt{Succ}_{\text{beat}}\texttt{(n}_2\texttt{,n}_1\texttt{)} \wedge \texttt{LocKey(k}^j\texttt{,n}_2\texttt{)} \quad (24)$$

for all pairs of keys $(k^i, k^j), i,j \in [1,24]$. Key modulations are modeled similarly to chord transitions and we use $w^{\text{transKey}}_{ij} = w^{\text{trans}}_{ij}$ (see footnote 16).

We also add a rule to capture our hypothesis from Sec. IV-A1 that key changes inside a measure are very unlikely. We add evidence indicating frames belonging to a same bar using the temporal predicate $\texttt{Same}_{\text{bar}}\texttt{(n}_2\texttt{,n}_1\texttt{)}$ (see Tab. III). We include in the model the template formula:

$$w_{\text{keyDur}} \quad \texttt{LocKey(k}^l\texttt{,n}_1\texttt{)} \wedge \texttt{Same}_{\text{bar}}\texttt{(n}_2\texttt{,n}_1\texttt{)} \wedge \texttt{LocKey(k}^l\texttt{,n}_2\texttt{)} \quad (25)$$

for each key $k^l, l \in [1,24]$, and with weight w_{keyDur}, reflecting how strong the constraint is, manually set. In practice, w_{keyDur} is a positive value (in our experiments, $w_{\text{keyDur}} = -\log(0.95)$) to avoid key changes inside a measure.

3) Global Semantic Structure Level (MLNStruct): Following the idea of designing a "structurally consistent" mid-level representation of music [69], we show that prior structural information can be used to enhance chord and key estimation in an elegant and flexible way within the framework of MLNs. As opposed to [70], we do not constrain the model to have the exact same chord progression in all sections of the same type, but we only *favor* same chord progressions for all instances of the same segment type, so that variations between similar segments can be taken into account. Here, we focus on popular music where pieces can be segmented into specific repetitive segments with labels such as *chorus*, *verse*, or *refrain*. Segments are considered as similar if they represent the same musical content, regardless of their instrumentation.

Prior structural information at the global semantic level is incorporated using the time predicate $\texttt{Succ}_{\text{struct}}$. The position of segments of same type in the song is given as evidence (see Fig. 6 for an example). Let K denote the number of distinct segments. Each segment $s_k, k \in [1,K]$ may be characterized by its beginning position (in frames) $b_k \in [1,N]$, and its length in beats l_k. For each pair of same segment type $(s_k, s_{k'})$, the position of matching beat-synchronous frames (likely to be the same chord type) is given as evidence[19]:

$$\begin{aligned} &\texttt{Succ}_{\text{struct}}\texttt{(s}_k\texttt{(b}_k\texttt{),s'}_k\texttt{(b}_{k'}\texttt{))} \quad \cdots \quad (26)\\ &\texttt{Succ}_{\text{struct}}\texttt{(s}_k\texttt{(b}_k\texttt{+l}_k\texttt{-1),s'}_{k'}\texttt{(b}_{k'}\texttt{+l}_{k'}\texttt{-1))} \end{aligned}$$

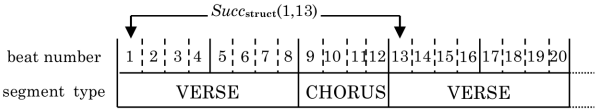

Fig. 6. Position of similar frames within a pair of same segments.

The following set of formulas is added to the Markov logic network to express the constraint that two same segments should have a similar chord progression:

$$w_{\text{struct}} \quad \texttt{State(c}^i\texttt{,n}_1\texttt{)} \wedge \texttt{Succ}_{\text{struct}}\texttt{(n}_2\texttt{,n}_1\texttt{)} \wedge \texttt{State(c}^i\texttt{,n}_2\texttt{)}$$

for all chord $c^i, i \in [1,24]$, and with weight w_{struct}, reflecting how strong the constraint is, manually set. In practice, w_{struct} will be a small positive value (in Sec. V $w_{\text{struct}} = -\log(0.95)$) to favor similar chord progressions in same segment types.

4) A Multi-scale Tonal Harmony Analysis (MLNMultiScale): The two models *MLNLocalKey* and *MLNStruct* can be unified by simply combining all the formulas into the same MLN. In this model, the chord and local key progressions are jointly estimated relying on the metrical and the semantic structure. In Sec. IV-A3, we mentioned that when adding formulas that share variables with others, this has an influence on the weights in the MLN, and it may be needed to modify them (in general by training). However, in our case, we obtained good results by combining the rules about local key and structure without changing the weights.

