Using Multidimensional Sequences For Improvisation In The OMax Paradigm

Ken Déguernel, Emmanuel Vincent, Gérard Assayag

Using Multidimensional Sequences For Improvisation In The OMax Paradigm

Ken Déguernel
Inria
STMS Lab, Ircam / CNRS / UPMC
ken.deguernel@inria.fr

Emmanuel Vincent
Inria
emmanuel.vincent@inria.fr

Gérard Assayag
STMS Lab, Ircam / CNRS / UPMC
gerard.assayag@ircam.fr

ABSTRACT

Automatic music improvisation systems based on the OMax paradigm use training over a one-dimensional sequence to generate original improvisations. Different systems use different heuristics to guide the improvisation but none of these benefits from training over a multidimensional sequence. We propose a system creating improvisation in a closer way to a human improviser where the intuition of a context is enriched with knowledge. This system combines a probabilistic model taking into account the multidimensional aspect of music trained on a corpus, with a factor oracle. The probabilistic model is constructed by interpolating sub-models and represents the knowledge of the system, while the factor oracle (structure used in OMax) represents the context. The results show the potential of such a system to perform better navigation in the factor oracle, guided by the knowledge on several dimensions.

1. INTRODUCTION

Current automatic music improvisation systems such as OMax [1] are able to learn the style of a one-dimensional musical sequence (a melody represented by a sequence of pitches or timbral audio features) in order to generate original improvisations by recombining the musical material. This style modeling can be performed live from a musician's playing or offline with a corpus. Several systems have been developed over the years using statistical sequence modeling [2], Markovian models [3] and other machine learning techniques [4]. However, most of these systems do not take the correlations between several musical dimensions (pitch, harmony, rhythm, dynamic, timbre...) into account.

Taking into consideration multiple dimensions and the relations between them has been an issue for systems out of the OMax paradigm. ImproTek [5, 6] makes use of a prior knowledge of a scenario (for example a chord chart) to guide the improvisation. SoMax [7] uses an active listening procedure enabling the system to react to its environment by activating places in its memory. PyOracle [8] uses information dynamics on audio features to create improvisations. Donze et al. [9] use an automaton in order to control the melodic improvisation with information about other dimensions. But in all of these, the actual training is still done on a one-dimensional sequence.

Training on multidimensional sequences has been studied by Conklin et al. [10] with multiple viewpoint systems where different attributes of a melody (such as pitches, intervals, contour...) are linked together for melody prediction on Bach chorales. These systems have also been studied for four part harmonisation [11]. Raczyński et al. use interpolated probabilistic models to do melody harmonisation [12]. This work proposes a flexible way to create a global model from chosen sub-models whose weight can be optimised and can be used in practice since the size of the model is reduced in order to learn the dependencies between dimensions. This method also uses smoothing techniques [13] to reduce overfitting issues that would otherwise arise. Some multidimensional models based on deep neural networks have also been proposed for the harmonisation problem [14] or to create jazz melodies [15]. In this case, the dependencies between dimensions are implicitly represented in the hidden layers.

In this article we present a way to use interpolated probabilistic models to create improvisations taking into account multiple musical dimensions and the correlations between them while keeping the benefits of the OMax paradigm and its factor oracle based representation [16], in particular its linear time oriented graph structure and optimised navigation scheme that make it a proficient tool for improvised performance and interaction. These are well-established methods that can profit from advanced smoothing and optimisation techniques. Moreover, they provide more explanatory models than neural network and therefore can provide us a deeper insight into the studied musical style or the improviser's mind.
We combine these models with the factor oracle [17] structure used in OMax, thus creating a new system with a musical training, able to use prior multidimensional knowledge to guide itself in an improvisation context described by the factor oracle.

In section 2, we explain how interpolation of probabilistic models can be used to take multiple dimensions into account for melody generation. Then, in section 3, we introduce a system combining probabilistic models with the factor oracle. And finally, in section 4 we present some results of experimentations done with this new system.

2. INTERPOLATION OF PROBABILISTIC MODELS

2.1 Method

Our system relies on the work of Raczyński et al. in [12] on automatic harmonisation. We want to create a probabilistic model able to predict the melody given information from different musical dimensions. Let us denote by M_t the melody played at time t, represented by the pitch. We want to predict :

$$P(M_t|X_{1:t}) \tag{1}$$

where $X_{1:t}$ is a set of musical variables from times 1 to t. This model is able to take into account multiple musical dimensions since the musical variables included in $X_{1:t}$ can be from several dimensions.

