Probabilistic Factor Oracles for Multidimensional Machine Improvisation

Ken Déguernel, Emmanuel Vincent, Gérard Assayag

Probabilistic Factor Oracles for Multidimensional Machine Improvisation

Ken Déguernel[1,2] ken.deguernel@inria.fr

Emmanuel Vincent[1] emmanuel.vincent@inria.fr

Gérard Assayag[2] gerard.assayag@ircam.fr

[1]Université de Lorraine, CNRS, Inria, Loria, F-54000 Nancy

[2]STMS Lab - IRCAM, CNRS, Sorbonne Université

Authors' final version. This article has been accepted for publication in *Computer Music Journal* 42:2.

Abstract

This paper presents two methods using training over multidimensional sequences for automatic improvisation. We consider as dimensions musical features such as melody, harmony, timbre, etc. We first present a system combining interpolated probabilistic models with a factor oracle. The probabilistic models are trained on a corpus to learn the correlation between dimensions and are used to guide the navigation in the factor oracle that ensure a logical improvisation. Improvisations are therefore created in a way where the intuition of a context is enriched with multidimensional knowledge. We then introduce a system creating multidimensional improvisations based on communication between dimensions via probabilistic message passing. The communication infers some anticipatory behaviour on each dimension now influenced by the others, creating a consistent multidimensional improvisation. Both systems are evaluated by professional improvisers during listening sessions. Overall, they receive good feedback and show encouraging results, first on how multidimensional knowledge can help performing better navigation in the factor oracle and second on how communication through message passing can emulate the interactivity between dimensions or musicians.

Introduction

Our goal is to design a system able to generate multidimensional musical improvisations. By "dimensions", we mean musical features such as melody, harmony, rhythm, timbre, etc. [Bimbot et al. (2014)]. To achieve this goal, this system must be able to learn correlations between dimensions on a large musical corpus and, at the same time, be able to follow a local frame constructed from a musician's live playing or from a smaller corpus (e.g. a single composer or a single piece) that constrains the improvisation.

Several systems have been developed over the years for machine improvisation, focusing first on one-dimensional improvisation with one-dimensional training, using different methods from statistical sequence modelling such as compression-inspired incremental parsing [Dubnov et al. (1998)], Markovian models [Pachet (2002)] and other machine learning techniques [Conklin and Witten (1995); Dubnov et al. (2003)]. The use of a factor oracle from the field of string processing, paved the way to the popular OMax interactive improvisation software [Assayag et al. (2006); Surges and Dubnov (2013)]. Several ideas have spawned around the OMax project to approach the concept of polyphonic information in automatic improvisation. ImproteK [Nika et al. (2017)] has been developed for music based on temporal scenarios (for instance a chord chart in jazz music). This system uses prior knowledge of a scenario that can represent another dimension than the one being generated, to guide the improvisation. Donze et al. (2013) use an automaton to control a melodic improvisation through rule based grammars with information from other dimensions. However, in these examples, the generated improvisations are still one-dimensional, and the training is also one-dimensional. Indeed, Improtek focuses on co-occurrences between the generated dimension and the specific scenario and Donze et al. (2013) assume manually specified rules, which do not generalise to other musical styles. The SoMax project [Bonasse-Gahot (2014)] uses active listening over several dimensions to guide the improvisation. However, the views of each dimension activate places in the memory separately and do not consider the relations between the different dimensions. Training over several dimensions for one-dimensional generations has been studied for music analysis and automatic composition. Raczyński et al. (2013) interpolate probabilistic models of melody, harmony and tonality for a harmonisation task. Methods using deep and/or recursive neural networks have also been employed to create harmonisations [Bellgard and Tsang (1999)] and melodies over chord sequences [Bickerman et al. (2010)]. However, these systems are not constrained by a local frame, making it difficult to adapt to the particular style of human musician in real time. More recently, multidimensional generation with multidimensional training has been studied. Padilla and Conklin (2016) generate counterpoints in the style of Palestrina with vertical viewpoints [Conklin (2002)] representing the correlation between two voices. Valle et al. (2016) propose an extension of the work by Donze and colleagues, where improvisation rules are obtained through data-mining and therefore could be adapted automatically to different styles given a representative corpus. However, in these models they use multidimensional symbols raising overfitting issues and making it impossible to generate co-occurences of elements that were not seen in the training corpus. Methods using convolutional neural networks have also been used to generate multidimensional music from raw audio by Van Den Oord et al. (2016) and on symbolic data by Yang et al. (2017). However, once again, these systems cannot adapt to a local frame.

In this article, we propose two systems. First, we present a system using training over multidimentional sequences to guide its one-dimensional improvisation. Then, we introduce a system generating multidimensional improvisations based on multidimensional training. The first system was introduced in [Déguernel et al. (2016)] and combines interpolated probabilistic models with a factor oracle. On the one hand, the interpolated probabilistic models enable the system to consider the correlations between dimensions and to benefit from advanced smoothing and optimisation techniques. They represent the "cultural background" of the system and can be trained on different corpora. On the other hand, the factor oracle represents the local frame of the improvisation as usual in OMax. This enables the system to consider a context of variable length, similar to Variable Markov Models [Wang and Dubnov (2014)], usually longer than the one represented by the probabilistic model and to benefit from the expertise of the heuristics developed for the navigation in the factor oracle in OMax [Assayag and Bloch (2007)]. By combining these two aspects, we are able to create improvisations following the logic of a local frame enlightened by a global multidimensional knowledge. In the present article, we extend our work on the first system by conducting an evaluation with listening sessions with professional improvisers. The second system uses several agents communicating through a cluster graph via message passing [Koller and Friedman (2009)]. Probabilistic Graphical Models have proven to be an efficient representation for the communication between musicians during a situation of free improvisation by Kalonaris (2016) but have not yet been used for multi-agent music generation. Each agent represents either a dimension or a musician and is represented by both a cultural background and a local frame. The communication between agents makes them make a decision following their own logic and knowledge, but influenced by the others in an interactive way. This combination results in a multidimensional improvisation. This system is also evaluated by professional improvisers.

In the first section we recall the theory behind probabilistic model interpolation and smoothing techniques, the factor oracle and the heuristics used in OMax for navigation. In the second section, we introduce the system combining probabilistic models with the factor oracle. In the third section, we introduce the use of a cluster graph for the communication between agents (each represented in our case by one factor oracle). We first explain the theory of cluster graphs and the belief propagation algorithm and then propose a model combining a cluster graph and probabilistic factor oracles. Finally, we present the results of our listening session for both systems.

