SUPERVISED AND UNSUPERVISED SEQUENCE MODELLING FOR DRUM TRANSCRIPTION

Olivier Gillet, Gaël Richard
GET / Télécom Paris (ENST)
CNRS LTCI
37 rue Dareau, 75014 Paris, France

ABSTRACT

We discuss in this paper two post-processings for drum transcription systems, which aim to model typical properties of drum sequences. Both methods operate on a symbolic representation of the sequence, which is obtained by quantizing the onsets of drum strokes on an optimal tatum grid, and by fusing the posterior probabilities produced by the drum transcription system. The first proposed method is a generalization of the N-gram model. We discuss several training and recognition strategies (style-dependent models, local models) in order to maximize the reliability and the specificity of the trained models. Alternatively, we introduce a novel unsupervised algorithm based on a complexity criterion, which finds the most regular and well-structured sequence compatible with the acoustic scores produced by the transcription system. Both approaches are evaluated on a subset of the ENST-drums corpus, and yield performance improvements.

1 INTRODUCTION

Many useful applications can be derived from the knowledge of a semantic description of music signals. As a result, the field of music information retrieval (MIR) is receiving a continuously growing interest from the scientific community. MIR has primarily focused on the extraction of melodic and tonal information, though it is now acknowledged that the rhythmic content, and the drum track in particular, is of primary importance for a number of applications such as drum track remixing, automatic genre recognition, automatic DJing or query by beatboxing.

The problem of drum transcription has already been addressed in several studies (see [1] for a review of existing systems). However, most of the studies on drum track transcription only use short-term acoustic information (as carried in acoustic features, or for example, in the coefficients of a non-redundant decomposition), and thus consider each drum event independently from the other adjacent events. Nevertheless, by analogy with speech, where a sequence of random phonemes does not constitute a syntactically correct sentence, most of the sequences of drum events do not represent musically interesting drum tracks.

In fact, as already shown in previous works [2, 3, 4], the acoustic clues should be combined with sequence models to take into account the structural specificities of drum sequences. Some of these specificities are listed here: some drum subsequences may never be played (either because they are musically irrelevant or because they are too complex to be played by a drummer), some subsequences are frequent, independently of the style (a tom fill for example), and some subsequences are typical of a given style (for example a disco rhythm has the bass drum played on each beat). Furthermore, a drum sequence may contain repetitive patterns that span several hierarchical levels. For instance, a simple one-bar-long pattern may be repeated to create a musical phrase, this phrase may be repeated several times during the chorus, which itself is played several times during the piece.

The aim of this paper is to present two strategies to include such information in drum transcription systems: a supervised strategy, based on a generalization of N-gram models (described in section 3); and an unsupervised strategy (described in section 4) which aims to find the drum sequence that exhibits the largest degree of repetitivity and structure, while still being compatible with the acoustic scores. Both these methods operate on a symbolic representation of the drum sequence. We therefore briefly discuss in section 2 how to obtain such a representation from a list of unquantized onsets and posterior probabilities produced by a drum transcription system. Finally, experimental results are given in section 5, and some conclusions are suggested in the last section.

2 SYMBOLIC REPRESENTATION EXTRACTION

Drum transcription systems output a sequence of pairs $(t_i, \pi_{ij})_{1 \leq i \leq N}$ where π_{ij} expresses the probability that the drum instrument $\mathcal{I}_j$ is played at time t_i [1]. In this study, we focus on three drum instruments: $\mathcal{I}_1 =$ bass drum, $\mathcal{I}_2 =$ snare drum and $\mathcal{I}_3 =$ hi-hat.

[1] In some drum transcription systems, the (t_i) are obtained by an onset detector, and the posterior probabilities (π_{ij}) by probabilistic models trained for each class of drum instrument (bass drum, snare drum...) to detect. Some other systems may require additional post-processings to output a probability score, for example from a detected amplitude or a distance to a template.

2.1 Temporal quantization

For further processing, the detected events should be aligned on the same quantized time basis or grid. An ideal time basis for this alignment is the *tatum*, which is defined as the pulsation that most highly coincides with all note onsets. Several approaches have been proposed to extract the tatum directly from an audio signal or from the inter-onset intervals (IOI). In this work, we used a histogram-based method similar to the one described in [5]. A smoothed histogram of the observed IOIs is considered, and each subdivision of the most frequent IOI is a tatum candidate. Among the candidates, the one whose multiples coincide the most with peaks in the IOI histogram is selected.

