A conditional model for automatic orchestration Application to a real-time Live Orchestral Piano

Philippe Esling and Léopold Crestel

September 17, 2015

Abstract

1 Introduction

Musical orchestration is the subtle art of of writing musical pieces for orchestra, by combining the spectral properties specific to each instrument in order to achieve a particular sonic goal. This complex discipline involves a wide set of intricate mechanisms, most of which have not yet been satisfactorily theorized. Indeed, famous composers often conjectured that orchestration would mainly remain an empirical discipline, which could only be learned through experience and never axiomatized in books. Even if several famous musicians have written orchestration treatises [?, ?], those mostly remain recommendations and sets of existing orchestration examples from which one can draw inspiration. We focus more specifically in this work on *projective* orchestration, which is the transformation from a piano score to an orchestral piece. Many composers have worked in a projective manner, and a large amount of example can be found in the repertoire. For instance, one of the most famous is the orchestration of *Les tableaux d'une exposition*, a Modest Moussorgsky piano piece, by Maurice Ravel.

The objective in this work is to be able to automatically perform in real-time the *projective* orchestration of a piano performance. More specifically, such a system rely on the task of inferring an orchestration (output of our system) from a piano score (input). The vast combinatorial set of instrument possibilities added to the complex temporal structure of polyphonic music make this problem a particularly daunting task. Several attempts to build an automatic orchestration system can be found in the literature. Orchestration can be viewed as assigning the different notes of the piano score to a certain number of instrument according to constraints over the number of instruments, their tessitura, a certain voice leading inside an instrument. Interpreting orchestration as a Constraint Solving Problem (CSP) lead to a first solution [?]. However, as Steven McAdams pointed it out, timbre is "a structuring force in music" [?] in the sense that it should be used to emphasize the already existing movements of the original piano piece. We believe that a system only built only on symbolic constraint will undoubtedly fail at grasping this underlying structure. Hence, orchestrating first require to understand the harmonic, rhythmic and melodic structure of the original piano piece. *Orchids* ([**?**]) is an other interesting work set in an other paradigm called *injective* orchestration. It consists in trying to reconstruct a target timbre for a small temporal frame. The major drawback being that the orchestration is limited to short (less than 10 seconds) examples. In order to build an automatic orchestration system being able to work on a macro-temporal timescale while structuring the musical discourse, statistical inference appeared us as a promising solution. Their should indeed exist strong correlation between the information contained in the original piano score and the orchestral rendering we want to produce. Statistical inference would allow us to extract the knowledge and rules underlying in the many orchestration proposed by famous composers over the years.

We decided to work with a set of models called conditional models [**?**], which derive from a particular type of Markov Random Field called the Restricted Boltzmann Machine (RBM) [**?**]. While being able to model complex distribution through a multi-layer architecture of latent units, those models implement a notion of context which is interesting in our case in order to model the influence of the past over the present and of the piano over the orchestra. Those models are generative which is a requirement in our case. In conditional models, it means that once trained on a dataset, an input vector can be recreated given a certain context. This is through this mechanism that orchestral inference can be performed. To model sequences of symbolic music, the pianoroll representation is often used. A pianoroll is a matrix whose rows and columns are a discretisation of pitch and time figure **??** on page **??**. Note that this discretisation flows naturally from the scores notation in western since notes are aligned on a discrete pitch scale and rhythmically on the beat. The pitch p being played at time t is then represented in the pianoroll representation by $Pianoroll(p,t) = 1$, 0 meaning that no note is played . The dynamics are ignored and each time frame is a binary vector that indicates either a pitch is on or off. This representation, usually defined for a single polyphonic instrument can easily be extended to an orchestra composed by N instruments by simply concatenating the pianoroll of each instrument. Note that we respect the usual simplifications used when writing orchestral scores which consists in grouping all the instruments of a same section (e.g. violin 1) as a unique instrument.

We propose in this article a new evaluation framework for the orchestral inference task in order to evaluate the different model we proposed. Building a quantitative evaluation framework for generative models is rarely straightforward, especially since computing the likelihood of a test sample is intractable in the model we used. Hence, we rely on a frame-level accuracy measure to compare the orchestration proposed by our model with a *reference* orchestration. The results of the proposed model are then presented. We picked out the best model and included it in a real-time orchestration system called *LOP*.

