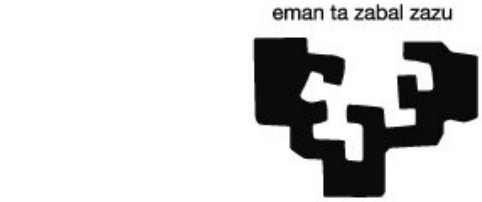


Universidad del País Vasco

Euskal Herriko Unibertsitatea

Konputazio Zientziak eta Adimen Artifiziala Saila

Methodological Contributions by Means of Machine Learning Methods for Automatic Music Generation and Classification

Thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy by:

Izaro Goienetxea Urkizu

Supervisors Basilio Sierra and Iñigo Mendialdua

July 2019

Izaro Goienetxea Urkizu

Methodological Contributions by Means of Machine Learning Methods for Automatic Music Generation and Classification

July 2019

Supervisors: Basilio Sierra and Iñigo Mendialdua

Euskal Herriko Unibertsitatea (UPV/EHU)

Konputazio Zientziak eta Adimen Artifiziala Saila

Donostia

This is supposed to be a happy occasion.
Let's not bicker and argue about who killed who.

Monty Python and the Holy Grail

Abstract

In the recent years, with the advances on computation sciences, many works on different topics related to music have been developed in order to make easier tasks that had to be made by hand. Different approaches have been developed, working both with audio and symbolic data, and several techniques have also been used, such as machine learning, rule based or deep learning methods. The applications to these methods are also diverse, and they go from automatic music generation to music recommendation systems.

In this research work two main topics have been studied: automatic music generation and classification. For the automatic music generation part a corpus of *bertso* melodies has been used to create a method that is able to generate new understandable tunes. The works made in this field have been based on the idea that having structures of repeated segments within a piece, at least in some abstract level, is necessary to understand it when we listen to it. Three versions of the generation method are presented in this manuscript, using different repetition structure definitions.

In the music classification part three main tasks have been tackled: genre classification, tune family identification and composer recognition. Different music representations have been used in the different tasks. Several machine learning techniques have also been applied in order to test which ones offer better classification results.

Finally, work on supervised classification has also been made, specifically on class binarization, where an attempt to optimize a previous binarization technique has been made. This optimization has been applied to several databases, among them one that consists of features of musical pieces of well known classical composers.

Laburpena

Azken urteetan konputazio zientzien gorakadarekin batera musikaren inguruan garatu diren lanak asko eta gai anitzetakoak dira. Bai audio formatuan oinarrituta eta baita formatu sinbolikoan egindako ekarpenak gero eta ugariagoak dira. Ekarpen hauek metodo ezberdinak erabiltzen dituzte, besteak beste, ikasketa automatikoko metodoak, erregeletan oinarritutakoak edo gaur egun hainbeste erabiltzen diren ikasketa sakoneko metodoak. Hauen aplikazioen artean musikaren sorkuntza automatikoa, sailkapena edota gomendio sistemen garapena aurki ditzakegu.

Ikerketa lan honetan bi gai nagusi landu dira: musikaren sorkuntza automatikoa eta sailkapena. Musikaren sorkuntzarako bertso doinuen corpus bat hartu da abiapuntu moduan doinu ulergarri berriak sortzeko gai den metodo bat sortzeko. Doinuei ulergarritasuna hauen barnean dauden errepikapen egiturek ematen dietela suposatu da, eta metodoaren hiru bertsio nagusi aurkeztu dira, bakoitzean errepikapen horien definizio ezberdin bat erabiliz.

Musikaren sailkapen automatikoan hiru ataza garatu dira: generoen sailkapena, familia melodikoen taldekatzea eta konposatzaileen identifikazioa. Musikaren errepresentazio ezberdinak erabili dira ataza bakoitzerako, eta ikasketa automatikoko hainbat teknika ere probatu dira, emaitzarik hoberenak zeinek ematen dituen aztertzeko.

Gainbegiratutako sailkapenaren alorrean ere binakako sailkapenaren gainean lana egin da, aurretik existitzen zen metodo bat optimizatuz. Hainbat datu baseren gainean probatu da garatutako teknika, baita konposatzaile klasikoen piezen ezaugarriez osatutako datu base batean ere.

Eskertzak

Azken 5 urteetako lanaren alde tekniko guztiak dokumentu honetan idatzi ondoren ez dakit nola hasi honaino iristen lagundu nauen jende guztiari eskerrak ematen. Askotan pentsatu dut momentu hau ez zela inoiz iritsiko, ez baita batere prozesu erreza izan, baina azkenean hemen gaude. Momentu askotan infernu baten antzekoa izan da, eta dena utzi eta supermerkatu batean lana bilatu nahi izan dut, baina asko ikasi dut, bai alde zientifikoan eta baita beste alderdi askotan ere, eta hori jende askori eskertu behar diot. Sekzio hau luzea izango da.

Lehenik eta behin nire zuzendariei eskerrak, Iñigo eta Basi, hasita zegoen eta zuen gaikoa ez zen tesi proiektu bat hartzera arriskatu eta salbatzeagatik. Iñigo, zuri ere bai. Zuek iritsi arte besteen tesietan zuzendariei eskerrak ematen zitzaizkienean pentsatzen nuen ez nuela jakingo nireari buruz zer jarri, eta orain askoz errazagoa da dena. Mila esker bihotzez, zuei esker bukatu ahal izan dut eta.

RSAIT taldeko kide guztiei ere eskerrak eman behar dizkiet, nahiz eta doktore ez izan lehenengo egunetik taldeko parte sentiarazi nautelako, inoiz azpimarratu gabe oraindik “bekaria” bat baino ez naizela. Ez dut uste zuek baino talde hobeagorik badagoenik.