Once the tatum τ is estimated, a tatum grid $G(\phi) = \{\phi + i\tau, 0 \leq i \leq \frac{L}{\tau}\}$ can be considered to quantize events. The phase parameter ϕ can be obtained to maximize the coincidences of the grid with note onsets. However, it is also important to adjust each point of the quantization grid to take into account slight tempo changes (ritardandi, accelerandi), swing, and the limited temporal resolution of the tatum estimation algorithm. In this paper, a simple approach based on dynamic programming is followed, to optimize the individual position of each tatum event, in an interval around the initial position. We further denote τ_n the position of the n-th point on the tatum grid.

2.2 Temporally aligned acoustic scores

The next step consists in representing the rhythmic sequence as a set of symbols $S_n \in A$, each of them mapped by ϕ to a combination of drum instruments played at time τ_n of the tatum grid. $\phi(S_n)$ can be any subset of the set $\mathcal{A} = \{\text{bass drum, snare drum, hi-hat}\}$ of drum instruments of interest.

Let T_n be the set of indices of the onsets t_i whose closest tatum point is τ_n (i.e. $T_n = \{i, n = \arg\min_k |\tau_k - t_i|\}$). For a given symbol S denoting a combination $\phi(S) \subset \mathcal{A}$ of drum instruments, the probability that this combination is played at τ_n is given by:

$$P(S_n = S|t, \pi) = \prod_j \begin{cases} 1 - \prod_{i \in T_n} \overline{\pi}_{ij} \text{ if } \mathcal{I}_j \in \phi(S) \\ \prod_{i \in T_n} \overline{\pi}_{ij} \text{ if } \mathcal{I}_j \notin \phi(S) \end{cases} \tag{1}$$

where $\overline{\pi}_{ij} = (1 - \pi_{ij})$. For example, the probability that the symbol denoting the combination $\{sd, hh\}$ is played at time τ_n is calculated as the probability that there is at least one snare drum and one hi-hat strokes detected in the interval related to τ_n and that there is no bass drum stroke detected in this same interval.

As a result of the processes described in this section, the output of the drum transcription system can be represented as a sequence of tatum pulsations τ_n, and probabilities $P(S_n = S)$; a candidate transcription being represented as a sequence s_n. Using only the acoustic clues, the most likely transcription is simply:

$$s_n^* = \arg\max_{s \in A} P(S_n = s)$$

In the next section, we propose several means of combining these acoustic cues with sequence models. We also consider, in section 4, a different decision rule which includes a regularization term penalizing complex transcriptions.

3 SUPERVISED SEQUENCE MODELLING

3.1 Generalized N-gram models

Several techniques have been proposed in the past to describe the dependencies of symbols S_n in a drum sequence. The most straightforward approach (as followed in [6]) is the traditional N-gram model, where the probability to observe a sequence $s = (s_n)_{1 \leq n \leq L}$ is:

$$P(s) = \prod_{n=1}^{L} P(s_n | s_{n-1} \dots s_{n-N+1}) \tag{2}$$

A different approach was proposed in [3] where the dependencies are tracked at the level of successive bars. In this periodic N-gram model, the probability to observe a sequence s is given by:

$$P(s) = \prod_{n=1}^{L} P(s_n | s_{n-M} \dots s_{n-(N-1)M}) \tag{3}$$

In this paper we propose a generalization of the previous approaches by considering dependencies at different scales. Let $\mathcal{T}$ be the sequence model time support – in other words, the temporal levels at which the dependencies are considered. Following this model, the probability to observe a sequence s is then equal to:

$$P(s) = \prod_{n=1}^{L} P(s_n | s_{n-\mathcal{T}_1} \dots s_{n-\mathcal{T}_N}) \tag{4}$$

Note that the traditional N-gram model corresponds to the case where $\mathcal{T} = (1, 2, \dots, N-1)$ and that the periodic N-gram to the case where $\mathcal{T} = (M, 2M, \dots, (N-1)M)$. As such, our model allows the use of a time support $\mathcal{T}$ that achieves a trade-off between the observation length and the number of probabilities to estimate. For instance, when the tatum corresponds to a sixteenth note, with a $\binom{4}{4}$ time signature, the choice $\mathcal{T} = (1, 4, 16)$ allows the model to learn dependencies between successive bars, beats, and tatum points, while keeping the overall complexity and the number of degrees of freedom of the model low.