However, the combinatorics behind such a model are too high, the set of possibilities being the cartesian product of the set of possibilites of each dimension. Therefore such a prediction cannot be used in practice. To make it applicable, we approximate this global model by interpolating several sub-models P_i, which are easier to compute, depending only a subset of the musical variables $A_{i,t} \subset X_{1:t}$. For instance, we can use an n-gram model over a single dimension, or models representing the direct interaction between dimensions, for example, “which note should I play at time t knowing the harmony at this time?”.

The interpolation can be linear [18] :

$$P(M_t|X_{1:t}) = \sum_{i=1}^{I} \lambda_i P_i(M_t|A_{i,t}) \tag{2}$$

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 [19] :

$$P(M_t|X_{1:t}) = Z^{-1} \prod_{i=1}^{I} P_i(M_t|A_{i,t})^{\gamma_i} \tag{3}$$

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

$$Z = \sum_{M_t} \prod_{i=1}^{I} P_i(M_t|A_{i,t})^{\gamma_i}. \tag{4}$$

The optimisation over the interpolation coefficients enable the system to accept as many sub-models as possible. The most relevant sub-models will have a high interpolation coefficient while irrelevant sub-models will receive an interpolation coefficient close to zero. This could be extended with some sub-model selection similar to Model M [20].

Two methods of smoothing techniques are used, the latter being a generalisation of the former. [13].

- First we are going to use an additive smoothing which consist of considering that every possible element appears δ times more than it actually appears in the corpus, with usually $0 < \delta \leq 1$.

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

  where c is the function counting the number of times an element appears in the corpus. This smoothing enable the model to overcome the problem of zero probabilities which often occurs with small training corpora.

- Then, we are going to use a back-off smoothing which consist of using information from 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. For instance, if $P(X|Y)$ is a n-gram, then $P(X|Z)$ could be a $(n-1)$-gram. This smoothing enable the model to overcome the problem of overfitting

2.2 Application to improvisation

In order to test sub-model interpolation for melody generation, we have used a corpus of 50 tunes from the Omnibook [21] composed, played and improvised on by Charlie Parker. We divided this corpus into three 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 using cross-entropy minimisation,
- a test corpus consisting of 5 tunes and improvisations.

We decided to use two sub-models :

$$P_1(M_t|X_{1:t}) = P(M_t|M_{t-1})$$

$$P_2(M_t|X_{1:t}) = P(M_t|C_t)$$

where M_t represents the melody at time t, and C_t represents the chord label at time t.

We applied a combination of additive smoothing and back-off smoothing techniques using $P(M_t)$ as a lower order model. Therefore, for the linear interpolation, we have :

$$\begin{aligned} P(M_t|X_{1:t}) &= \alpha P(M_t) + \beta U(M_t) \\ &\quad + \lambda_1 P(M_t|M_{t-1}) + \lambda_2 P(M_t|C_t) \end{aligned} \tag{7}$$

where α and β are the smoothing coefficients corresponding respectively to the back-off smoothing and additive smoothing, U is the uniform distribution and λ_1 and λ_2 are the interpolation coefficients. The conditional probabilities are estimated using the counting function c.

	coefficients				cross-entropy
	λ_1	λ_2	α	β	$H(M)$
B+M	**0.582**	**0.129**	**0.289**	**0**	**4.543**
B	0.672	0	0.328	0	4.572
M	0	0.639	0.361	0	4.881
U	0	0	0.998	0.002	5.858

Table 1. Cross-entropy results (bits/note) with linear interpolation. The results are shown for the smooth interpolation of the bigram model and melody/chord model (B+M), then for the bigram model with smoothing (B), then for the melody/chord model with smoothing (M), and finally with the smoothing alone (U) as a point of comparison.

In order to evaluate this model, we used the cross-entropy on the test corpus :

$$H(M) = -\frac{1}{T}\sum_{t=1}^{T} \log_2 P(M_t|X_{1:t}) \,. \tag{8}$$

This metric is in this case equivalent to the KL-divergence up to an additive constant and represents the lack of understanding of the system. Therefore, the lower the cross-entropy, the better the model prediction power.

In Table 1, we present some of the results obtained with linear interpolation. Note that all the results are shown with the same smoothing technique in order to allow a proper comparison. As shown, the model has a better prediction power when using sub-model interpolation. However, the improvement is quite small in term of cross-entropy. This can be explained by the fact that the cross-entropy represents the system's ability to reproduce the test data, while improvisation is not about reproduction but about creativity, and as we said improvisation possibilities are unlimited.

