DISCRIMINATING SYMBOLIC CONTINUATIONS WITH GENDETECT

Jeff Ens
Simon Fraser University
jeffe@sfu.ca

Philippe Pasquier
Simon Fraser University
pasquier@sfu.ca

ABSTRACT

We describe GenDetect, an algorithm that was submitted to the 2019 MIREX Patterns for Prediction task. GenDetect is used to discriminate between two possible continuations of a prime, distinguishing a genuine continuation from a generated one. Each musical excerpt is represented by a collection of categorical distributions and a Gradient Boosting Classifier is trained to predict the genuine continuation using this representation. Two versions of the algorithm were submitted, one for polyphonic music and another for monophonic music.

1. INTRODUCTION

The 2019 MIREX Patterns for Prediction task consists of two sub-tasks. GenDetect is designed for the second subtask, which involves discriminating between two continuations $(\mathbb{M}_a, \mathbb{M}_b)$ of a prime $\mathbb{M}_{\text{prime}}$. In contrast to *BachProp* [1], which ranks $\mathbb{M}_a$ and $\mathbb{M}_b)$ according to their probability under a recurrent neural network that is trained to model musical material, we train a Gradient Boosting Classifier [3] to predict the genuine continuation given a feature-based representation of $\mathbb{M}_a$, $\mathbb{M}_b$ and $\mathbb{M}_{\text{prime}}$. We build on the data representation used by *StyleRank*, representing each musical excerpt (e.g. $\mathbb{M}_a$) with a collection of categorical distributions [2].

2. METHODOLOGY

In what follows we adopt the following notation. Given a set x, $||x||$ denotes the number of elements in the set x, and x_i denotes the i^{th} element in x (1-indexed). $\max(x)$ and $\min(x)$ denote the maximum and minimum elements in x respectively. $\lll$ indicates a left bitwise shift. $\mathbf{I}(\cdot)$ is a function that returns 1 if the predicate $\cdot$ is true and 0 otherwise. Let $\oplus$ denote the concatenation operation.

2.1 Data Representation

Consider a musical excerpt $\mathbb{M} = [m_1, ..., m_n]$, consisting of n notes (m_i) ordered lexicographically, sorting first by onset and then by pitch height. Let $\mathbb{P} = [\texttt{pitch}(m_i) : 1 \leq i \leq n]$ [1], $\mathbb{O} = [\texttt{qnt}(\texttt{ons}(m_i)) : 1 \leq i \leq n]$, and $\mathbb{D} = [\texttt{qnt}(\texttt{dur}(m_i)) : 1 \leq i \leq n]$, where `pitch`, `ons` and `dur` are functions returning the pitch, onset and duration of a note respectively. `qnt` refers to Eq. (1), which accepts time-based values (e.g. onset and duration) and returns an integer rounded to the nearest r^{th} subdivision of a beat.

$$\texttt{qnt}(x) = \lceil xr - 0.5 \rceil \tag{1}$$

The following procedure is applied to segment $\mathbb{M}$ into chords, where $\texttt{off}(m_i) = \texttt{qnt}(\texttt{ons}(m_i) + \texttt{dur}(m_i))$. First we construct two sets, one containing all unique note onsets $\mathbb{B}_{\text{onset}} = \{\texttt{ons}(m_i) : m_i \in \mathbb{M}\}$ and another containing all unique note offsets $\mathbb{B}_{\text{offset}} = \{\texttt{off}(m_i) : m_i \in \mathbb{M}\}$. Then we construct the ordered set $\mathbb{B} = \mathbb{B}_{\text{onset}} \cup \mathbb{B}_{\text{offset}}$, where the elements are arranged in ascending order. The i^{th} chord is the set of notes that completely overlap the interval $[\mathbb{B}_i, \mathbb{B}_{i+1}]$, and can be calculated using Eq. (2). As a result, there are $||\mathbb{B}|| - 1$ chords in $\mathbb{M}$, and rests are equivalent to chords containing no notes $(\mathbb{C}_i = \emptyset)$. In what follows, let $\mathbb{C}_i^j$ denote the j^{th} note in the i^{th} chord, and $\psi(\mathbb{C}_i) = \{\texttt{pitch}(\mathbb{C}_i^j) : \mathbb{C}_i^j \in \mathbb{C}_i\}$. In addition , we sort the notes in each chord in ascending order according to pitch height.

$$\mathbb{C}_i = \{n : (n \in \mathbb{M}) \wedge (\texttt{ons}(n) \leq \mathbb{B}_i) \wedge (\texttt{off}(n) \geq \mathbb{B}_{i+1})\} \tag{2}$$

We use distinct pitch class sets (PCD) [2] to represent pitched material, which reduces the $2^{12} = 4096$ possible pitch class sets to 352 equivalence classes, grouping pitch class sets that are transpositionally equivalent. For example, the pitch class sets $\{0, 4, 7\}$ and $\{2, 5, 10\}$ are transpositionally equivalent, as both are major chords, the only difference being their root. $\texttt{PCD}(\cdot)$ is a function that accepts a pitch class set and returns an integer corresponding to the PCD. For more details on calculating the PCD, see the original paper [2].