Probabilistic model interpolation and factor oracle

We present here the different theoretical tools used by our systems based on the state of the art. We first present the probabilistic model interpolation method enabling our systems to take the multidimensional aspect of music into account. Then, we present the factor oracle, the structure used in OMax that enables our systems to adapt to the local frame of an improvisation.

Probabilistic model interpolation

In this section, we adapt probabilistic model interpolation methods for music generation. We first describe the global method and then present how the use of smoothing techniques can prevent overfitting when using training corpora of a limited size.

Method

[Raczyński et al. (2013)] used probabilistic models for automatic harmonisation on a classical music corpus. We adapt this method for music generation. The goal is to create probabilistic models able to predict the evolution of one musical dimension using information from multiple dimensions. For instance, let us consider the problem of predicting the melody note M_t played at time t (encoded by the pitch). We want to estimate $P(M_t|X_{1:t})$ where $X_{1:t}$ is a set of musical variables from various dimensions from time 1 to t. Such a model cannot be computed in practice due to its high combinatorics when using several dimensions on several time frames, the set of possibilities being the Cartesian product of the set of possibilities of each dimension in each time frame. Using probabilistic model interpolation enables us to consider several tractable sub-models P_i depending only on a subset of musical variables $A_{i,t} \subset X_{1:t}$ in order to approximate the global model. The interpolation can be linear [Jelinek and Mercer (1980)]:

$$P(M_t|X_{1:t}) = \sum_{i=1}^{I} \lambda_i P_i(M_t|A_{i,t}) \tag{1}$$

where I is the number of sub-models and $\lambda_i \geq 0$ are the interpolation coefficients such that $\sum_{i=1}^{I} \lambda_i = 1$. The interpolation can also be log-linear [Klakow (1998)]:

$$P(M_t|X_{1:t}) = Z^{-1} \prod_{i=1}^{I} P_i(M_t|A_{i,t})^{\gamma_i} \tag{2}$$

where $\gamma_i \geq 0$ are the interpolation coefficients and Z is the normalising factor:

$$Z = \sum_{M_t} \prod_{i=1}^{I} P_i(M_t|A_{i,t})^{\gamma_i}. \tag{3}$$

This method enables us to consider as many sub-models as we want. For instance, a sub-model can be a bigram $P(M_t|M_{t-1})$ predicting which melody note to play given the previous melody note, or a model representing "which melody note to play on which chord": $P(M_t|C_t)$. The chosen sub-models are trained on a training corpus: the probability of each submodel is estimated using a counting function over all the elements appearing in the corpus. Then the interpolation coefficients are estimated on a validation corpus in order to approximate at best the global model. This estimation is done using the cross-entropy metric, which is equivalent in this case to the Kullback-Leibler divergence [Kullback and Leibler (1951)] between the model and the validation corpus up to an additive constant :

$$H(M) = -\frac{1}{T} \sum_{t=1}^{T} \log_2 P(M_t|X_{1:t}) \ , \tag{4}$$

where T is the number of time frames in the validation corpus. The cross-entropy represents the system's lack of understanding. Therefore, the interpolation coefficients are estimated to minimise the cross-entropy. The most relevant sub-models will be assigned large interpolation coefficients while irrelevant sub-models will receive interpolation coefficients close to zero.

Smoothing techniques

When learning from a training corpus, it is common that all the observed elements in the training corpus do not include every single element that could appear during the test. This especially occurs when the training corpora are limited, which is usually the case for music improvisation, because corpora cannot be expected to reach the virtually infinite possibilities of a free improvisation. This leads to zero-value probabilities that can prevent some possible elements to be taken into consideration. Moreover, if the sub-models chosen to represent the corpus are too complex, overfitting can occur. Smoothing techniques are used to correct the probabilities estimated from a limited corpus and prevent overfitting. Plenty of smoothing techniques have been created to fit various applications. The following two techniques are among the most popular [Chen and Goodman (1998)] :

1. *Additive smoothing :* we consider that every possible element appears δ times more than it actually appears in the corpus.

$$P_{\text{add}}(X|Y) = \frac{\delta + \text{count}(X, Y)}{\sum\limits_{X'} \delta + \text{count}(X', Y)} \tag{5}$$

   where X is the event, Y is the context and count is the function counting the number of times an element (here, a pair of elements) appears in the corpus. This smoothing enables the model to overcome the problem of zero-value probabilities, since every element will appear at least δ times.

2. *Back-off smoothing :* we interpolate the considered model with a lower order model.

$$P_{\text{back-off}}(X|Y) = \lambda P(X|Y) + (1-\lambda)P(X|Z) \tag{6}$$

   where Z is a subset of Y and λ is the interpolation coefficient. For instance, if $P(X|Y)$ is an n-gram, then $P(X|Z)$ could be an $(n-1)$-gram. This smoothing enables the model to overcome the problem of overfitting. This smoothing technique can be used recursively. We can notice that back-off smoothing is actually a generalisation of additive smoothing, since by recursion we always end up with a uniform distribution of all elements (0-gram).

Using probabilistic models enables us to take into consideration several dimensions and the correlation between them. However, when the models are used alone for generation, there is a lack of consistency, because there is no component enforcing some kind of repetition and local logic in the improvisation.

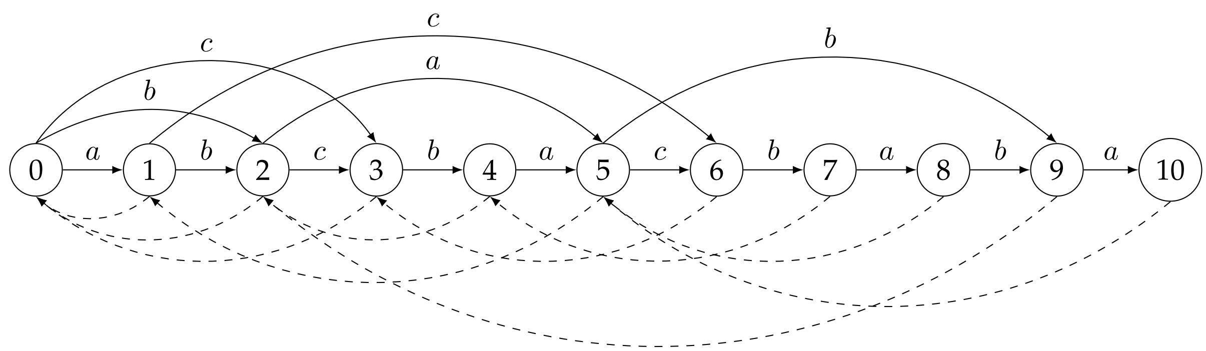

Figure 1. Example of factor oracle constructed on the word $w = abcbacbaba$. Solid arrows are the transitions and dashed arrows are the suffix links. The suffix links connect each state to the leftmost previous state with which it shares the largest common context.