However, informal listening tests show some improvement when using the interpolated model compared to a classic n-gram model. But generated improvisation with just this probabilistic model lack of consistency and of a local organisation. Therefore, we have decided to go further using this type of probabilistic model by combining them with the oracle factor.

3. FACTOR ORACLE EXPLOITING A PROBABILISTIC MODEL

The factor oracle is a structure coming from the field of bioinformatics and language theory [17, 22] that has been widely used in automatic improvisation systems such as OMax [1, 16], ImproTek [5], SoMax [7] or PyOracle [8]. This structure is able to keep the linear aspect of what is being learnt and create links, called suffix links, between places in the memory with a similar context. An example of factor oracle is shown Figure 1.

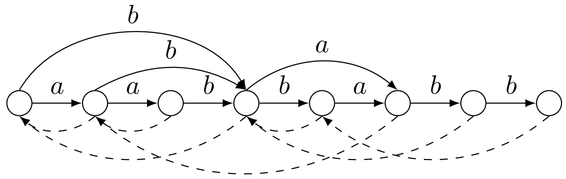

Figure 1. Example of factor oracle constructed on the word $w = aabbabb$. Horizontal solid arrows are the transition, bent solid arrows are the factor links and dashed arrows are the suffix links.

We designed a system combining the probabilitic model able to take into account the multidimensional aspect of music, with the contextual setting brought by the factor oracle. The idea was to conceive a system creating improvisation in a way closer to a human improviser. We were inspired by this quote from Marilyn Crispell's *Elements of Improvisation* [23] (written for Cecil Taylor and Anthony Braxton) :

> *The development of a motive should be done in a logical, organic way, not haphazardly (improvisation as spontaneous composition) – not, however, in a preconceived way – rather in a way based on intuition enriched with knowledge (from all the study, playing, listening, exposure to various musical styles, etc., that have occurred through a lifetime – including all life experiences); the result is a personal musical vocabulary.*

First, we create a probabilistic module with all the sub-models we want to take into consideration and the corresponding interpolation and smoothing coefficients necessary to the creation of the global probabilistic model. This module can be trained on a substantial corpus offline, but can also be trained (or updated) online with a musician's playing. In Crispell's quote, this matches with the knowledge acquired through the system's lifetime.

Second, we create an oracle factor for which the construction of states, edges and suffix links only depends on one dimension (usually the melody). The states can represent a single note as in OMax or a musical fragment (for instance a beat) as in ImproTek. In Crispell's quote, this correponds to the logic of the context in which the motive must be developed. The oracle is created online with a musician's playing, or with a corpus (usually smaller than the one used to create the probabilistic module).

The system is now able to improvise music, creating a path in the factor oracle that is guided and enriched by the knowledge from the probabilistic module. At each step, knowing the state the system is in, all the reachable states, and the musical contents in those states, we compute a score for each possible transition corresponding to the interpolation of the sub-models in the probabilistic module. Thus, we are enriching with external knowledge the decision of which edge to follow. We can then normalise the scores to obtain the probabilites of transitions and make a random choice following the resulting probabilities.

Let Att(i) be the set of reachable states from state i follow-

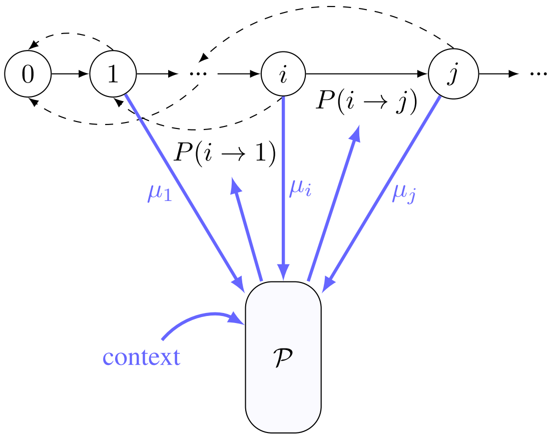

Figure 2. Using a multidimensional probabilistic model $\mathcal{P}$ with an oracle factor. Let us consider that from state i, the only reachable states are state j and state 1. Using the context, μ_1, and μ_i, $\mathcal{P}$ is able to compute a score for the transition from state i to 1. Same thing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

ing the heuristics explained in [16] (using suffix links and reverse suffix links for instance). Let $\mu_i = \{\mu_i^M, \mu_i^C, ...\}$ be the musical contents of state i, that is to say the set of musical variables stored in state i during the oracle construction (for instance, μ_i^M represents the musical content's melody of state i). Then, for all $j \in \text{Att}(i)$, the transition probability in the oracle from state i to state j, knowing the past context is :

$$P(i \to j|X_{1:t}) = \frac{P(M_t = \mu_j^M|X_{1:t})}{\sum\limits_{k \in \text{Att}(i)} P(M_t = \mu_k^M|X_{1:t})} \quad (9)$$

In practice, for $X_{1:t}$, we use the musical contents from the previous and current states of the path of the factor oracle. Figure 2 illustrates this process for one step.