We represent each musical excerpt $(\mathbb{M})$ by applying a non-empty set of feature transformations $\mathcal{F} = \{f_1, ..., f_d\}$, producing a set of categorical distributions $\mathcal{F}^{\mathbb{M}} = \{f_1^{\mathbb{M}} ..., f_d^{\mathbb{M}}\}$. A categorical distribution is a discrete probability distribution describing a random variable that has k possible distinct states. Concretely, $f_i^{\mathbb{M}} = [f_i(k) : a \leq k < b]$, where a and b are the upper and lower bounds

[1] Note that we adapt the set-builder notation to construct a list (e.g., $[i/2 : 0 \leq i < 4] = [0, 0, 1, 1]$), which unlike a set, may contain duplicate values, and has a specific order.

	Feature Name	Function	Domain
Mono.	Chord Size ⋆	$\sum_{i=1}^{\|\|\mathbb{B}\|\|-1} \mathbf{I}_k(\|\|\mathbb{C}_i\|\|)(\mathbb{B}_{i+1} - \mathbb{B}_i)$	[0,2)
Mono.	Melodic n-gram PCD	$\sum_{i=1}^{\|\|\mathbb{P}\|\|-w+1} \mathbf{I}_k(\texttt{PCD}(\{\mathbb{P}_j \mod 12 : i \leq j < i+w\}))$	[0,352)
Both	Note Duration	$\sum_{i=1}^{\|\|\mathbb{D}\|\|} \mathbf{I}_k(\mathbb{D}_i)$	[0,16r)
Both	Note Duration Difference	$\sum_{i=1}^{\|\|\mathbb{D}\|\|-1} \mathbf{I}_k(\mathbb{D}_{i+1} - \mathbb{D}_i + 16r)$	[0,32r)
Both	Note Offset	$\sum_{i=1}^{\|\|\mathbb{O}\|\|} \mathbf{I}_k((\mathbb{O}_i + \mathbb{D}_i) \mod 16R)$	[0,16R)
Both	Note Onset	$\sum_{i=1}^{\|\|\mathbb{O}\|\|} \mathbf{I}_k(\mathbb{O}_i \mod 4R)$	[0,4r),
Both	Note Onset Difference	$\sum_{i=1}^{\|\|\mathbb{O}\|\|-1} \mathbf{I}_k(\mathbb{O}_{i+1} - \mathbb{O}_i)$	[0,16r)
Both	Pitch Interval	$\sum_{i=1}^{\|\|\mathbb{P}\|\|-1} \mathbf{I}_k(\mathbb{P}_{i+1} - \mathbb{P}_i + 128)$	[0,256)
Polyphonic	Chord Duration ⋆	$\sum_{i=1}^{\|\|\mathbb{B}\|\|-1} \mathbf{I}(\|\|\mathbb{C}_i\|\| > 0)\mathbf{I}_k(\mathbb{B}_{i+1} - \mathbb{B}_i)$	[0,16r)
Polyphonic	Chord Jaccard Distance	$\sum_{i=1}^{\|\|\mathbb{B}\|\|-2} \mathbf{I}_k\Big(\Big\lceil (d-1)\frac{\|\|\psi(\mathbb{C}_i) \cap \psi(\mathbb{C}_{i+1})\|\|}{\|\|\psi(\mathbb{C}_i) \cup \psi(\mathbb{C}_{i+1})\|\|} - 0.5\Big\rceil\Big)$	[0,d)
Polyphonic	Chord Onset	$\sum_{i=1}^{\|\|\mathbb{B}\|\|-1} \mathbf{I}_k\big(\sum_{j=1}^{\|\|\mathbb{C}_i\|\|}(1 \ll j)\mathbf{I}(\texttt{ons}(\mathbb{C}_i^j) = \mathbb{B}_i)\big)$	[0,352)
Polyphonic	Chord Onset ⋆	$\sum_{i=1}^{\|\|\mathbb{B}\|\|-1} \mathbf{I}_k\big(\sum_{j=1}^{\|\|\mathbb{C}_i\|\|}(1 \ll j)\mathbf{I}(\texttt{ons}(\mathbb{C}_i^j) = \mathbb{B}_i)\big)(\mathbb{B}_{i+1} - \mathbb{B}_i)$	[0,352)
Polyphonic	Chord Onset Difference	$\sum_{i=1}^{\|\|\mathbb{B}\|\|-1} \mathbf{I}_k(\mathbb{B}_{i+1} - \mathbb{B}_i + 128)$	[0,256)
Polyphonic	Chord Onset PCD ⋆	$\sum_{i=1}^{\|\|\mathbb{B}\|\|-1} \mathbf{I}_k\big(\texttt{PCD}(\{\texttt{pitch}(x) \mod 12 : (x \in \mathbb{C}_i) \wedge (\texttt{ons}(x) = \mathbb{B}_i)\})\big)(\mathbb{B}_{i+1} - \mathbb{B}_i)$	[0,352)
Polyphonic	Chord Outer Interval	$\sum_{i=1}^{\|\|\mathbb{B}\|\|-1} \mathbf{I}_k\big((\max(\psi(\mathbb{C}_i)) - \min(\psi(\mathbb{C}_i))) \mod 12\big)$	[0,12)
Polyphonic	Chord PCD	$\sum_{i=1}^{\|\|\mathbb{B}\|\|-1} \mathbf{I}_k\big(\texttt{PCD}(\{\texttt{pitch}(x) \mod 12 : x \in \mathbb{C}_i\})\big)$	[0,352)
Polyphonic	Chord PCD ⋆	$\sum_{i=1}^{\|\|\mathbb{B}\|\|-1} \mathbf{I}_k\big(\texttt{PCD}(\{\texttt{pitch}(x) \mod 12 : x \in \mathbb{C}_i\})\big)(\mathbb{B}_{i+1} - \mathbb{B}_i)$	[0,352)
Polyphonic	Chord Size ⋆	$\sum_{i=1}^{\|\|\mathbb{B}\|\|-1} \mathbf{I}_k(\|\|\mathbb{C}_i\|\|)(\mathbb{B}_{i+1} - \mathbb{B}_i)$	[0,12)
Polyphonic	Note Pitch	$\sum_{i=1}^{\|\|\mathbb{P}\|\|} \mathbf{I}_k(\mathbb{P}_i)$	[0,128)