Factor oracle in the OMax paradigm

The factor oracle is a structure from the field of text algorithms, first introduced by [Allauzen et al. (1999)] for optimal string matching and then used for computing repeated factors and data compression by [Lefebvre and Lecroq (2000)]. It is an acyclic automaton representing al least all the factors in a word w. The construction algorithm is incremental and $O(|w|)$ in time and space. This structure was first adapted to music generation by [Assayag and Dubnov (2004)]. An example of factor oracle is shown in Figure 1 on the word $w = abcbacbaba$. This structure offers two main points of interest. First, it keeps the linear aspect of what is being learnt. For instance, in Figure 1, we can notice that the full word can be found following the horizontal arrows. Second, suffix links are created during its construction. These link places in the memory with a similar context. For instance, in Figure 1, we can notice that the states 5 and 8, linked by a suffix link, share the context cba. The musical idea is that it is possible to jump from one point in the memory to another one linked by a suffix link, thereby creating a new musical sentence but still preserving the musical style.

In [Assayag and Bloch (2007)], heuristics are developed for navigation in the factor oracle in order to create more realistic improvisations, with for instance the use of a continuity factor in order to avoid too many jumps, the use of a taboo list to avoid loops, etc. These heuristics also prevent the use of suffix links connecting states with a common context smaller than a fixed threshold. The factor oracle showed good results for improvisation style modelling and has since been widely used in machine improvisation systems such as OMax, ImproteK or PyOracle. However, this structure is not appropriate for multidimensional sequences. When considering several dimensions, the amount of possible events is drastically increased (the alphabet would be the Cartesian product of the alphabet of each dimension). Therefore, places in the memory with a similar context would be rare, even perhaps inexistent, limiting the generation to something extremely similar, or an exact replica of the memory, which would not be considered as an original improvisation.

Factor oracle exploiting a probabilistic model

We introduce a system creating improvisations in a closer way to a human improviser whose intuition of a context is enriched by knowledge and a cultural background [Crispell (2000)]. The idea is to benefit from both the multidimensional training of probabilistic models and the proficiency of the heuristics developed for the factor oracle and for its extremely efficient scheme for incrementally building up a variable Markov type of linear memory.

On the one hand, a probabilistic model is created to represent the knowledge and cultural background of the musician we want to emulate. We select a set of sub-models over the dimensions we want to take into consideration and apply interpolation and smoothing techniques in order to compose our global probabilistic model. This probabilistic model can be trained offline, prior to the performance, on a significant corpus representing the multidimensional knowledge acquired through our musician avatar's lifetime. On the other hand, during the performance we construct a factor oracle from a human musician's playing in a way similar to that of OMax, in an online fashion or from any reduced set of music, such as a single piece, following the dimension we want to generate (for instance, the melody). This constitutes the local frame of the improvisation.

We then generate the improvisation creating a path in the factor oracle as with OMax except that we guide the improvisation using the probabilistic model. The factor oracle enforces the sequential logic and organic development of the motive being generated and enables the system to consider a longer context than the probabilistic model. This is thanks to the suffix links connecting each state with the previous state with the longest common context and to the heuristics developed in OMax ensuring the use of suffix links connecting states with at least a minimal common context. The probabilistic model provides a deeper knowledge of music, thanks to its training on a larger corpus, and enables the system to consider multidimensional information and re-enforce higher level structures such as harmony over the purely sequential logic.

At each step of the navigation, if we are in state i of the factor oracle, we compute the set of attainable states $\mathrm{Att}(i)$ considering the heuristics from [Assayag and Bloch (2007)]. Then considering the musical contents of state i $\mu_i = \{\mu_i^M, \mu_i^C, ...\}$, that is to say the set of musical variables stored in state i during the factor oracle construction (for instance, μ_i^M represents the melody note of state i and μ_i^C represents the chord of state i), the musical contents of all attainable states and possibly some information from the environment, we compute a score for each potential transition corresponding to the interpolation of the smoothed sub-models from the probabilistic model. The scores are then normalised by the sum of the scores for each attainable state to obtain tran-

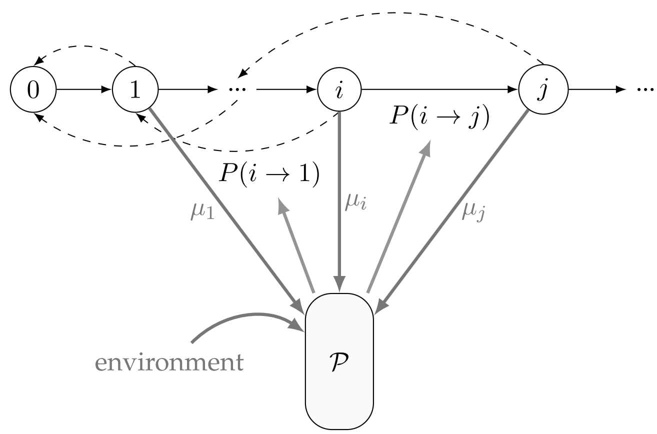

Figure 2. Using a multidimensional probabilistic model $\mathcal{P}$ with a factor oracle. Let us consider that from state i, the only reachable states are state j and state 1. Using the context, μ_1, and μ_i, $\mathcal{P}$ is able to compute a score for the transition from state i to 1. Similarly, for the transition from state i to j using the context, μ_i and μ_j. The scores are then normalised to get $P(i \to 1)$ and $P(i \to j)$.

sition probabilities. For instance, if we are generating the melody note M_t, for all $j \in \text{Att}(i)$, the transition probability from state i at time $t-1$ to state j at time t is :

$$P(i \to j | X_{1:t}) = \frac{P(M_t = \mu_j^M | X_{1:t})}{\sum\limits_{k \in \text{Att}(i)} P(M_t = \mu_k^M | X_{1:t})} \tag{7}$$

Finally, for generation, we chose the transition at random using those proper transition probabilities. Figure 2 illustrates the process for one step. The decision process for the navigation in the factor oracle is therefore enriched by the cultural background encoded in the probabilistic model.

