Learning Structured Models of Musical Syntax

Robert Lieck, Daniel Harasim, Martin Rohrmeier

École Polytechnique Fédérale de Lausanne

prename.surname@epfl.ch

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No 760081 – PMSB.

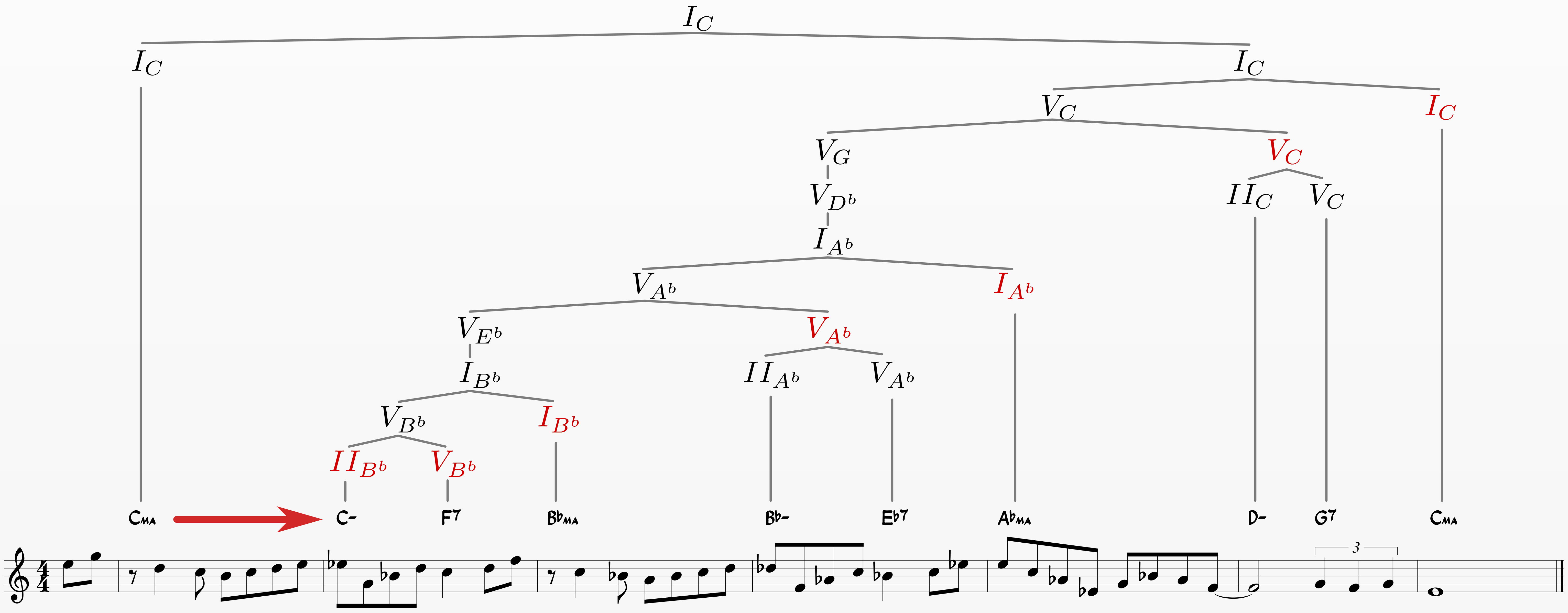

Figure 1: Afternoon in Paris, John Lewis (A-Part)

Probabilistic Context-Free Grammars

A probabilistic context-free grammar (PCFG) for musical syntax is able to generate harmony trees such as the one shown in **Figure 1** for the Jazz standard *Afternoon in Paris*. Any PCFG G, defined as in **Figure 2(a)**, can be described as a pushdown automaton (PDA). When generating a harmony sequence via leftmost derivation, at any time the left-hand side σ of the string corresponds to terminal symbols (the actual harmonies) while the right-hand side αv corresponds to non-terminal symbols and represent the state of the PDA. Transitions in the PDA only depend on the left-most non-terminal symbol α as depicted in **Figure 2(b)**. Harmony transitions are thus generated by a series of state transition of the PDA, as shown in **Figure 2(c)** for the first harmony transition in **Figure 1**.

$G = (V, \Sigma, R, S, P)$

V : non-terminal symbols

Σ : terminal symbols

R : rules $V \longrightarrow (V \cup \Sigma)^*$

S : start symbol $S \in V$

P : probability distribution over rules

(a) Probabilistic Context-Free Grammar

$$
\begin{aligned}
r(\alpha,\beta) &\in \{0,1\} : \alpha \mapsto \beta \\
p_r &= p(\sigma\beta v \,|\, \sigma\alpha v) \\
&\propto \exp \sum_{r\in R} \log(p_r)\, r(\alpha,\beta) \\
&\scriptstyle r \in R, \alpha \in V, \beta \in (V\cup\Sigma)^* \\
&\scriptstyle p_r \in P, \sigma \in \Sigma^*, v \in V^*
\end{aligned}
$$