4. EXPERIMENTATION

To test the system proposed in the previous part, we generated some improvisations on Charlie Parker's music following three methods.

1. Some improvisations were made with OMax without any probabilistic module. The factor oracle was constructed on one tune (theme and Parker's improvisation).

2. Some improvisations were made with OMax with a probabilistic module. The sub-models considered are an n-gram model over the melody, and a relational model between melody and harmony. The probabilistic module was trained on Charlie Parker's whole Omnibook (50 themes and improvisations), and the factor oracle was constructed on one tune. The Omnibook corpus was created manually using MusicXML and includes both melodic information and chord labels. The idea here is to have a probabilistic module trained on a larger but similar corpus to the tune used for the factor oracle.

3. Some improvisations were made with OMax with a probabilistic module, similarly to the previous one, but the corpus used to train the probabilistic module is a classical music corpus of over 850 non improvised tunes while the factor oracle is constructed on a Charlie Parker tune (theme and improvisation). The classical music corpus was user-generated using MusicXML with both melodic and chord information and was screened for improper chord labels [12]. The idea here is to see how the system performs when trained on a corpus of a different style than the tune used for the factor oracle.

In the second and third method, the probalistic modules were trained using both melodic and harmonic information over all the tunes of each corpus. Three sub-models were used:

$$P_1(M_n|X_{1:n}) = P(M_n|M_{n-1})$$
$$P_2(M_n|X_{1:n}) = P(M_n|C_n)$$
$$P_3(C_n|X_{1:n}) = P(C_n|C_{n-1})$$

where n is an index over the note of the melody. M_n is the n^{th} notes of the melody, and C_n is the chord played over M_n.

Due to the nature of our dataset, we chose to use a small amount of sub-models and very simple one as a proof of concept. Better results would be expected with more sub-models (as mentioned in 2.1) but would require more complete data.

For each method, 15 improvisations were generated using 3 Charlie Parker tunes as reference : Au Private, Donna Lee and Yardbird Suite.

The generated improvisations can be listened online at `members.loria.fr/evincent/files/smc16` and the MusicXML Omnibook corpus can be found at `members.loria.fr/evincent/files/omnibook`.

First of all, the most significant difference seems to be the harmonic stability appearing while using a probabilistic module trained with either the Omnibook or a classical music corpus. The improvisations generated using these methods seem to follow a harmonic framework, while the factor oracle is only constructed with the melody. For instance, this can be heard on the first example of Au Private. Second, when the probabilistic module is trained on a classical music corpus, while the harmonic stability is stronger, Charlier Parker's musical language looses its distinctiveness, as if the harmonic aspect was too strong a constraint. For instance, this can be noticed on the third example of Yardbird Suite. This comforts our initial idea that using a multidimensional training over an appropriate corpus enables our system to generate improvisations closer to a specific style.

Furthermore, according to listeners, the improvisations with a probabilistic module are more diverse, fluid and creative than the simple oracle one. This is in part because the combination of dimensions and the smoothing provide escape mechanisms from usual mono-dimensional attractors (the obsessive jingle phenomenon due to high conditional probabilities and overfitting). For instance, this can be clearly heard in the first example of Donna Lee.

These results are encouraging. We only tested this system using melodic and harmonic relations, yet we can already hear a significant improvement on how the improvisations are guided through the factor oracle. This system could be extended to represent other interdimensional relations, in particular rhythm, beat phase and dynamic, with more detailed data from live playings, and therefore can be used for any style of music.
Moreover, this system's modularity makes it very adaptable, and could be integrated in other existing systems :

- A probabilistic module could be integrated in ImproTek [5], where the evolution of one dimension is predefined in a scenario. This would add some smoothing in ImproTek's improvisation and therefore expand its expressiveness.

- Similarly, a probabilistic module could be integrated in SoMax [7] where some of the context would come from active listening.

- Finally, this system could be adapted for PyOracle [8] using an interpolation where the dimensions are actually audio features.