This system could therefore provide better guidance on the leading dimension with multidimensional information. For instance, it could help reducing the brutal density or intensity changes that can be sometimes heard in systems such as OMax, if these dimensions were to be considered. Moreover, with active listening, the multidimensional aspect of this system can be used to guide the improvisation with environmental information as in SoMax. It is also possible to consider the creation of hybrid musicians, combining knowledge and local context from different styles. An extension of this system could be to replace the probabilistic model with a recurrent or deep neural network [Eck and Lapalme (2008)], in order to learn longer-term dependencies and/or more intricate relations between dimensions.

Cluster graphs and message passing between oracles

In this section, we propose a model where several factor oracles can communicate through message passing. Each oracle can represent either a musical dimension or a musician. The main idea is to get closer to multi-agent systems that are more representative of a real free collective improvisation scenario. The method we propose could therefore be used to create a polyphonic and/or multidimensional improvisation, for instance a multi-instrument improvisation, a florid counterpoint or a melody / accompaniment duet. In the case of a multi-instrument scenario, this could represent the interactions between musicians, all trying to anticipate what the others are going to play in order to guide their own logic in their improvisation to have a real collective play. This could also represent the cognitive process of an individual musician playing over several dimensions, trying to figure out the best way to conduct their improvisation using knowledge from all these dimensions (e.g. by improvising simultaneously over the melodic and harmonic dimensions). The different oracles communicate with probabilistic messages giving information about what they are about to do to inform the others. This way every agent can make an informed decision accordingly. Message passing is organised on a graph representing which dimension each agent is working on and which dimensions it is listening to.

We chose to use belief propagation on a cluster graph because it deals with inferring information with probabilistic model, and therefore is compatible with the system in the previous section. The main idea of this technique is that each agent has an initial belief based on its knowledge on a set of variables. These agents then communicate on some of the variables they share. Considering its initial belief and the information from the other agents, each agent can better estimate the marginal probability of its set of variables. We first present the theoretical tools needed to use the belief propagation algorithm and then we present our method for multidimensional improvisation.

Cluster graph and message passing

In this section, we present the theoretical tools used to model the interaction between the factor oracles. We first present the cluster graph structure and its properties and then present the belief propagation algorithm. This section is a summary of the work presented in [Koller and Friedman (2009)] about these tools.

Cluster graph

Let X be a set of random variables. A factor ϕ is a function from Val(X) to $\mathbb{R}$. Note that this includes both joint probabilities and conditional probabilities. This will correspond to our sub-models, for instance a bigram on the melody $P(M_t|M_{t-1})$. The set of variables X is called the scope of the factor and noted Scope$[\phi]$. A cluster graph $\mathcal{U}$ for a set of factors Φ over a set of variables X is an undirected graph for which each vertex is associated a subset of variables $\mathcal{K}_i \subseteq X$ named cluster and each edge between two clusters $\mathcal{K}_i$ and $\mathcal{K}_j$ is associated with a "separation set" or *sepset* $\mathcal{S}_{i,j} \subseteq \mathcal{K}_i \cap \mathcal{K}_j$, that is a subset of variables shared by the two clusters about which they will communicate.

Considering a set of factors $\Phi = \{\phi_1, ..., \phi_k\}$, each ϕ_k is assigned to a cluster $\mathcal{K}_{\alpha(k)}$ such that

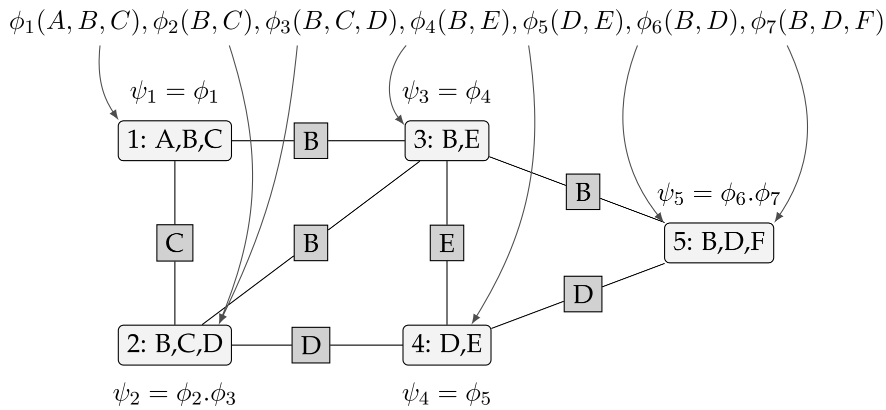

Figure 3. Example of factor distribution on a cluster graph. The clusters are represented by the rounded rectangle, they consist of a set of variables. The sepsets are represented by the dark squares, they consist of a subset of the common variables between the two clusters they connect. The clusters can communicate with their neighbours about the variables of their shared sepset.

Scope$[\phi_k] \subseteq \mathcal{K}_{\alpha(k)}$. The initial belief of the cluster $\mathcal{K}_i$ is defined by

$$\psi(\mathcal{K}_i) = \prod_{k;\alpha(k)=i} \phi_k \; . \tag{8}$$

Figure 3 gives an example on how to distribute factors on a cluster graph. Note that, in this example, other distributions could have been chosen, for instance ϕ_2 could have been assigned to $\mathcal{K}_1$ instead of $\mathcal{K}_2$. In this case, we would have $\psi_1(A,B,C) = \phi_1(A,B,C).\phi_2(B,C)$ and $\psi_2(B,C,D) = \phi_3(B,C,D)$.

A cluster graph must follow these properties (the example in Figure 3 satisfies them):

1. *Family Preservation :* for each factor $\phi_k \in \Phi$, there must be a cluster $\mathcal{K}_i$ such as Scope$[\phi_k] \subseteq \mathcal{K}_i$. This way, we make sure that every factor can be assigned to a cluster and more generally that all the information we want to take into account can be included in the cluster graph.