5) Inference in MLN: The inference step consists in computing the answer to a query. Finding the most likely state of the world y consistent with some evidence x is generally known in Bayesian networks as *Maximum Probability Explanation* (MPE) inference, while in Markov networks it is known as Maximum *A Posteriori* (MAP) inference. In Markov logic, the problem of finding the most probable configuration of a set of query variables given some evidence reduces to finding the truth assignment that maximizes the sum of weights of satisfied clauses:

$$\underset{y}{\text{argmax}}\ p(y|x) = \underset{y}{\text{argmax}} \frac{1}{Z(x)} \exp\Big(\sum_i w_i n_i(x,y)\Big) \quad (28)$$

This problem is generally NP-hard. Both exact and approximate weighted satisfiability solvers exist [19], [88], [101]. We use here exact inference with the toulbar2 branch & bound MPE inference [102] implemented in the ProbCog toolbox[20].

[19] Note that the values $s_k(b_k), \ldots, s'_{k'}(b_{k'}+l_{k'}-1)$ in Eq. (26) correspond to beat time-instants. Note also that here $l_{k'} = l_k$.

[20] Although manageable on a standard laptop, the MLN inference step has a high computational cost compared to the Viterbi algorithms for HMM and CRF (≈ 2min for *MLNChord* against 6s for *HMMChord* for processing 60s of audio on a MacBook Pro 2.4GHz Intel Core 2 Duo with 2GB RAM). We plan to explore the use of approximate algorithms and also to take advantage of the current developments on scalable inference [88]–[90].

TABLE IV
CHORDS LABEL ACCURACY *EE* RESULTS.

	Pop test-set	*Mozart test-set*
HMMChord	74.02 ± 14.61	52.80 ± 6.04
CRFChord	74.14 ± 14.59	52.94 ± 5.69
CRFPriorKey	**75.42 ± 14.10**	**53.59 ± 5.62**
MLNChord	74.02 ± 14.61	52.80 ± 6.04
MLNPriorKey	**75.31 ± 13.48**	**53.41 ± 5.76**
MLNLocalKey	72.59 ± 14.68	52.62 ± 6.44

V. EVALUATION AND DISCUSSION

In this section, we analyze and compare the performances of the various models on two test-sets of different music styles annotated by trained musicians originally proposed in [28].

A. Test-sets and Evaluation measures

The *Mozart test-set* consists of 5 movements of Mozart piano sonatas corresponding to 30 minutes of audio music. The *Pop test-set* contains 16 songs from various artists and styles that include pop, rock, electro and salsa. Details can be found in [28] and annotation are available on demand. As in [103], we map the complex chords in the annotation (such as major and minor 6^{th}, 7^{th}, 9^{th}) to their root triads.

Tonal analysis at the micro- and meso-scale rules (impact of chords and local key rules) is evaluated on both test-sets while the incorporation of macro-scale rules (impact of semantic structure rules) is only evaluated on the *Pop test-set* since the considered scenario of incorporating semantic structure rules is not relevant for classical music[21].

For chord and key evaluation, we consider *label accuracy*, which measures how the estimated chord/key is consistent with the ground truth. *EE* (*Exact Estimation*) results correspond to the mean and standard deviation of correctly identified chords/keys per song. Parts of the pieces where no key can be labeled (e.g. when a chromatic scale is played) have been ignored in the evaluation, and "Non-existing chords" (noise, silent parts or non-harmonic sounds) are unconditionally counted as errors. For local key label accuracy, we also consider the *ME* score, which gives the estimation rate according to the MIREX 2007 key estimation task [22].

Paired samples t-test at the 5% significance level are used to measure whether the difference in the results from one method to another is statistically significant or not.

B. Beat level - Equivalence With HMM, CRF and HMM

The main interest of the proposed model lies in its simplicity and expressivity for compactly encoding physical content and semantic information in a unified formalism. As an illustration of the theory, results show that the HMM and linear-chain CRF structures can be concisely and elegantly embedded in a MLN. Although the inference algorithms used for each model are different, a song by song analysis shows that chord progressions estimated by the two models are quasi identical and the difference in the results between *HMMChord*, *CRFChord* and *MLNChord* in Tab. IV is not statistically significant.

[21]This would require a much more complex model since in classical music, parts structurally similar often present strong variations such as key modulations that would need to be taken into account.

[22]The score is obtained using the following weights: 1 for correct key estimation, 0.5 for perfect fifth relationship between estimated and ground-truth key, 0.3 if detection of relative major/minor key, 0.2 if detection of parallel major/minor key. Fore more details, see http://www.mirex.org.

To illustrate the flexibility of MLNs, we also tested a scenario where some partial evidence about chords was added by adding evidence predicates of the form $State(c_i^{\mathrm{GT}}, 0)$, $State(c_i^{\mathrm{GT}}, 9)$, $State(c_i^{GT}, 19)$, $\cdots$, $State(c_i^{\mathrm{GT}}, N-1)$, as prior information of 10% of the ground-truth chords $c_i^{\mathrm{GT}}, i \in [1, 24]$. We tested this scenario on the *Fall out boy* song *This ain't a scene its an arms race* for which the *ChordMLN* estimation results are poor. They were increased from 60.5% to 76.2%, showing how additional evidence can easily be added and have a significant impact.