Table 1. Formal definitions for the feature transformations used by monophonic, polyphonic and both models. $\star$ denotes feature transformations that are weighted by chord duration, using the term $(\mathbb{B}_{i+1} - \mathbb{B}_i)$. The domain $[a, b)$ sets the bounds of the categorical distribution. In our implementation, we set $d = 25$ and $r = 8$.

of the domain respectively. The categorical distributions in $\mathcal{F}^{\mathbb{M}}$ are concatenated, resulting in a single vector representing $\mathbb{M}$, which we refer to as $\mathbf{v}^{\mathcal{F},\mathbb{M}}$. Table 1 provides formal definitions for each of the feature transformations (f_i), and specifies the domain used to construct the corresponding categorical distribution $(f_i^{\mathbb{M}})$. Note that in some cases, the domain is dependant on the number of subdivisions per beat (r). Let $\mathbf{I}_k(\cdot)$ be a function that returns 1 if $\cdot = k$ and 0 otherwise.

2.2 Training

Given a prime $\mathbb{M}_{\text{prime}}$, and two possible continuations ($\mathbb{M}_a$, $\mathbb{M}_b$), we train a Gradient Boosting Classifier [3] to predict whether $\mathbb{M}_a$ or $\mathbb{M}_b$ is the genuine continuation given $\mathbf{v}^{\mathbb{M}_{\text{prime}}} \oplus \mathbf{v}^{\mathbb{M}_a} \oplus \mathbf{v}^{\mathbb{M}_b}$ as input. Concretely, the classifier is trained to output a 0 if $\mathbb{M}_a$ is the genuine continuation and 1 otherwise. Notably, we were able to attain the same level of accuracy by training a Gradient Boosting Classifier to output 1 if the continuation $(\mathbb{M}_x)$ is genuine and 0 otherwise given $\mathbf{v}^{\mathbb{M}_{\text{prime}}} \oplus \mathbf{v}^{\mathbb{M}_x}$ as input.

The code was implemented in Python using the scikit-learn module [4]. Notably, a model can be trained on $10,000$ training examples in several minutes on an Intel Core i7-9700, which is much faster than training *BachProp*.

3. ACKNOWLEDGMENTS

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

4. REFERENCES

[1] Florian Colombo. Mirex 2018: Generating and discriminating symbolic music continuations with bachprop. `https://www.music-ir.org/mirex/abstracts/2018/FC1.pdf`. Accessed on August 19, 2019.

[2] Jeff Ens and Philippe Pasuier. Quantifying musical style: Ranking symbolic music based on similarity to a style. In *ISMIR*, page forthcoming, 2019.

[3] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.