This paper is organized as follows. In sections 2, 3 and 4 we introduce the

RBM, the CRBM and the FGCRBM architectures. The orchestration inference task is presented in the section 5 along with an evaluation framework based on a frame-level accuracy measure. The previously introduced models are then evaluated in this framework and the results displayed. The section 7 introduces a real-time *projective* orchestration system using the presented architectures.

2 Restricted-Boltzmann Machine

A Restricted-Boltzmann Machine (RBM) [HOT06] is an energy-based model that represent the joint distribution of a visible vector $\mathbf{v} = (v_1, ..., v_m)$ and a hidden vector $\mathbf{h} = (h_1, ..., h_n)$. This distribution is given by $p(v,h) = \frac{\exp^{-E(v,h)}}{Z}$ where

$$E(\mathbf{v},\mathbf{h}) = -\sum_{i=1}^{m} a_i v_i - \sum_{j=1}^{n} b_j h_j - \sum_{i=1}^{m}\sum_{j=1}^{n} v_i W_{ij} h_j \tag{1}$$

and $Z = \sum_{v,h} \exp^{-E(v,h)}$ is a usually intractable partition function. $\Theta = \{W, a, b\}$ are the weights of the network. Each unit represent an activation function which is a simple non-linear function. Unfortunately, the gradient of the negative log-likelihood of a vector from the training database $\mathbf{v}^{(l)}$ is intractable. A training algorithm called Contrastive divergence (CD) [**?**] rely on an approximation of the model driven term of this equation by running a k-step Gibbs chain to obtain a sample $v^{(l,k)}$ ((2))

$$-\frac{\partial \ln(p(\boldsymbol{v^{(l)}}|\boldsymbol{\Theta}))}{\partial \boldsymbol{\Theta}} = \mathbb{E}_{p(\mathbf{h}|\boldsymbol{v^{(l)}})}\left[\frac{\partial E(\boldsymbol{v^{(l)}},\mathbf{h})}{\partial \boldsymbol{\Theta}}\right] - \mathbb{E}_{p(\mathbf{h},\mathbf{v})}\left[\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \boldsymbol{\Theta}}\right] \tag{2}$$

$$\approx \mathbb{E}_{p(\mathbf{h}|\boldsymbol{v^{(l)}})}\left[\frac{\partial E(\boldsymbol{v^{(l)}},\mathbf{h})}{\partial \boldsymbol{\Theta}}\right] - \mathbb{E}_{p(\mathbf{h}|\boldsymbol{v^{(l,k)}})}\left[\frac{\partial E(\boldsymbol{v^{(l,k)}},\mathbf{h})}{\partial \boldsymbol{\Theta}}\right] \tag{3}$$

Running a Gibbs sampling chain consists in alternatively sampling the hidden units knowing the visible units then the visible units knowing the inferred hidden units by using the marginal probabilities ((4)).

$$p(v_i = 1|\mathbf{h}) = sigm\left(b_i^{(v)} + \sum_j W_{ij} h_j\right) \tag{4}$$

$$p(h_j = 1|\mathbf{v}) = sigm\left(b_j^{(h)} + \sum_i W_{ij} v_i\right) \tag{5}$$

where *sigm* is the sigmoid function. It has been proved that the samples we obtain after an infinite number of iteration will be drawn from the joint distribution of the visible and hidden units of our model. An other approximation consists in starting the Gibbs chain from the sample $v^{(l)}$, which increases the convergence of the chain, and to limit the number of alternate sampling steps to a fixed number K, which leads to the CD-K algorithm used to train a RBM.