C. Bar level - Key as Prior Information or Query

TABLE V
LOCAL KEYS *EE* EXACT AND *ME* MIREX ESTIMATION RATE.

		Pop test-set	*Mozart test-set*
MLNLocalKey	*EE*	54.12 ± 34.69	**83.06 ± 19.38**
	ME	71.80 ± 35.08	**90.61 ± 16.36**
[28] *best*	*EE*	**61.31 ± 36.50**	80.21 ± 13.56
	ME	73.18 ± 27.56	84.81 ± 11.86
MLNMultiScale	*EE*	59.90 ± 31.50	
	ME	**76.28 ± 28.19**	

1) Prior Key Information: The MLN formalism incorporates prior information about key in a simple way with minimal model changes. It improves in general the chord estimation results (compare lines *MLNChord* and *MLNPriorKey* in Tab. IV). Fig. 7 shows an excerpt of the *Pink Floyd* song *Breathe* in E minor key. In the first instance of the Verse, at [1:15-1:20]min (dashed grey circle on measure D-1), the underlying Em harmony is disturbed by passing notes in the voice and estimated as EM with *MLNChord*. Prior key information favors Em chords and removes this error in *MLNPriorKey*.

Note that the overall improvement is not statistically significant for the *Pop test-set*, because the WMCR key templates are not adapted model chord/key relationships for some of the songs. A detailed discussion on the choice of relevant key templates according to the music genre (out of the scope of this article) can be found in [28].

2) Local Key Estimation: By considering the key as a query (i.e. by simply removing the evidence predicates about key), the model can jointly estimate chords and keys. For our datasets, this does not help improving the chord estimation results in average, which are even degraded for the *Pop test-set* (see the last line in Tab. IV). This is due to some special musical cases. For instance, the *Pink Floyd* song *Breathe* mainly consists of successive AM and Em chords. The correct key should be E dorian key, but modal keys are not modeled here. The algorithm estimates A Major key almost all the time. As a result, in the jointly estimated chord progression, most of the Em chords are labeled as EM chord (that are more likely than Em chords in A Major key).

Nevertheless, the key progression can be fairly inferred with our algorithm. In Tab. V, we report the local key estimation results obtained with our algorithm and with a state-of-the art algorithm tested on the same dataset [28]. In [28], the local key is modeled with a HMM that takes as input either chord or chroma observations. From a modeling prospective, it is difficult to make a fair comparison with [28] because the observations, the key templates and the modeling hypothesis are different from our MLN algorithm. In particular, in [28], the

metrical structure is explicitly taken into account in the model to infer the key progression. But to get an idea of the performances of the proposed model against the state-of-the art, we report the best results of all configurations tested in [28].

Local key estimation results are especially high for the *Mozart test-set* and significantly better than in [28]. Results for the *Pop test-set* are not as satisfying, again because the WMCR templates do not always reflect accurately the tonal content of pieces in this test-set[23]. However, thanks to its flexibility, the MLN allows room for improvement. Indeed, when incorporating information about the structure (see *MLNMultiScale* in Tab. V), key estimation results are comparable to the state-of-the-art results in [28], and even better for the MIREX score.

Note that local key estimation results without the rule on key transitions (see Eq. (24) in Sec. IV-C2) turned out to be poor. Also, with the chosen parameter settings, the rule that disfavors key changes inside a measure using the predicate $Same_{\text{bar}}$ (see the rule expressed by Eq. (25) in Sec. IV-C2) did not have a significant impact on the results probably because in rule (24) the weight in the clauses corresponding to transitions between two same keys is already high enough compared to those corresponding to transition from a key to a different one[24].

D. Global Semantic Structure Level

TABLE VI
CHORD *EE* RESULTS WITH SEMANTIC STRUCTURE INFORMATION. *Stat. Sig.*: STATISTICAL SIGNIFICANCE BETWEEN *MLNStruct* AND OTHERS.

	EE	*Stat. Sig.*
MLNChord	74.02 ± 14.61	} *yes*
MLNStruct	**75.31 ± 15.62**	
[70]	74.44 ± 15.13	} *no*

1) Structurally Consistent Chord Progression Estimation: In Tab. VI, we compare the results of the model *MLNStruct* with the baseline *MLNChord* modified to account for the structure in a similar way to [70], by replacing chromagram portions of same segments types by their average. The basis signal features (chroma) are the same for both methods.