3.2 Probabilities estimation

The learning of sequence models consists in estimating the probabilities of observing a given symbol s_n knowing its context. These probabilities can simply be estimated by counting in a training corpus the number of times s_n is observed in a given context normalized by the number of times this context is encountered. This simple approach is unable to deal with infrequent or unseen substrings in the training corpus. In this work, we used Witten-Bell smoothing [7] to estimate the probabilities in such cases.

3.3 Most likely sequence

Given the probabilities $P(S_n = S|t,\pi)$ (noted $P(S|t,\pi)$ for sake of clarity), and a generalized N-gram model of time support $\mathcal{T}$, the most likely sequence is:

$$\arg\max_{s} \prod_{1\leq n\leq L} P(s_n|t,\pi)P(s_n|s_{n-\mathcal{T}_1}\dots s_{n-\mathcal{T}_N}) \quad (5)$$

This optimal sequence can be approximated with a causal greedy search ($\mathcal{O}(L|A|)$ complexity) or the non-causal Viterbi algorithm ($\mathcal{O}(L|A|^2)$ complexity). It can also be found exactly by a Viterbi search through the space of all contexts ($\mathcal{O}(L|A|^{\mathcal{T}_N+1})$ complexity), where $|A|$ is the number of rhythmic symbols.

3.4 Training sequence models: What to learn?

So far, we considered that a training corpus is available to estimate the probabilities $\hat{P}(s_n|s_{n-\mathcal{T}_1}\dots s_{n-\mathcal{T}_N})$. An open question is the choice of this training corpus. Here, we discuss several strategies, and evaluate the predictive power of the models that can be obtained by following them.

Generic model The training corpus consists in a set of heterogenous sequences of different styles played by different drummers. This approach is the simplest to follow: one unique model has to be trained once and for all.

Drummer-dependent model The training corpus contains only sequences played by the same drummer as the sequences to transcribe. Such an approach is only feasible for a few limited applications – where the system can be calibrated to a given performer.

Style-dependent model The training corpus contains only sequences of a given style (e.g. salsa, reggae or rock) played by various drummers. Thus, a different model is trained for each style. In order to select the model to be used for the recognition, several approaches are possible: an independent style classification system can be used (hierarchical classification), a human user with infallible skills can select the model corresponding to the sequence (classification with style oracle), or the recognition can be performed in parallel by each model, the final result being the one produced by the model having the highest likelihood.

Individual sequence model We assume, in this case, that the sequence to be transcribed is known in advance, and that we can estimate the probabilities from this sequence. This approach is of limited interest, except for applications like computer aided teaching of drumming, or score following, where the score is known in advance. A more feasible solution consists in using an initial model (for example, a generic model) to obtain a first transcription; and then to train a local individual sequence model on the recognized sequence. Assuming that the errors made by the transcription system are independent of the context, the probabilities estimated from the recognized sequence will be close to those estimated on the correct sequence. The recognition is then performed using this local model.

These four approaches are evaluated by comparing the predictive power of the learned sequence models, which is measured by the mutual information between a rhythmic symbol s and its context c (this context being defined by the support $\mathcal{T}$):

$$I(\mathrm{C},S) = \sum_{c\in A^{N-1}} \sum_{s\in A} P(\mathrm{c}\ s) \log_{|A|} \frac{P(\mathrm{c}\ s)}{P(\mathrm{c})P(s)} \quad (6)$$

Since $I(\mathrm{C},S) = H(S) - H(S|\mathrm{C})$, the mutual information measures the certainty with which a symbol is determined, knowing its context. A null value implies that the context has no predictive power on the observed symbol. The results obtained on the different training corpora (Refer to section 5 for a description of the database) with different time supports $\mathcal{T}$ are summarized in table 1.