2. *Running Intersection Property :* for each pair $(\mathcal{K}_i, \mathcal{K}_j)$ of clusters and any variable $A \in \mathcal{K}_i \cap \mathcal{K}_j$, there is a unique path between $\mathcal{K}_i$ and $\mathcal{K}_j$ on which every cluster and sepset includes A. This is equivalent to the fact that, for any variable A, the set of clusters and sepsets including A forms a tree. This property has two consequences. First, the existence of this path enables the information about A to travel to every cluster including A. Second, the uniqueness of this path prevents the situation where the information about A goes in circle spawning false rumours. For example, in Figure 3, although cluster 1 and cluster 2 both contain the variable B, the sepset $\mathcal{S}_{1,2}$ contains only C, omitting B in order to avoid a circular path that would include cluster 3.

Belief Propagation algorithm

The belief propagation algorithm is based on probabilistic message passing between clusters. The message passed from cluster i to cluster j over the variables from the sepset $\mathcal{S}_{i,j}$ is noted $\delta_{i\to j}(\mathcal{S}_{i,j})$ and is defined by :

$$\delta_{i\to j}(\mathcal{S}_{i,j}) = \sum_{\mathcal{K}_i - \mathcal{S}_{i,j}} \psi_i \prod_{k\in(N_i - \{j\})} \delta_{k\to i} \tag{9}$$

where N_i is the neighbourhood of i, i.e., the set of clusters that share a sepset with $\mathcal{K}_i$.
For instance, in Figure 3, the messages passed between cluster 1 and 3 are :

$$\delta_{1\to 3}(B) = \sum_{A,C} \psi_1(A,B,C)\delta_{2\to 1}(C)$$

$$\delta_{3\to 1}(B) = \sum_{E} \psi_3(B,E)\delta_{2\to 3}(B)\delta_{4\to 3}(E)\delta_{5\to 3}(B)$$

Note that $\delta_{i\to j}(\mathcal{S}_{i,j})$ does not depend on $\delta_{j\to i}(\mathcal{S}_{i,j})$. This prevents a repetitive sending of the information received from a cluster back to the same cluster which would result in the spawning of false rumors.

The belief propagation algorithm follows these steps :

1. Assign each factor ϕ_k in Φ to a cluster $\mathcal{K}_{\alpha(k)}$.
2. Compute the initial beliefs $\psi_i(\mathcal{K}_i) = \prod_{k:\alpha(k)=i} \phi_k$.
3. Initialise all the messages to 1.
4. Repeat message updates following formula (9).
5. Compute final beliefs :

$$\beta_i(\mathcal{K}_i) = \psi_i \prod_{k\in N_i} \delta_{k\to i} \tag{10}$$

For a cluster, the final belief is a new factor based on its initial belief updated by inference of the information from the other clusters. $\beta_i(\mathcal{K}_i)$ is an approximation of the marginal probability $P(\mathcal{K}_i)$.

The convergence of the belief propagation algorithm is not guaranteed for any cluster graph. Theoretically, the more complex the graph is the more complex convergence is. Note also that the order in which the messages are updated can have an influence on the convergence and on how fast it is. However, there is no way to determine the optimal order for message updates, this being completely dependent on the cluster graph construction. Synchronous message updates, where all messages are updated at the same time, have been proven to give the worst result in practice. In what follows, we have chosen to do message updates in a random order to avoid any bias. Even if theoretical convergence is not guaranteed, this algorithm shows good results in practice, except for very complex graphs with more than a thousand variables, which is not the case here [Koller and Friedman (2009)].

Communication between oracles for improvisation

Our goal is to use the combination of smoothed sub-models with the belief propagation algorithm on a cluster graph in order to make several factor oracles communicate with each other and therefore create a multidimensional improvisation where several dimensions are generated at the same time. Each oracle represents a dimension or a musician and is trained on a context accordingly. The paths on the oracles are guided both by the probabilistic models defining the initial potentials and by the messages passed between oracles through the cluster graph. The idea is that the agents will try to find a common ground, communicating to each other their musical expectations, considering their separate knowledge. This way, the oracles make a general choice of their path from internal and external knowledge. In Figure 4 we show the cluster graph we used to create an improvisation with both melodic and harmonic data. This could be use for instance to represent a pianist freely improvising both the chords and melody, or the interplay between a pianist and a saxophonist in a free improvisation context. We use n-gram models for melody and for harmony, respectively $P(M_n|M_{n-1})$ and $P(C_n|C_{n-1})$ and models representing the direct relations between melody and harmony : $P(M_n|C_n)$ and $P(C_n|M_n)$. Two oracles are constructed on the local context : Oracle M on melody and Oracle C on harmony. For each oracle two clusters are created : a first one for the temporal aspect of the dimension, and a second one for direct relation between the two dimensions. Note that this cluster graph respects both the family preservation and running intersection properties and is therefore suitable for the belief propagation algorithm.

At each step of the generation, each oracle provides its attainable states and its musical contents. The probabilistic model computes the factors ϕ_i corresponding to the smoothed sub-models. This provides the initial potential for each cluster of the graph. The messages $\delta_{i,j}$ for the belief propagation are then all updated ten times, each time in a new random order. We then compute the final beliefs $\beta_i(\mathcal{K}_i)$ for each cluster. Therefore, we can estimate $P(M_n)$ from $\beta_1(\mathcal{K}_1) = \beta(M_n, M_{n-1})$ or $\beta_2(\mathcal{K}_2) = \beta(M_n, C_n)$. For instance :

$$P(M_n) \simeq \sum_{C_n} \beta(M_n, C_n). \tag{11}$$

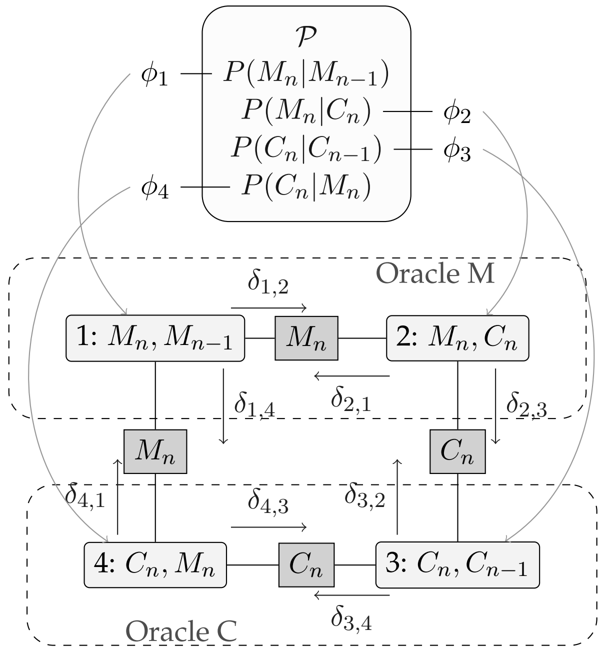

Figure 4. Cluster graph for multidimensional melody and harmony improvisation.