The proposed approach compactly encodes physical signal content and higher-level semantic information in a unified formalism. Global semantic information can be concisely and elegantly combined with information at the beat-level time-scale so that chord estimation results are significantly improved, and more consistent with the global structure, as illustrated in Fig. 7. For instance (see the plain black rectangle measures D-1 and D-13), the ground-truth chord of the first bar of the verse is Em. *MLNChord* correctly estimates the second instance of this chord, but makes an error for the first instance (EM instead of Em). This is corrected by *MLNStruct* that favors same chord progression in same segment types.

The results obtained with the proposed model fairly compare with the previous approach [70]. The difference is not statistically significant, but the proposed model allows for taking into account variations between segments by favoring instead of exactly constraining the chord progression to be the same for segments of the same type. For instance, in measures D-6:D-7 and D-18:D-19 of the two verses of Fig. 7 (see the two dashed black rectangles), the position of the D-19:D-20 chord change is more accurate with *MLNStruct* than with *MLNChord*, presumably because of the similarity with the D-7:D-8 chord change. Also it can be seen that the two *MLNStruct* chord progressions are not exactly the same (compare measures D-6 and its counterpart D-18 in *MLNStruct*), which illustrates the flexibility of the proposed model (see the four plain grey rectangles for another example). We expect that music styles such as jazz music, where repetitions of segments result in more complex variations due to improvisation would further benefit from the flexibility of the proposed model.

TABLE VII
CHORD *EE* RESULTS OBTAINED FOR THE *Pop test-set*, WITH A MULTI-SCALE TONAL HARMONY DESCRIPTION.

MLNChord	*MLNPriorKey*	*MLNLocalKey*
74.02 ± 14.61	75.31 ± 13.48	72.59 ± 14.68
MLNStruct	*MLNMultiScale-PriorKey*	*MLNMultiScale*
75.31 ± 15.62	**76.31 ± 13.58**	74.34 ± 14.32

2) Multi-scale Tonal Harmony Analysis: The combination of all the previously described rules results in a tonal harmony analysis at multiple temporal and semantic levels. This allows improving the analysis at both the micro and meso time scales. The chord progression estimated with *MLNChord* is significantly improved with the *MLNMultiScale-PriorKey*. Moreover, the results of *MLNMultiScale-PriorKey* are also better than both those obtained with *MLNStruct* and *MLNPriorKey*, which illustrates the benefit of using multiple cues for the analysis. Also, as seen in Sec. V-C2, incorporating structure information allows significantly improving local key estimation results. Moreover, some of the errors in the chord progression estimated by*MLNLocalKey* are removed when incorporating rules on structure information in *MLNMultiScale* (see Tab. VII).

The two grey dashed rectangles in Fig. 7 (measures D-3 and D-15) illustrate the effect of the combined rules. With *MLNChord* the position of the right boundary is not accurate for chord D-3 but it is correct for chord D-15. Local key information with *MLNPriorKey* is not sufficient to correct the D-3 boundary. When similar chord progression in same segment types is enforced with *MLNStruct*, the model relies on chord D-3 and the position change is incorrect for both instances. But when combining the two rules in *MLNMultiscale*, the position of chord change is correct for both instances.

VI. CONCLUSION AND FUTURE WORK

In this article, we have introduced Markov logic as a formalism that enables intuitive, effective, and expressive reasoning about complex relational structure and uncertainty of music data. We have shown how MLNs relate to hidden Markov models and conditional random fields, two models that are typically used in the MIR community, especially for sequence labeling tasks. MLNs encompass both HMM and CRF, while being much more flexible and offering new interesting prospects for music processing.

To illustrate the potential for music processing of Markov logic networks, we have progressively designed a model for tonal harmony analysis, starting from a simple HMM. The final proposed model combines harmony-related information at various time-scales (analysis frame, phrase and global

[23]In fact the results in [28] we report here for the *Pop test-set* are obtained with other key templates (the Krumhansl key templates [98]).

[24]However, in our experiments, we saw that, as it could be expected, decreasing the value of the w_{keyDur} weight would have the impact of favoring key changes inside a measure.