$\mathcal{T}$	Generic	Drummer	Style	Sequence
Generalized trigrams				
-2,-1	0.153	0.237	0.357	0.405
-4,-1	0.157	0.237	0.347	0.396
-8,-1	0.192	0.262	0.359	0.403
-16,-1	0.185	0.254	0.348	0.391
-4,-2	0.179	0.253	0.356	0.398
-8,-2	0.204	0.265	0.353	0.390
-16,-2	0.213	0.273	**0.370**	**0.407**
-8,-4	0.219	0.279	0.354	0.392
-16,-4	0.196	0.254	0.344	0.380
-16,-8	**0.229**	**0.283**	0.348	0.379
-32,-16	0.208	0.264	0.325	0.361
Generalized quadrigrams				
-3,-2,-1	0.281	0.414	0.523	0.552
-4,-2,-1	0.297	**0.429**	0.528	0.555
-8,-2,-1	0.307	**0.429**	**0.531**	**0.558**
-16,-8,-1	0.311	0.423	0.517	0.546
-8,-4,-2	0.318	0.428	0.515	0.540
-16,-4,-2	0.308	0.418	0.525	0.551
-16,-8,-2	**0.322**	0.423	0.514	0.541
-16,-8,-4	0.312	0.408	0.500	0.526
-48,-32,-16	0.309	0.403	0.470	0.504

Table 1. Predictive power of the sequence model, measured by the mutual information between a symbol and its context $I(\mathrm{C},S)$, for different time supports $\mathcal{T}$ and training corpora (all of them subsets of the ENST-drums database).

Firstly, the results show that drummer dependent models are only slightly more efficient than generic models. Such models have a lower predictive power than style-dependent models. Secondly, the efficiency of style-dependent models, and their predictive power close to the one of individual sequence models, suggests that sequences played in a specific style are rather homogeneous and played with a limited degree of variability. Finally the results highlight the usefulness of the generalized N-grams introduced: in fact, they are, in most cases, more efficient than traditional N-grams and pure periodic N-grams since they can simultaneously track short and long term dependencies. How-

ever, it is worth noting that we have only evaluated the predictive power of the sequence model here. Such models might not offer any improvement in a full transcription system, as their efficiency depends on the reliability of the acoustic scores.

4 UNSUPERVISED SEQUENCE MODEL

Although powerful, the previously described supervised approach suffers from two main drawbacks. On the one hand, it needs a training phase for which a trade-off between genericity and predictive power needs to be found. On the other hand, if higher performances can be obtained by a generalized N-gram approach, the choice of the time support $\mathcal{T}$ requires prior knowledge of the duration of a bar or a sequence; and when N is large, the accuracy of the estimated probabilities is poor. As a consequence, a novel alternative approach, entirely unsupervised, is proposed in this section.

This approach is based on the following assumption: drum sequences are built with rather regular and repetitive patterns at different time scales. The basic idea for our approach is thus to correct (or rather simplify) the drum sequence obtained by the transcription system so that it can be more easily described in terms of hierarchical, repetitive patterns.

4.1 Complexity criterion

The Kolmogorov complexity of a sequence $K(S)$ is defined as the length of the shortest program, represented with a binary alphabet for a given model of computation (for example, a universal Turing Machine), which outputs the sequence S. $K(S)$ is not computable, but can be approximated by compression algorithms. In this case, the shortest program generating S is a compressed version of S followed by a program decompressing it.

Complexity criteria have been used in [8] and [9] to measure the similarity between melodies; and in [10] to detect the melody in a polyphonic piece (the main melody is considered to be the part of maximal complexity). All these studies use the LZ77 or LZ78 [11] compression algorithms as an approximation of Kolmogorov complexity.

Here, we propose to use a different compression algorithm to measure the complexity of rhythmic sequence: the SEQUITUR algorithm [12]. First, this algorithm outperforms LZ78 for various text compression tasks, and thus, gives a better approximation of the minimal description length. Moreover, this algorithm infers from the observed sequence, not a dictionary of frequent prefixes (as is the case with LZ78), but a context-free grammar, which thus takes into account the hierarchical and recursive structure of the sequence. Finally, this algorithm can be easily modified to include domain-specific transformation operators (transposition, time-reversal, etc.) in the inferred grammar.