(b) Transition Probabilities

$$
\begin{aligned}
&|\, I_C \\
&|\, I_C I_C \\
\text{CMA} \,&|\, I_C \\
\text{CMA} \,&|\, V_C I_C \\
\text{CMA} \,&|\, V_G V_C I_C \\
\text{CMA} \,&|\, V_{D^\flat} V_C I_C \\
\text{CMA} \,&|\, I_{A^\flat} V_C I_C \\
\text{CMA} \,&|\, V_{A^\flat} I_{A^\flat}, V_C I_C \\
\text{CMA} \,&|\, V_{E^\flat} V_{A^\flat} I_{A^\flat}, V_C I_C \\
\text{CMA} \,&|\, I_{B^\flat} V_{A^\flat} I_{A^\flat}, V_C I_C \\
\text{CMA} \,&|\, V_{B^\flat} I_{B^\flat} V_{A^\flat} I_{A^\flat}, V_C I_C \\
\text{CMA} \,&|\, II_{B^\flat} V_{B^\flat} I_{B^\flat} V_{A^\flat} I_{A^\flat}, V_C I_C \\
\text{CMA C-} \,&|\, V_{B^\flat} I_{B^\flat} V_{A^\flat} I_{A^\flat}, V_C I_C
\end{aligned}
$$

(c) Generating a harmony transition

Figure 2

Non-Markov Feature-Based Models

Non-Markov feature-based models can be used to generalize Markov and k-Markov (n-gram) models for sequence data. More specifically, using a log-linear condition random field model transition probabilities can be defined as

$$p(x \,|\, h) \propto \exp \sum_{f\in\mathcal{F}} \theta_f f(x,h) \tag{1}$$

$$f : \Sigma \times \Sigma^* \to \{0,1\} \ , \tag{2}$$

where $x \in \Sigma$ is the next event in the sequence, $h \in \Sigma^*$ is the history of all past events, and $\mathcal{F}$ is a set of binary features with weights $\theta_f \in \Theta$.

Unified Model

Non-Markov feature-based models are easily extended to include the (latent) state of the pushdown automaton (PDA) corresponding to a probabilistic context-free grammar (PCFG)

$$p(x,y \,|\, h,z) \propto \exp \sum_{f\in\mathcal{F}} \theta_f f(x,y,h,z) \tag{3}$$

$$f : \Sigma \times V^* \times \Sigma^* \times V^* \to \{0,1\} \ , \tag{4}$$

where x and h as above and $z, y \in V^*$ correspond to the current and next state of the PDA. In the general case, features $f \in \mathcal{F}$ may depend on the entire history of events as well as on all elements of the stack z. A PCFG, as shown in **Figure 2(b)**, corresponds to a special case where features do not depend on h and depend only on the left-most/top-most element in z

$$f_r(x,y,h,z) \equiv r(\alpha,\beta) \tag{5}$$

$$h \circ z \longrightarrow h \circ x \circ y \equiv \sigma \circ \alpha \circ v \longrightarrow \sigma \circ \beta \circ v \ , \tag{6}$$

where x, y, z, h as above and σ, α, β, v as in **Figure 2(b)**, that is, σ corresponds to the current history h of terminal symbols, α corresponds to the top-most element on the stack z, β contains x as prefix and ans suffix any new non-terminal symbols on the new stack y, while v corresponds to the unchanged portion of the stack common to z and y.

A unified model is capable of jointly modeling complex long-term dependencies, as represented by harmonic syntax trees, and local surface structures, as represented by (non-)Markov sequential models. This descriptive power is indispensable for adequately modelling smooth harmonic transitions between adjacent sub-trees, such as the first transition in **Figure 1**.

Feature-Learning with *PULSE*

The *PULSE* framework is a meta-learning framework that allows learning complex model structures, such as the feature set of a non-Markov feature-based model. *PULSE* makes use of an extension operation N^+ and a reduction operation N^- to search the structure space. The extension operation serves as a search heuristic suggesting promising extensions of the current model structure while the reduction operation implements the principle of parsimony (Occam's Razor) to find a trade-off between model complexity and descriptive power. Applying the *PULSE* framework to feature-based models iterates in three steps as illustrated in **Figure 3**: (1) the feature set $\mathcal{F}$ is extended by applying N^+, (2) the feature weights Θ are optimized using some kind of sparsity regularization (such as L_1-regularization), and (3) features with zero weight are removed from the feature set. The reduction operation N^- corresponds to step (2) and (3).

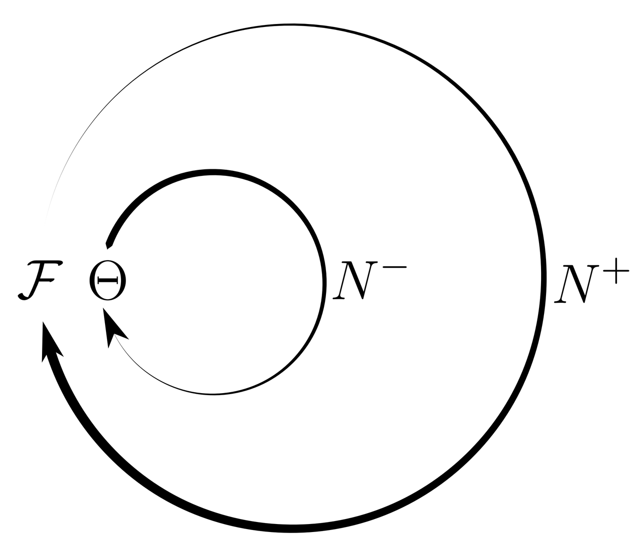

Figure 3