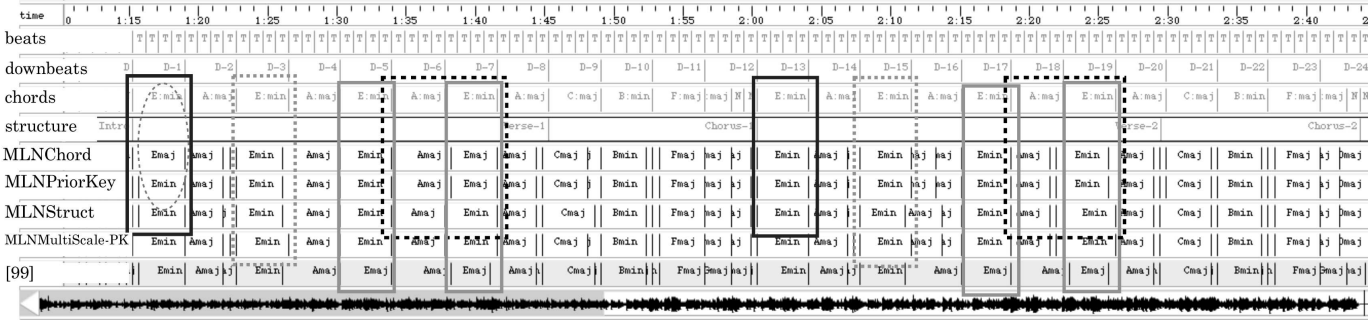

Fig. 7. *Pink Floyd* song *Breathe*. The first 4 lines (beats, downbeats, chords, structure) correspond to the ground truth annotations. The others indicate the results obtained with the various proposed models. Analysis in the text: i) Dashed grey circle (comparison between *MLNChord* and *MLNPriorKey*), ii) plain black rectangles (comparison between *MLNChord* and *MLNStruct*), iii) dashed black rectangles and plain grey rectangles (comparison of the flexibility of *MLNStruct* versus [70]), iv) dashed grey rectangles (combination of all the rules in *MLNMultiScale-PriorKey*, structure and local key given as prior information).

structure) in a single unified formalism, resulting in a more elegant and flexible model, compared to existing more ad-hoc approaches. This work is a new step towards a unified multi-scale description of audio, and toward the modeling of complex tasks such as music functional analysis.

The proposed model has a great potential of improvement in the future. Context information (metrical structure, instrumentation, chord patterns, etc.) could be compactly and flexibly embedded in the model moving toward a unified analysis of music content. Here, relational structure has been derived from background musical knowledge. Learning from labelled examples might overcome some of the shortcomings of the proposed model. The possibility of combining training with expert knowledge [86] may help leverage music complexity.

An appealing property of MLNs is their ability of constructing new formulas by learning from the data and creating new predicates by composing base predicates (*predicate invention* [104]). This should be of particular interest to the MIR community in the incoming years, considering the current expansion of annotated databases. As more and more complementary heterogeneous sources of music-related information are becoming available (e.g. video, music sheets, metadata, social tags, etc.), the development of multimodal approaches for music analysis is becoming essential. This aspect should strongly benefit from the use of statistical relational models.

MLNs are currently becoming more and more attractive to many research fields, leading to an increasing number of compelling developments, including interesting connections with deep learning [105] and deep transfer [106]. We believe that MLNs open new interesting perspectives for the field of music content processing.

VII. Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable feedback and suggestions for improvement.