4.1.1 Inferring a grammar from a sequence

We recall here the principle of the SEQUITUR algorithm, which processes the sequence, symbol by symbol, from left to right, to update its representation as a context-free grammar G verifying the following two properties:
Bi-gram uniqueness a bigram should not appear more than once [2] in the right member of a production rule. Two cases are possible:

- when G contains the rules $A \rightarrow XabY$ and $B \rightarrow ZabT$, a new rule $C \rightarrow ab$ is created and original rules are modified as $A \rightarrow XCY$ and $B \rightarrow ZCT$.
- when G contains the rules $A \rightarrow XabY$ and $B \rightarrow ab$, the first rule is modified into $A \rightarrow XBY$.

Usefulness of a rule. Each production rule should be used at least twice. Thus, if the grammar contains $A \rightarrow XBY$ and $B \rightarrow ZT$, and if the non-terminal B only appears in the first rule, the second rule is deleted and the first rule becomes $A \rightarrow XZTY$.

As an example, the following grammar will be inferred from the sequence $abcbcabcbc$:

$$S \rightarrow AA \mid A \rightarrow aBB \mid B \rightarrow bc$$

4.1.2 Complexity from SEQUITUR

A context-free grammar is entirely defined by the string obtained by joining the right members of the production rules with a separation marker noted $\#$. For example, the grammar given in the previous example is entirely defined by the string $AA\#aBB\#bc$. If an entropic code (e.g. a Huffman code) is used to compress this sequence, an approximation of the length of the corresponding binary message is given by:

$$l(G) \approx \sum_{a \in \Omega} -C(a) \log_2 \frac{C(a)}{N} \tag{7}$$

where $\frac{C(a)}{N}$ is the frequency of symbol a in the sequence, N the length of the sequence and Ω the alphabet of symbols. The whole procedure for the approximation of the complexity of a rhythmic sequence is summarized below:

1. Inference of a context-free grammar $G(s)$ from the sequence s with SEQUITUR.
2. Reduction of the grammar production rules into a string.
3. Compression of this string into a binary message of length $l(G(s))$ with an entropic code.

[2] Although it is not used in this work, it is possible to generalize this rule to include bijective transformations in the production rules (such rules would be of the form $A \rightarrow \varphi(B)C$, where φ is the transformation). In the context of note sequences, useful transformations that can be considered include transposition or reversal. In the context of drum sequences, a useful transformation is the substitution of one cymbal by another – for example, a sequence can be repeated with the ride cymbal played instead of the closed hi-hat. In this work, we only consider the hi-hat, and thus do not use such rules.

We observed in our database that the average complexity of the original (ground-truth annotation) drum sequences is 984 bits per sequence, while the average complexity of the corresponding transcriptions [3] is 1179 bits per sequence. This supports our initial assumption that the errors made by the transcription system break the structure and repetitiveness of the drum sequences.

4.2 Penalized likelihood for sequences

We propose the following penalized likelihood criterion in order to find the sequence that is both simple, according to the previously defined complexity measure, and compatible with the acoustic scores:

$$s^* = \arg\max_s F(s) \quad (8)$$

$$F(s) = \sum_{n=1}^{L} \log P(S_n = s_n|t,\pi) - \alpha l(G(s)) \quad (9)$$

The first term requires the sequence to be compatible with the acoustic score, and the second term penalizes complex sequences. Unfortunately, there exists to our knowledge no deterministic algorithm to find s^* (above all, contrary to the model we described in section 3, dynamic programming cannot be used), and a search in the space of all possible sequences is obviously intractable. Genetic algorithms appear as an interesting approach to solve this problem, especially because in our case, there exists a trivial representation of the parameter to optimize as "chromosomes", for which the cross-over operator makes sense: a good transcription is likely to be obtained by combining parts of good transcriptions. The procedure is described below :

Initialization of a population of $N_{pop} = 200$ sequences (s^i). This population is initialized with mutations of the best sequence obtained without the complexity penalization term, i.e. $\arg\max_s \sum_{n=1}^{L} \log P(S_n = s_n|t,\pi)$.