3 Conditional RBM

The Conditional Restricted Boltzmann Machine (CRBM) model ([**?**]) is a standard RBM in which a dynamic biases conditioned is added to the visible and hidden units. The dynamic bias linearly depends of context units $\boldsymbol{x}$. To model time series, if we consider that the visible units represent the current time frame, those context units can be defined as the concatenation of the N last time frames. We call this vector $\boldsymbol{v}_{k<t} = \left(v_1^{(t-N)}, ..., v_m^{(t-N)}, ..., v_1^{(t)} ..., v_m^{(t)}\right)$, where N denotes the order of the model. The energy function of the Conditional RBM is given by ((6))

$$E(v_t, h_t | v_{<t}) = -\sum_i \hat{a}_{i,t} v_{i,t} - \sum_{ij} W_{ij} v_{i,t} h_{j,t} - \sum_j \hat{b}_{j,t} h_{j,t} \tag{6}$$

where the biases are defined by $\hat{a}_i = a_i + \sum_k A_{ki} v_{k,<t}$ and $\hat{b}_j = b_j + \sum_k B_{kj} v_{k,<t}$. This model appeared as an interesting solution in order to model the strong temporal relations underlying in symbolic music data. It can be trained by contrastive divergence, since the marginal probability of visible and hidden units can be easily obtained from the energy distribution.

4 Factored Gated Conditional RBM

The Factored Gated Conditional RBM model [**?**] proposes to extend the Conditional RBM model by adding a layer of feature unit z_l which modulate the weights of the conditional architecture in a multiplicative way. Hence, the weights of the networks become $\Theta = \left\{W_{ijl}, A_{ikl}, B_{jkl}, \hat{a}_i, \hat{b}_j\right\}$. This multiplicative influence can be understood as a modification of the energy landscape of the model. Since the number of parameter to train becomes high, the 3 dimensional tensors can be factorized into a product of three matrices by including factors $W_{ijl} = W_{if}.W_{jf}.W_{lf}$. The energy function of this Factored Gated Conditional RBM is then given by

$$E(v_t, h_t | v_{<t}, y_t) = -\sum_f \sum_{ijl} W^v_{if} W^h_{jf} W^z_{lf} v_{i,t} h_{j,t} z_{l,t} - \sum_i \hat{a}_{i,t} v_{i,t} - \sum_j \hat{b}_{j,t} h_{j,t} \tag{7}$$

where the dynamic biases of the visible and hidden units are defined by

$$\hat{a}_{i,t} = a_i + \sum_m \sum_{kl} A^v_{im} A^{v<t}_{km} A^z_{lm} v_{k,<t} z_{l,t} \tag{8}$$

$$\hat{b}_{j,t} = b_j + \sum_n \sum_{kl} B^h_{jn} B^{v<t}_{kn} B^z_{ln} v_{k,<t} z_{l,t} \tag{9}$$

5 Orchestration inference

An orchestration inference task is introduced in this section, along with an evaluation framework and the performances of our model in this framework. The evaluation is a frame-level prediction task that rely on an accuracy measure.

5.1 Generative models

The previously introduced models are generative models. After the training phase, the distribution represented by the networks is supposed to be close to the underlying distribution of the data. It is then possible to sample from this distribution to reproduce data that are alike the data of the training set. Conditional models allow to impose a certain context, which enables to generate sequences of data under a certain context. We remind that our objective is to transform a sequence of binary vectors drawn from a piano score (refered to as piano vectors) into a sequence of orchestral vectors (section 1). (SCH2MA) If we consider that the visible units represent the orchestral vector for the time frame t, conditional units can be used to model the influence of the past orchestral vector $t-1, ..., t-N$ (the concatenation of those N orchestral frames are then referred to as past orchestral vector) and the (strong) influence of the piano frame at time t over the visible units. In the CRBM model, the conditional units are the concatenation of the past orchestral vector and the current piano vector. The FGCRBM offer the possibility to distinguish between the influence of the current piano vector and the past orchestral vectors by assigning the first one to the features units z_l and the second to the context units x_k.

5.2 Frame-level accuracy

5.3 Database

We used a parallel database of piano scores and their orchestration by famous composers. The database consists of 76 *XML* files. Given the complexity of the distribution we wanted to model and the reduced size of the fatabase we have accessed to, we decided to keep as a test dataset only the last half of one track from our database. Hence 75 and a half files were used to train our model. We chose to do so in order to have the best generation ability. For each instrument, the pitch range is reduced to the tessitura observed in the training dataset. We used a rhythmic quantization of 8 frame per beat, which means that the smallest symbolic rhythm is a 32^{th} note.

5.4 Results

6 Live Orchestral Piano

7 Conclusion and future works

References

[HOT06] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7):1527–1554, July 2006.