References

[1] B. Schuller, "Applications in intelligent music analysis," in *Intelligent Audio Analysis*. Springer Berlin, 2013, pp. 225–298.
[2] R. Malkin, "Machine listening for context-aware computing," Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, 2006.
[3] P. Grosche, M. Müller, and J. Serrà, "Audio content-based music retrieval," in *Multimodal Music Processing*, 2012, pp. 157–174.
[4] M. Goto, "An audio-based real-time beat tracking system for music with or without drum sounds," *J New Music Res*, vol. 30, no. 2, 2001.
[5] M. Mauch and S. Dixon, "Simultaneous estimation of chords and musical context from audio," *IEEE Trans. Aud., Sp. and Lang. Proc.*, vol. 18, no. 6, pp. 1280–1289, 2010.
[6] H. Papadopoulos, "Joint Estimation of Musical Content Information From an Audio Signal," Ph.D. dissertation, Univ. Paris 6, 2010.
[7] J. Paiement, D. Eck, S. Bengio, and D. Barber, "A graphical model for chord progressions embedded in a psychoacoustic space," in *ICML*, 2005.
[8] J. Prince, S. M.A., and W. Thompson, "The effect of task and pitch structure on pitch-time interactions in music," *Memory & Cognition*, vol. 37, pp. 368–381, 2009.
[9] N. Nilsson, "Probabilistic logic," *Artif. Intell.*, vol. 28, pp. 71–87, 1986.
[10] J. Halpern, "An analysis of first-order logics of probability," in *IJCAI*, vol. 46, 1989.
[11] L. Getoor and B. Taskar, *Introduction to Statistical Relational Learning*. The MIT Press, 608 p., 2007.
[12] L. d. Raedt and K. Kersting, "Probabilistic inductive logic programming," in *Probabilistic Inductive Logic Programming*. Springer Berlin, 2008, vol. 4911, pp. 1–27.
[13] B. Taskar, P. Abbeel, and D. Koller, "Discriminative probabilistic models for relational data," in *UAI*, 2002.
[14] L. Getoor, "Learning statistical models from relational data," Ph.D. dissertation, Stanford, 2001.
[15] N. L. and P. Haddawy, "Probabilistic logic programming and bayesian networks," in *ASIAN*, 1995.
[16] K. Kersting and L. Raedt, *Towards Combining Inductive Logic Programming with Bayesian Networks*. Springer Berlin, 2001, vol. 2157, pp. 118–131.
[17] M. Richardson and P. Domingos, "Markov logic networks," *Mach. Learn.*, vol. 62, pp. 107–136, 2006.
[18] P. Domingos, S. Kok, H. Poon, M. Richardson, and P. Singla, "Unifying logical and statistical ai," in *AAAI*, 2006.
[19] P. Domingos and D. Lowd, *Markov Logic: An Interface Layer for Artificial Intelligence*. Morgan & Claypool, San Rafael, 2009.
[20] R. Crane and L. McDowell, "Investigating markov logic networks for collective classification," in *ICAART*, 2012.
[21] S. Riedel and I. Meza-Ruiz, "Collective semantic role labelling with markov logic," in *CoNLL*, 2008.
[22] H. Papadopoulos and G. Tzanetakis, "Modeling chord and key structure with markov logic," in *ISMIR*, 2012.
[23] ——, "Exploiting structural relationships in audio signals of music using markov logic," in *ICASSP*, 2013.
[24] D. Radicioni and R. Esposito, "A conditional model for tonal analysis." Springer Berlin, 2006, vol. 4203, pp. 652–661.
[25] D. Temperley, *The Cognition of Basic Musical Structures*. MIT, Cambridge, 2001.
[26] L. Rabiner, "A tutorial on HMM and selected applications in speech," *Proc. IEEE*, vol. 77, no. 2, 1989.
[27] A. Sheh and D. Ellis, "Chord segmentation and recognition using EM-trained HMM," in *ISMIR*, 2003.
[28] H. Papadopoulos and G. Peeters, "Local Key Estimation from an Audio Signal Relying on Harmonic and Metrical Structures," *IEEE Trans. Aud., Sp. and Lang. Proc.*, 2011.
[29] A. Klapuri, A. Eronen, and J. Astola, "Analysis of the meter of acoustic musical signals," *IEEE Trans. Aud., Sp. and Lang. Proc.*, vol. 14, no. 1, pp. 342–355, 2006.
[30] C. Raphael, "Automatic segmentation of acoustic musical signals using HMMs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 4, pp. 360–370, 1999.
[31] M. Ryynänen and A. Klapuri, "Transcription of the singing melody in polyphonic music," in *ISMIR*, 2006.
[32] A. Cont, "A coupled duration-focused architecture for real-time music-to-score alignment," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 6, pp. 974–987, 2010.
[33] R. Scholz, E. Vincent, and F. Bimbot, "Robust modeling of musical chord sequences using probabilistic n-grams," in *ICASSP*, 2008.
[34] K. Murphy, *Machine Learning: A Probabilistic Perspective*, ser. Adaptive Computation and Machine Learning. MIT Press, 2012.