In theory, if the algorithm converged and produced exact inference :

$$\sum_{C_n} \beta(M_n, C_n) = \sum_{M_{n-1}} \beta(M_n, M_{n-1}) = P(M_n). \tag{12}$$

On the other hand, we can estimate $P(C_n)$ from $\beta_3(\mathcal{K}_3) = \beta(C_n, C_{n-1})$ or $\beta_4(\mathcal{K}_4) = \beta(C_n, M_n)$. The estimated $P(M_n)$ and $P(C_n)$ are normalised to obtain transition probabilities respectively in Oracle M and Oracle C, as in the previous section. Each oracle then takes a decision regarding its own transition following these transition probabilities.

This model can be extended to a higher number of dimensions, musicians or a higher number of sub-models as long as the constructed cluster graph follows the family preservation and running intersection properties. Moreover, one of the main benefits of this method is that it would be possible to use several probabilistic models (one per oracle) trained on different corpora to emulate the style of different musicians, creating an individuality for each agent, and making this system more versatile than a system using a centralised knowledge with joint probabilities.

Experimentation

To evaluate the methods presented in this paper, we have generated improvisations using Charlie Parker's Omnibook [Parker and Aebersold (1978)] as a corpus. This corpus consists of 50 tunes composed, played and improvised on by Charlie Parker with symbolic melodic and harmonic data. This corpus can be found at http://repmus.ircam.fr/dyci2/ressources.

This bebop jazz musician has a fairly distinctive style and is therefore a good choice to assess the style modelling of our methods. We divided this corpus into three non-overlapping sub-corpora : a training corpus consisting of 40 tunes and improvisations in order to train the different sub-models; a validation corpus consisting of 5 tunes and improvisations in order to optimise the interpolation and smoothing coefficients ; a test corpus consisting of 5 tunes and improvisations used to create the factor oracles during generation.

In order to have a qualitative evaluation of the generated improvisations, we conducted listening sessions with three professional jazz musicians: Pascal Mabit, a saxophonist and jazz teacher, graduated from the Conservatoire National Supérieur de Musique et de Danse de Paris, Louis Bourhis, a double bassist, graduated from the Haute École de Musique de Lausanne and Joel Gauvrit, a pianist and jazz teacher, graduated from the Conservatoire National Supérieur de Musique et de Danse de Lyon. As professional jazz musicians, they are very familiar with the music of Charlie Parker and therefore able to provide us valuable feedback.

Factor oracle and probabilistic model

In this section, we present the results of the two experiments done with our first system. We first evaluate the impact of using a probabilistic model to guide a factor oracle. Then, we evaluate the impact of the training corpus when guiding a factor oracle with a probabilistic model.

Guiding improvisation with a probabilistic model

In order to evaluate our model with a factor oracle exploiting a probabilistic model, we conducted two experiments. First, we generated free improvisations, in order to compare improvisations generated by a factor oracle alone to improvisations generated by a factor oracle combined with a probabilistic model. For the latter, we chose to use two sub-models:

1. a bigram on the melody $P_1(M_t|X_{1:t}) = P(M_t|M_{t-1})$,
2. a model representing the correlations between melody and harmony $P_2(M_t|X_{1:t}) = P(M_t|C_t)$.

Those sub-models and the interpolation and smoothing coefficients are trained on the Omnibook corpus, respectively on the training corpus and the validation corpus. In this experiment, the harmony is not played. However, the chord chart of the original tune is followed when generating an improvisation with the probabilistic model. We generated a dozen improvisations by both methods on two tunes: *Anthropology* and *Donna Lee* (*Anthropology* uses rhythm changes, i.e., the same chord progression as Gershwin's *I've Got Rhythm*).Examples of generations for this experiment are available at http://repmus.ircam.fr/dyci2/demos/probabilistic_fo.

After only a few examples, the three musicians noticed a clear difference between the two methods in the organisation of the improvisation. With the first method, Bourhis said that the improvisations are *"patchwork"* of Parker elements without a feeling of consistency. Mabit added that with this method the harmonic progressions were not clear, or arranged in a random way, except when the improvisation consisted of direct quotes from the theme:

> *"Harmony makes sense in a continuity. [...] At the moment, it doesn't take that into account, or it is juxtaposing them in a random manner. We don't really hear harmony. We hear note after note, or phrases after phrases. And even inside phrases, there is not necessarily any harmonic sense."*

When using a probabilistic model, both Bourhis and Mabit were able to say which chords the improvisation was playing on, despite them not being played. Moreover, Mabit found that there was a clear sense of the succession of tonal centres. Despite that, the improvisation preserves the global style of Charlie Parker thanks to the local context provided by the factor oracle. On *Donna Lee*, Bourhis underlined that this harmonic clarity made the improvisation easier to follow and it would be easier to do some comping with it :

> *"We feel much more at home when we hear that. [...] For instance, you can clearly hear the modulation to the fourth degree or the relative on the two places where they are characteristic. It does it the right way. We hear that it follows something. So it is much easier to understand."*

Gauvrit, focusing more on the melodic phrases noticed an improvement on the organisation of the sequence of phrases when using the probabilistic model. The phrases feel less disjointed and more structured:

> *"I feel like the elements are more developed, that there is more unity. [...] It feels like there is an idea being developed. Not globally, but something that sounds like reality where there is an idea that brings another one, that brings another and so on."*

However, Gauvrit and Mabit noticed that on top of some harmonic mistakes, there are still some hazy moments in the improvisation, especially on the bridge of *Anthropology* due to a lack of understanding of the global form of the chord chart.

More generally, the improvisations make sense from a harmonic point on view on a local scale, but lack of construction and logic with regard to the position in the chord chart. This comment was expected since this problem exists in every system in the OMax paradigm and our method did not intend to solve this particular problem. Bourhis said:

> *"When it will understand the idea of global form, it will be even better, because at the moment, I feel as if it takes the chords one after the other. [...] What it does works with the chords but it doesn't always make sense."*

Moreover, Bourhis regrets the lack of harmonic anticipation and melodic leading to the future chord saying that the improviser *"knows what it does but not where it goes"*. This comment was also expected since no anticipatory methods were implemented.