Reproduction $N_{exp} = 4N_{pop}$ children sequences are produced by the following procedure:

1. Random choice of two parents s^1 and s^2 amongst the current population.
2. Crossing-over. A recombination point $p \in [1, L]$ is randomly chosen. The child sequence is then determined by $s^c(n) = s^1(n), \forall n \in [1, p]$ and $s^c(n) = s^2(n), \forall n \in [p+1, L]$.
3. Mutation. A mutation position $p \in [1, L]$ is randomly chosen. The probability that the symbol at position p mutates into a is given by the acoustic score $P(S_p = a|t,\pi)$.

Selection. A population of N_{pop} sequences survives. This population contains the $0.9N_{pop}$ sequences for which the criterion F is the largest and $0.1N_{pop}$ sequences randomly selected amongst the remaining sequences.

[3] Produced by a system described in yet unpublished work.

	BD	SD	HH	BD	SD	HH
	Baseline			Unsupervised		
	79.4	59.6	76.7	81.3	61.7	80.4
$\mathcal{T}$	Indiv. sequence with sequence oracle			Style-dep. model with style oracle		
-1	**82.6**	63.3	79.2	79.4	60.4	78.0
-2,-1	82.0	**67.0**	80.6	80.2	60.9	79.7
-4,-1	81.7	64.6	80.9	80.5	61.2	79.5
-8,-1	82.3	63.8	80.3	**81.2**	**61.9**	79.1
-16,-1	81.0	63.2	80.2	80.2	60.2	78.8
-3,-2,-1	80.9	66.7	81.5	78.1	61.8	**80.3**
-4,-2,-1	82.2	65.7	82.5	78.5	60.1	79.3
-8,-2,-1	81.2	65.4	81.2	78.7	59.8	79.3
-16,-2,-1	82.4	66.0	82.1	78.8	59.2	78.6
-4,-3,-2,-1	78.7	66.0	**82.9**	76.4	61.5	**80.3**
-16,-8,-2,-1	81.4	64.7	81.3	79.3	59.0	79.4
	with local model			with most likely style		
-1	80.8	60.2	77.9	79.4	60.4	78.0
-2,-1	81.3	60.6	**78.2**	80.2	60.8	79.6
-4,-1	81.2	**61.2**	77.6	80.9	60.9	78.7
-8,-1	81.0	60.9	77.8	**81.2**	61.4	78.8
-16,-1	81.2	60.1	77.7	80.1	60.1	78.6
-3,-2,-1	81.3	60.8	77.2	78.1	**61.8**	**80.3**
-4,-2,-1	**81.6**	61.1	77.6	77.4	59.8	78.8
-8,-2,-1	81.5	61.1	77.7	77.2	59.9	78.6
-16,-2,-1	**81.6**	60.8	77.2	78.5	59.1	78.2
-4,-3,-2,-1	81.1	61.0	77.5	75.4	61.6	80.0
-16,-8,-2,-1	81.5	60.1	76.5	79.3	59.0	79.1

Table 2. Performance of the supervised sequence models for several contexts and training strategies, and of the unsupervised error correction method

Iteration on $N = 50$ generations.

A specificity of this implementation is the control of the mutation probabilities. This trick avoids the exploration of regions of the solution space where the likelihood term is too low. Actually, we observed that even when $\alpha >> 1$ (i.e., when the regularization term dominates the likelihood term), the solutions have a high likelihood. We used $\alpha = 0.5$ in the following experiments.

5 EXPERIMENTAL RESULTS

The experiments were conducted on the *minus one* sequences of the ENST-Drums corpus [13], with a balanced mix between drums and musical accompaniment to recreate realistic use conditions. This corpus consists of 17 sequences, of various genre, played by 3 drummers – we selected for these experiments the sequences played by the second and third drummers. For each instrument, the performance is measured by the F-measure. Results are given in table 2. The baseline corresponds to the raw output of the drum transcription system, without sequence modeling.

First of all and not surprisingly, we observe that the greater gains are obtained by using an individual sequence model with oracle – that is to say, by using a sequence model trained in advance on the sequence to be recog-

nized. Interestingly, the best performances for each instrument are achieved by considering a different context: the hi-hat requires long contexts, while the snare drum and bass drum require shorter context. The best results are obtained with generalized quadrigrams taking into account the local context, and the event played 4 or 8 tatum points earlier. The performance gain offered by the local model is lower – this is probably due to the lack of training data for the adapted model.