[35] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *ICML*, 2001.
[36] S. Essid, "A tutorial on conditional random fields with applications to music analysis," 2013, presented at ISMIR.
[37] J. Burgoyne, L. Pugin, C. Kereliuk, and I. Fujinaga, "A cross validated study of modeling strategies for automatic chord recognition in audio," in *ISMIR*, 2007.
[38] T. Fillon, C. Joder, S. Durand, and S. Essid, "A conditional random field system for beat treaking," in *ICASSP*, 2015.
[39] M. McVicar, R. Santos-Rodriguez, and T. De Bie, "Learning to separate vocals from polyphonic mixtures via ensemble methods and structured output prediction," in *ICASSP*, 2016.
[40] C. Joder, S. Essid, and G. Richard, "A conditional random field viewpoint of symbolic audio-to-score matching," in *ACM*, 2010.
[41] ——, "A conditional random field framework for robust and scalable audio-to-score matching," *IEEE Trans. Aud., Sp. and Lang. Proc.*, vol. 19, no. 8, pp. 2385–2397, 2011.
[42] E. Benetos and S. Dixon, "Joint multi-pitch detection using harmonic envelope estimation for polyphonic music transcription," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 6, pp. 1111–1123, 2011.
[43] Z. Duan, L. Lu, and C. Zhang, "Collective annotation of music from multiple semantic categories." in *ISMIR*, 2008.
[44] E. Schmidt and Y. Kim, "Modeling musical emotion dynamics with conditional random fields," in *ISMIR*, 2011.
[45] K. Sumi, M. Arai, T. Fujishima, and S. Hashimoto, "A music retrieval system using chroma and pitch features based on conditional random fields," in *ICASSP*, 2012.
[46] A. Ramakrishnan, S. Kuppan, and S. Devi, "Automatic generation of tamil lyrics for melodies," in *CALC*, 2009.
[47] R. Ramirez, A. Hazan, E. Maestre, X. Serra, V. Petrushin, and L. Khan, *A Data Mining Approach to Expressive Music Performance Modeling*. Springer London, 2007, pp. 362–380.
[48] A. Anglade and S. Dixon, "Towards logic-based representations of musical harmony for classification, retrieval and knowledge discovery," in *MML*, 2008.
[49] S. Muggleton, "Inductive logic programming," *New Generat. Comput.*, vol. 8, pp. 295–318, 1991.
[50] E. Morales and R. Morales, "Learning musical rules," in *IJCAI*, 1995.
[51] E. Morales, "Pal: A pattern-based first-order inductive system," *Mach. Learn.*, vol. 26, no. 2, pp. 227–252, 1997.
[52] R. Ramirez and C. Palamidessi, *Inducing Musical Rules with ILP*. Springer Berlin, 2003, vol. 2916, pp. 502–504.
[53] A. Anglade and S. Dixon, "Characterisation of harmony with inductive logic programming," in *ISMIR*, 2008.
[54] A. Anglade, R. Ramirez, and S. Dixon, "Genre classification using harmony rules induced from automatic chord transcriptions," in *ISMIR*, 2009.
[55] A. Anglade, E. Benetos, M. Mauch, and S. Dixon, "Improving music genre classification using automatically induced harmony rules," *J. New Music Res.*, vol. 39, no. 4, pp. 349–361, 2010.
[56] G. Widmer, "Discovering simple rules in complex data: A meta-learning algorithm and some surprising musical discoveries," *Artif. Intell.*, vol. 146, no. 2, pp. 129–148, 2003.
[57] M. Dovey, N. Lavrac, and S. Wrobel, *Analysis of Rachmaninoff's piano performances using ILP*. Springer Berlin, 1995, vol. 912, pp. 279–282.
[58] E. Van Baelen, L. d. Raedt, and S. Muggleton, *Analysis and prediction of piano performances using inductive logic programming*. Springer Berlin, 1997, vol. 1314, pp. 55–71.
[59] M. McVicar, R. Santos-Rodriguez, Y. Ni, and T. de Bie, "Automatic chord estimation from audio: A review of the state of the art," *IEEE Trans. Aud., Sp. and Lang. Proc.*, vol. 22, no. 2, pp. 556–575, 2014.
[60] C.-H. Chuan and E. Chew, "The KUSC classical music dataset for audio key finding," *IJMA*, vol. 6, no. 4, pp. 1–18, 2014.
[61] M. Paulus, J. Müler and A. Klapuri, "State of the art report: Audio-based music structure analysis," in *ISMIR*, 2010.
[62] L. Euler, *A attempt at a new theory of music, exposed in all clearness according to the most well-founded principles of harmony*. Saint Petersburg Academy, 1739.
[63] K. Noland and M. Sandler, "Signal processing parameters for tonality estimation," in *AES*, 2007.
[64] A. Shenoy, R. Mohapatra, and Y. Wang, "Key determination of acoustic musical signals," in *ICME*, 2004.
[65] C. Raphael and J. Stoddard, "Harmonic analysis with probabilistic graphical models," in *ISMIR*, 2003.
[66] C. Weiss, "Global Key Extraction From CLassical Music Audio Recordings Based on the Final Chord ," in *SMC*, 2013.
[67] B. Catteau, J. Martens, and M. Leman, "A probabilistic framework for audio-based tonal key and chord recognition," in *GFKL*, 2007.
[68] T. Rocher, M. Robine, P. Hanna, and L. Oudre, "Concurrent Estimation of Chords and Keys From Audio," in *ISMIR*, 2010.
[69] R. Dannenberg, "Toward automated holistic beat tracking, music analysis, and understanding," in *ISMIR*, 2005.