This first experiment showed good results overall. The impact of probabilistic model can be noticed by professional jazz musicians and the generated improvisations are preferred over those generated from a factor oracle alone.

About the corpus choice

We then conducted a second experiment to see if differences could be heard when using probabilistic models trained on different corpora. We generated several improvisations on *Anthropology* and *Donna Lee* without any rhythmic information (only quarter notes and quarter rests were played) to avoid rhythmic offsetting (that would occur for instance when playing only two thirds of a triplet) on the respective chord charts that are now being played along with the improvisation to highlight the melody/harmony relations. We first generated improvisations using the Omnibook corpus for training, and then using a training corpus consisting of about a thousand classical music pieces instead. The factor oracles are in both cases constructed on Charlie Parker's tune. Examples of generations for this experiment are available at http://repmus.ircam.fr/dyci2/demos/corpus_choice.

First, Mabit noticed that with both methods, the global idea of Charlie Parker's style is still present, even when using the classical music corpus. This can be explained by the dominance of the local context provided by the factor oracle. However, after more listening, he pointed out that when using the classical music corpus the improvisations seemed to aim more for the notes in the chords than when using the Omnibook corpus. The improvisations seemed more careful, and therefore sounded better from a harmony point of view.

> *"The most credible method in my opinion is the one with the classical music corpus. It works better because there is a better consideration of the harmonic spaces, it takes more into consideration what is going on on each chord. [...] It sounds like someone who plays with the harmony and takes some liberties."*

Bourhis explains the difference saying that the improvisations generated with the classical music corpus are more strict from a harmonic point of view but less representative of Parker's style from a melodic point of view. Gauvrit underlined the difference between the two methods in a similar way saying that when using the classical music corpus, the improvisations sounded *"less altered"*, and sometimes even *"Broadway-like"*:

> *"There is less harmonic inconsistency with the classical music corpus. With the Omnibook, there are quite a few things that sound out, a bit twisted, but at the same time, that's what*

makes them sound more jazz [...] It's less surprising with the classical one, it's more square, more academic."

The results of this experiment are encouraging, the three musicians were able to notice a difference when using different corpora. However, the preferred corpus depends of the esthetics and personnal tastes of the musicians. Mabit tended to prefer the classical music corpus while Bourhis and Gauvrit preferred the improvisations generated with the Omnibook.

Cluster graph and communication

To evaluate our interactivity model with cluster graph and message passing, we used the cluster graph previously shown in Figure 4 to generate both melody and harmony. Once again, no rhythmic information was considered for the melody which plays only quarter notes and quarter rests. The probabilistic model was trained on the Omnibook corpus. We generated multidimensional improvisations on *Anthropology* and *Donna Lee*, on which the melodic and harmonic factor oracles were constructed. Both dimensions were played. Examples of generations for this experiment are available at http://repmus.ircam.fr/dyci2/demos/cluster_graph.

First of all, the three musicians praised the logic of the generated harmonic progressions saying that it worked in all the examples generated, sounded like a real jazz song, and could have easily been played upon. Mabit thought that the generated improvisation were quite realistic, and could even represent a real life situation:

"It's funny, it really sounds like a wacky idea from the Conservatoire National Supérieur de Musique experimental improvisation class. Like, we work one month on Donna Lee, just Donna Lee, and now we know the chords and play Donna Lee but in an unstructured way."

The three musicians also said that the relations between the two dimensions made sense overall. Gauvrit and Bourhis raised the same criticism from the previous experiences were made about the melody about the lack of global organisation. On top of that, Mabit noticed that the melody followed the harmony properly, but might be too subordinated to the harmony, and therefore was less convinced by the generated melody that felt a bit bland at times and was not enough reactive:

"It seems like the two voices kind of know, or exactly know what is going on with each other at all time, so it is the point where they know too much and it restricts them."

Generally, the generated multidimensional improvisations seemed quite realistic and musical. Even if sometimes feeling a bit too constrained and lacking global organisation, this system improvises over several dimensions with both a horizontal and a vertical logic, and provides encouraging results.

Conclusion and discussion

We presented two methods able to learn multidimensional information in order to generate musical improvisations. First, we have shown the musical potentialities of combining probabilistic models with a factor oracle to guide the improvisation. The probabilistic models provide an efficient way to represent the relation between dimensions and can benefit from advanced smoothing techniques and optimisation for interpolation that make them an efficient and comprehensive way to model the cultural background of a musician. The factor oracle is a structure that exploits efficient heuristics to represent the local context and the logic behind the development of a motive played by a musician. Therefore the proposed method is able to follow the contextual logic of an improvisation while enriching its musical discourse from multidimensional knowledge in a closer way to a human improviser. Second, we have introduced a method modelling the interactivity between several musicians, or between several dimensions in an improviser's mind. This method is able to generate actual multidimensional improvisations. The communication between agents is conducted via a cluster graph. Smoothed probabilistic models are used as prior knowledge, and a belief propagation algorithm with message passing is used. Once again, the local context of each dimension is represented by a factor oracle. This way each agent is able to make a global decision regarding its own generation using both internal and external knowledge.

Both methods were evaluated during listening sessions with professional jazz musicians. Both methods received good feedback overall and seemed to be able to generate quite realistic improvisations. Some limitations of the current status of these methods were raised during the listening sessions, especially about the lack of a global form for the melodic improvisations. This could be studied for instance with the use of recurrent neural networks (for the probabilistic aspect) [Eck and Lapalme (2008)] or with a generative grammar describing the multi-scale organisation of the improvisation (for the deterministic aspect) [Lerdahl and Jackendoff (1983); Chomsky (1996)].

These methods could be adapted to work with other existing improvisation systems such as ImproteK, PyOracle, etc. in order to improve their results. It would also be interesting to compare the multi-agent system and message passing with other formalism than the belief propagation, for instance with gossip algorithms [Vanhaesebrouck et al. (2017)]. This work also opens the door to musicology research towards creating more realistic avatars of musicians, by trying to find out what were the influences of the human musician we want to emulate and by training probabilistic models on a corpus comprising these influences, while focusing generation on this musician's own music to create the factor oracle.

Acknowledgements

This work is made with the support of the French National Research Agency, in the framework of the project DYCI2 "Creative Dynamics of Improvised Interaction" (ANR-14-CE24-0002-01) http://repmus.ircam.fr/dyci2/project and with the support of Region Lorraine.