5. CONCLUSIONS

We have shown the musical potentialities of the combination of probabilistic models with the factor oracle. This creates a system 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. On the one hand, the probabilistic models enable the system to be trained on a multidimensional sequence and to take the relations between dimensions into account. They also profit from advanced smoothing and optimisation techniques which make them an efficient way to represent the musical knowledge acquired through a lifetime by a musician. On the other hand, the factor oracle is an efficient data structure able to represent the logic of a musical context. This system shows good potential to perform a better navigation in the factor oracle, generating improvisations closer to the desired style. Moreover, this system could be easily adaptated to other existing systems (ImproTek, SoMax, PyOracle...), potentially improving their results.

Acknowledgments

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), and with the support of Region Lorraine.

6. REFERENCES

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

[2] D. Conklin, "Music generation from statistical models," in *Proceedings of the AISB Symp. on Artificial Intelligence and Creativity in the Arts and Sciences*, 2003, pp. 30–35.

[3] F. Pachet and P. Roy, "Markov constraints: steerable generation of Markov sequences," *Constraints*, vol. 16, no. 2, pp. 148–172, March 2011.

[4] S. Dubnov, G. Assayag, O. Lartillot, and G. Bejerano, "Using machine-learning methods for musical style modeling," *IEEE Computer*, vol. 10, no. 38, pp. 73–80, 2003.

[5] J. Nika and M. Chemillier, "Improtek, integrating harmonic controls into improvisation in the filiation of OMax," in *Proceedings of the International Computer Music Conference*, 2012, pp. 180–187.

[6] J. Nika, J. Echeveste, M. Chemillier, and J.-L. Giavitto, "Planning human-computer improvisation," in *Proceedings of the International Computer Music Conference*, 2014, pp. 330–338.

[7] L. Bonasse-Gahot, "An update on the SoMax project," IRCAM, Tech. Rep., 2014.

[8] G. Surges and S. Dubnov, "Feature selection and composition using pyoracle," in *Proceedings of the 2nd International Workshop on Musical Metacreation*, 2013.

[9] A. Donze, S. Libkind, S. A. Seshia, and D. Wessel, "Control improvisation with application to music," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2013-183, November 2013.

[10] D. Conklin and I. H. Witten, "Multiple viewpoint systems for music prediction," *Journal of New Music Research*, vol. 1, no. 24, pp. 51–73, 1995.

[11] R. P. Whorley, G. A. Wiggins, C. Rhodes, and M. T. Pearce, "Multiple viewpoint systems: Time complexity and the construction of domains for complex musical viewpoints in the harmonisation problem," *Journal of New Music Research*, no. 42, pp. 237–266, 2013.

[12] S. A. Raczyński, S. Fukayama, and E. Vincent, "Melody harmonisation with interpolated probabilistic models," *Journal of New Music Research*, vol. 42, no. 3, pp. 223–235, 2013.

[13] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," Harvard University, Tech. Rep. TR-10-98, 1998.

[14] M. I. Bellgard and C. P. Tsang, "Musical networks," N. Griffith and P. M. Todd, Eds. MIT Press, 1999, ch. Harmonizing Music the Boltzmann Way, pp. 261–277.

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

[16] G. Assayag and G. Bloch, “Navigating the oracle : A heuristic approach,” in *Proceedings of the International Computer Music Conference*, 2007, pp. 405–412.

[17] C. Allauzen, M. Crochemore, and M. Raffinot, “Factor oracle : A new structure for pattern matching,” *SOFSEM’99, Theory and Practice of Informatics*, pp. 291–306, 1999.

[18] F. Jelinek and R. L. Mercer, “Interpolated estimation of Markov source parameters from sparse data,” in *Pattern Recognition in Practice*, 1980, pp. 381–397.

[19] D. Klakow, “Log-linear interpolation of language models,” in *Proceedings of the 5th International Conference on Spoken Language Processing*, 1998, pp. 1695–1698.

[20] S. F. Chen, L. Mangu, B. Ramabhadran, R. Sarikaya, and A. Sethy, “Scaling shrinkage-based language models,” in *Proceedings of the Automatic Speech Recognition & Understanding*, 2009, pp. 299–304.

[21] C. Parker and J. Aebersold, *Charlie Parker Omnibook*. Alfred Music Publishing, 1978.

[22] A. Lefebvre and T. Lecroq, “Computing repeated factors with a factor oracle,” in *Proceedings of the 11th Australasian Workshop On Combinatorial Algorithms*, 2000, pp. 145–158.

[23] M. Crispell, “Elements of improvisation,” in *Arcana : Musicians on Music*, J. Zorn, Ed., 2000, pp. 190–192.