[70] M. Mauch, K. Noland, and S. Dixon, "Using musical structure to enhance automatic chord transcription," in *ISMIR*, 2009.
[71] T. Fujishima, "Real-time chord recognition of musical sound: a system using common lisp music," in *ICMC*, 1999.
[72] M. Bartsch and G. Wakefield, "To catch a chorus using chroma-based representations for audio thumbnailing," in *WASPAA*, 2001.
[73] R. Klinger and K. Tomanek, "Classical probabilistic models and conditional random fields," Department of Computer Science, Dortmund University of Technology, Tech. Rep. TR07-2-013, 2007.
[74] C. Sutton and A. McCallum, "An introduction to conditional random fields," *Foundations and Trends in Machine Learning*, vol. 4, no. 4, pp. 267–373, 2012.
[75] J. Pearl, *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. Morgan Kaufmann, San Francisco, CA, 1988.
[76] C. Sutton and A. McCallum, "An introduction to conditional random fields for relational learning," in *Introduction to Statistical Relational Learning*, L. Getoor and B. Taskar, Eds. MIT Press, 2007.
[77] S. Sarawagi and W. Cohen, "Semi-markov conditional random fields for information extraction," in *NIPS*, 2004.
[78] G. Zweig and P. Nguyen, "A segmental CRF approach to large vocabulary continuous speech recognition," in *ASRU*, 2009.
[79] S. Haack, *Philosophy of Logics*. Cambridge Univ. Press, NY., 1978.
[80] D. Leivant, "Higher order logic," in *Handbook of Logic in Artificial Intelligence and Logic Programming, Volume2, Deduction Methodologies*. Clarendon Press, 1994, pp. 229–322.
[81] P. Singla and P. Domingos, "Markov logic in infinite domains," in *UAI*, 2007.
[82] M. Genesereth and N. N.J., *Logical Foundations of Artificial Intelligence*. Morgan Kaufmann, San Mateo, CA, 1987.
[83] D. Jain, "Knowledge engineering with markov logic networks: A review," in *DKB*, 2011.
[84] G. Cheng, Y. Wan, B. Buckles, and Y. Huang, "An introduction to markov logic networks and application in video activity analysis," in *ICCCNT*, 2014.
[85] T. Papai, H. Kautz, and D. Stefankovic, "Slice normalized dynamic markov logic networks," in *NIPS*, 2012.
[86] T. Pápai, S. Ghosh, and H. Kautz, "Combining subjective probabilities and data in training markov logic networks," in *ECMLPKDD*, 2012.
[87] L. Snidaro, I. Visentini, and K. Bryan, "Fusing uncertain knowledge and evidence for maritime situational awareness via markov logic networks," *Information Fusion*, vol. 21, no. 0, pp. 159–172, 2015.
[88] R. Beltagy, I. Mooney, "Efficient markov logic inference for natural language semantics," in *AAAI*, 2014.
[89] D. Venugopal, "Scalable Inference Techniques for Markov Logic," Ph.D. dissertation, University of Texas at Dallas, 2015.
[90] S. Sarkhel, D. Venugopal, T. Pham, P. Singla, and V. Gogate, "Scalable Training of Markov Logic Networks Using Approximate Counting," in *AAAI*, 2016.
[91] G. Peeters and H. Papadopoulos, "Simultaneous beat and downbeat-tracking using a probabilistic framework: theory and large-scale evaluation," *IEEE Trans. Audio, Speech, Lang. Proc.*, vol. 19, no. 6, 2011.
[92] P. Singla and P. Domingos, "Discriminative training of markov logic networks," in *AAAI*, 2005.
[93] D. Lowd and P. Domingos, "Efficient weight learning for markov logic networks," in *Knowledge Discovery in Databases: PKDD 2007*. Springer Berlin, 2007, vol. 4702, pp. 200–211.
[94] J. Fisseler, "Toward markov logic with conditional probabilities," in *FLAIRS*, 2008.
[95] M. Thimm, "Coherence and compatibility of markov logic networks," in *ECAI*, 2014.
[96] H. Papadopoulos and G. Peeters, "Joint estimation of chords and downbeats," *IEEE Trans. Aud., Sp. and Lang. Proc.*, vol. 19, no. 1, pp. 138–152, 2011.
[97] K. Noland and M. Sandler, "Key estimation using a HMM," in *ISMIR*, 2006.
[98] C. Krumhansl, *Cognitive foundations of musical pitch*. New York, NY, USA: Oxford University Press, 1990.
[99] K. Murphy, "HMM Toolbox for Matlab," http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html, 2005.
[100] M. Schmidt, "UGM: A matlab toolbox for probabilistic undirected graphical models," http://www.cs.ubc.ca/~schmidtm/Software/UGM.html, 2007.
[101] F. Niu, C. Christopher Ré, A. Doan, and J. Shavlik, "Tuffy: Scaling up statistical inference in markov logic networks using an rdbms," *PVLDB*, vol. 4, no. 6, pp. 373–384, 2011.
[102] D. Allouche, S. d. Givry, and T. Schiex, "Toulbar2, an open source exact cost function network solver," INRA, Tech. Rep., 2010.
[103] C. Harte, M. Sandler, S. Abdallah, and E. Gómez, "Symbolic representation of musical chords: a proposed syntax for text annotations," in *ISMIR*, 2005.
[104] S. Kok and P. Domingos, "Statistical predicate invention," in *ICML*, 2007.
[105] H. Poon and P. Domingos, "Sum-Product Networks: A New Deep Architecture," in *UAI*, 2011.
[106] J. Davis and P. Domingos, "Deep transfer via second-order markov logic," in *ICML*, 2009.