References

Allauzen, C., M. Crochemore, and M. Raffinot. 1999. "Factor oracle : a new structure for pattern matching." In *Proceedings of SOFSEM'99, Theory and Practice of Informatics*. pp. 291–306.

Assayag, G., and G. Bloch. 2007. "Navigating the oracle : a heuristic approach." In *Proceedings of the International Computer Music Conference*. pp. 405–412.

Assayag, G., G. Bloch, M. Chemillier, A. Cont, and S. Dubnov. 2006. "OMax Brothers : a dynamic topology of agents for improvization learning." In *Workshop on Audio and Music Computing for Multimedia*. pp. 125–132.

Assayag, G., and S. Dubnov. 2004. "Using factor oracles for machine improvisation." *Soft Computing* 8-9:604–610.

Bellgard, M. I., and C. P. Tsang. 1999. "Harmonizing music the Boltzmann way." In N. Griffith, and P. M. Todd, (editors) *Musical Networks*. MIT Press, pp. 261–277.

Bickerman, G., S. Bosley, P. Swire, and R. M. Keller. 2010. "Learning to create jazz melodies using deep belief nets." In *Proceedings of the International Conference on Computational Creativity*. pp. 228–236.

Bimbot, F., G. Sargent, E. Deruty, C. Guichaoua, and E. Vincent. 2014. "Semiotic description of music structure: an introduction to the Quaero/Metiss structural annotations." In *Proceedings of the AES 53rd International Conference on Semantic*. pp. 32–43.

Bonasse-Gahot, L. 2014. "An update on the SoMax project." Technical report, IRCAM.

Chen, S. F., and J. Goodman. 1998. "An empirical study of smoothing techniques for language modeling." Technical Report TR-10-98, Harvard University.

Chomsky, N. 1996. *Studies on semantics in generative grammar*, volume 107. Berlin: Walter de Gruyter.

Conklin, D. 2002. "Representation and discovery of vertical patterns in music." In *Proceedings of the 2nd International Conference of Music and Artificial Intelligence*. pp. 32–42.

Conklin, D., and I. H. Witten. 1995. "Multiple viewpoint systems for music prediction." *Journal of New Music Research* 1(24):51–73.

Crispell, M. 2000. "Elements of improvisation." In J. Zorn, (editor) *Arcana : Musicians on Music*. New York: Hips Road, pp. 190–192.

Déguernel, K., E. Vincent, and G. Assayag. 2016. "Using multidimensional sequences for improvisation in the OMax paradigm." In *Proceedings of the 13th Sound and Music Computing Conference*. pp. 117–122.

Donze, A., S. Libkind, S. A. Seshia, and D. Wessel. 2013. "Control improvisation with application to music." Technical Report UCB/EECS-2013-183, EECS Department, University of California, Berkeley.

Dubnov, S., G. Assayag, and R. El-Yaniv. 1998. "Universal Classification Applied to Musical Sequences." In *Proceedings of the International Computer Music Conference*. pp. 332–340.

Dubnov, S., G. Assayag, O. Lartillot, and G. Bejerano. 2003. "Using machine-learning methods for musical style modeling." *IEEE Computer* 10(38):73–80.

Eck, D., and J. Lapalme. 2008. "Learning musical structure directly from sequences of music." Technical Report 6128, University of Montreal, Department of Computer Science, CP.

Jelinek, F., and R. L. Mercer. 1980. "Interpolated estimation of Markov source parameters from sparse data." In *Pattern Recognition in Practice*. pp. 381–397.

Kalonaris, S. 2016. "Markov networks for free improvisers." In *Proceedings of the 42nd International Computer Music Conference*. pp. 181–185.

Klakow, D. 1998. "Log-linear interpolation of language models." In *Proceedings of the 5th International Conference on Spoken Language Processing*. pp. 1695–1698.

Koller, D., and N. Friedman. 2009. *Probabilistic graphical models : principles and techniques*. Cambridge: MIT Press.

Kullback, S., and R. A. Leibler. 1951. "On information and sufficiency." *Annals of Mathematical Statistics* 21(1):79–86.

Lefebvre, A., and T. Lecroq. 2000. "Computing repeated factors with a factor oracle." In *Proceedings of the 11th Australasian Workshop On Combinatorial Algorithms*. pp. 145–158.

Lerdahl, F., and R. Jackendoff. 1983. *A Generative Theory of Tonal Music*. Cambridge: MIT Press.

Nika, J., M. Chemillier, and G. Assayag. 2017. "ImproteK: introducing scenarios into human-computer music improvisation." *ACM Computers in Entertainment* 4(2):4 :1–27.

Pachet, F. 2002. "The Continuator : musical interaction with style." In *Proceedings of the International Computer Music Conference*. pp. 211–218.

Padilla, V., and D. Conklin. 2016. "Statistical generation of two-voice florid counterpoint." In *Proceedings of the 13th Sound and Music Computing Conference*. pp. 380–387.

Parker, C., and J. Aebersold. 1978. *Charlie Parker Omnibook*. Los Angeles: Alfred Music Publishing.

Raczyński, S. A., S. Fukayama, and E. Vincent. 2013. "Melody harmonisation with interpolated probabilistic models." *Journal of New Music Research* 42(3):223–235.

Surges, G., and S. Dubnov. 2013. "Feature selection and composition using PyOracle." In *Proceedings of the 2nd International Workshop on Musical Metacreation*. pp. 114–121.

Valle, R., A. Donzé, D. J. Fremont, I. Akkaya, S. A. Seshia, A. Freed, and D. Wessel. 2016. "Specification Mining for Machine Improvisation with Formal Specifications." *Computers in Entertainement - Special Issue on Musical Metacreation* 14(3):6 :1–20.

Van Den Oord, A., S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. 2016. "Wavenet : a generative model for raw audio." *arXiv preprint* arXiv:1609.03499.

Vanhaesebrouck, P., A. Bellet, and M. Tommasi. 2017. "Decentralized collaborative learning of personalized models over networks." In *Proceedings of the International Conference on Artificial Intelligence and Statistics*. pp. 509–517.

Wang, C., and S. Dubnov. 2014. "Guided Music Synthesis with Variable Markov Oracle." In *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*. pp. 55–62.

Yang, L., S. Chou, and Y. Yang. 2017. "MidiNet : a convolutional generative adversarial network for symbolic-domain music generation using 1D and 2D conditions." In *Proceedings of the 18th International Society for Music Information Retrieval Conference*. pp. 324–331.