The performance gains offered by the style-dependent models (be it with automatic style detection, or with an oracle indicating the style in which the sequence is performed) are very similar. Actually, the performance of the style identification stage (which consists in selecting the style-dependent model with the largest likelihood) is rather satisfying – the model corresponding to the style in which the sequence was played was selected 70% of the time. We observed that even when the wrong model was selected, some errors were eliminated in the transcription, as a model from a different style still carried general properties of drum sequences.

The unsupervised method based on complexity minimization performed similarly as style-dependent models. However, its usefulness is hampered by its large computational cost.

6 CONCLUSION AND FUTURE WORK

We described in this work two methods to improve the output of drum transcription systems by modeling several typical properties of drum sequences. Firstly, a supervised method based on a generalization of the N-gram models is described. We particularly focused on the selection of the corpus on which such models should be trained, and we proposed several strategies and classification schemes to efficiently use such models, since we observed that their predictive power is limited on too diverse corpora. Secondly, an unsupervised method based on a complexity criterion is introduced. This criterion favors sequences which exhibit a well-defined structure – more precisely, which can be described by a compact context-free grammar. However, there is, to our knowledge, no efficient way to maximize this criterion. In this work, we used genetic algorithms with controlled mutation rates as a heuristic to efficiently explore the space of possible drum sequences. Future work should address the problem of finding more efficient algorithms or heuristics to minimize this criterion. Even if we are pessimistic about the existence of a solution of polynomial complexity, the use of such complexity penalization terms could benefit to other MIR applications such as music transcription.

Our experimental results showed performance gains for the two methods. However, these gains remain modest. We suggest that this result is not due to the sequence models themselves, but rather to the lack of reliability of the acoustic scores produced by the transcription system. Our initial assumption that classifiers or detectors produce posterior probabilities close to the decision boundary, but on the wrong side, when they encounter a difficult case, is wrong: we rather observed that most of the mistakes made by classifiers indeed corresponded to posterior probabilities far from the decision boundary. It is thus very likely that the performance of sequence models is bounded by the performance of the detection or classification module.

7 REFERENCES

[1] D. FitzGerald and J. Paulus. Unpitched percussion transcription. In A. Klapuri and M. Davy, editors, *Signal processing methods for the automatic transcription of music*, pages 131–162. Springer, 2006.

[2] O. Gillet and G. Richard. Automatic labelling of Tabla signals. In *Proc. ISMIR'03*, October 2003.

[3] J. Paulus and A. Klapuri. Conventional and periodic n-grams in the transcription of drum sequences. In *Proc. ICME'2003*, 2003.

[4] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. Okuno. An error correction framework based on drum pattern periodicity for improving drum sound detection. In *Proc. ICASSP'06*, May 2006.

[5] C. Uhle and J. Herre. Estimation of tempo, micro time and time signature from percussive music. In *Proc. DAFX'03*, September 2003.

[6] O. Gillet and G. Richard. Automatic transcription of drum loops. In *Proc. ICASSP'04*, May 2004.

[7] I. H. Witten and T. C. Bell. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Trans. Inf. Theory*, 37(4):1085–1094, 1991.

[8] R. Cilibrasi, P. Vitanyi, and R. De Wolf. Algorithmic clustering of music based on string compression. *Computer Music Journal*, 28(4):49–67, 2004.

[9] M. Li and R. Sleep. Melody classification using a similarity metric based on kolmogorov complexity. In *Proc. of the 2nd Conf. on Sound and Music Computing*, 2005.

[10] S. T. Madsen and Gerhard Widmer. Music complexity measures predicting the listening experience. In *Proc. 9th Int. Conf. Music Perception and Cognition*, 2006.

[11] J. Ziv and A. Lempel. Compression of individual sequences via variable-rate coding. *IEEE Trans. Inf. Theory*, 24(5):530–536, September 1978.

[12] C. G. Nevill-Manning, I. H. Witten, and D. L. Maulsby. Compression by induction of hierarchical grammars. In *Proc. of the Data Compression Conf.*, pages 244–253, 1994.

[13] O. Gillet and G. Richard. ENST-drums: an extensive audio-visual database for drum signals processing. In *Proc. ISMIR'06*, 2006.