Tesi honetan azaltzen diren ataza asko aurrera eramatea ezinezkoa izango litzateke Xenpelar Dokumentazio Zentroko jendeari esker izan ez balitz. Mila esker batez ere Nere eta Karlosi, doinutegiarekin laguntzeko beti prest agertu zaretelako, eta bide batez mila esker Joanito Dorronsorori, doinutegia sortzeagatik.

Fakultateko jende askori ere eman behar dizkiot eskerrak. Onekin taldeko guztiei bazkarietan zuen taldean onartzeagatik eta “vida real” bezelako jokoetan jolasteagatik. Victor, gracias por acompañarme al piano en uno de los marrones más grandes de la tesis. Te estaré eternamente agradecida. Elsa eta Mikeli esker bereziak, urte hauetan guztietan nere txapak pazientziaz entzuteagatik eta egunik zailenetan nerekin kejatzeagatik, nahiz eta beste egun batzuetan oso ondo ere pasa dugun!

Tesia unibertsitatean egin badut ere nire ikerkuntzarekiko interesa Vicomtech-en lanean ari nintzela piztu zen, eta han ezagutu nuen jende askori ere eskerrak eman behar dizkiot, tesi osoan zehar nire ondoan egon direlako. Eider, Ainhoa eta

Esther, pintaje de uñas egiten elkar ezagutu eta hemen jarraitzen dugulako, oraindik metodoa perfekzionatzen, eta Aitor (Elso), Carlos eta Javi, sois lo más. Greg, merci beaucoup d'être toujours là et de m'ecouter quand je me plains, (et je me plains beaucoup!).

Badaude pertsona batzuk nahiz eta tesiaren bukaera ikustera ez diren iritsi, aipatu nahi nituzkeenak. John Congote, nigan beti nik baino fede gehiago izateagatik eta fakultatera etortzera bultzatzeagatik. Osaba, beti gehiago ikastera animatzeagatik, nahiz eta ez zenuen oso argi benetan zer egiten nuen hemen, badakit gustatuko litzaizukela lana bukatuta ikustea. Azkenik Mariano Ferrer, matematikak letrak bezain ongi menperatzen ez bazituen ere beti lan hau ulertzen saiatu zelako, animoak eman eta bukaeran "tú lo que tienes que traer es un sobresaliente" esanez.

Eskerrik asko lagun guztiei, momentu askotan tesi honekin aurrera jarraitzera behartzeagatik.

Eta azkenik, baina ez horregatik gutxiago, mila mila esker familiari, anai-arreba eta gurasoei, beti edozer gauza egin dezakedala sinesteagatik (batzuetan gehiegi). Bereziki eskerrak Igor eta Garbiñeri, momentu guztietan, onetan eta txarretan, nere ondoan egoteagatik zerbeza batekin, eta amari. Dakizkidan gauzarik inportanteenak zugandik ikasi ditut eta beti erakutsi didazu gauzak kostata bada ere lortu egiten direla.

Eskerrik asko guztioi bihotz bihotzez.

Contents

1 Introduction **1**

1.1 Motivation . . . 2
1.2 Music representation . . . 2
1.2.1 MIDI . . . 2
1.2.2 Viewpoint representation . . . 4
1.3 Contributions . . . 5
1.3.1 Automatic Music Generation . . . 5
1.3.2 Supervised Classification . . . 6
1.3.3 Automatic Music Classification . . . 6
1.4 Thesis structure . . . 7

2 Automatic music generation **9**

2.1 Introduction . . . 9
2.2 State of the art . . . 11
2.3 Contributions . . . 13
2.3.1 Corpus . . . 14
2.3.2 Note level coherence . . . 15
2.3.3 Music generation with a coherence structure with multiple abstraction levels . . . 16
2.3.4 Rhythmic structure in template . . . 27
2.4 Conclusions and future work . . . 32

3 Supervised Classification **35**

3.1 Introduction . . . 35
3.2 Dynamic Classifier Selection . . . 37
3.3 Class binarization . . . 38
3.3.1 OVO . . . 39
3.3.2 Dynamic Classifier Selection and OVO . . . 40
3.4 Contributions . . . 41
3.4.1 PSEUDOVO . . . 41
3.5 Conclusions . . . 46

4 Music Classification **49**

4.1 Introduction . . . 49

4.2 State of the art 51
4.3 Contributions 53
4.3.1 Unsupervised classification of bertso melodies 53
4.3.2 Tune family classification with pattern covering 58
4.3.3 Composer recognition with matrix representation 63
4.4 Conclusions 66

5 Conclusions and future work **69**

I Articles Related to Automatic Music Generation **71**

6 Transformation of a bertso melody with coherence **73**

7 Melody Transformation with Semiotic Patterns **79**

8 Statistics based music generation approach considering both rhythm and melody coherence **93**

II Article Related to Supervised Classification **113**

9 Problems Selection Under Dynamic selection of the best base classifier in One versus One: PSEUDOVO **115**

III Articles Related to Music Classification **143**

10 Towards the use of similarity distances to music genre classification: A comparative study **145**

11 Melody classification with pattern covering **165**

12 On the Use of Matrix Based Representation to Deal with Automatic Composer Recognition **173**

Bibliography **181**

List of Figures

1.1 Example MIDI data. 3
1.2 Small fragment of a Dutch tune. Each line shows a different viewpoint and its transformation of the event sequence. Top: basic viewpoints, middle: derived viewpoints, bottom: linked viewpoints. 5

2.2 A segment of a score from the corpus with the semiotic labels without any reduction algorithm (row Label) and with a contour reduction algorithm (row Reduction). 15
2.4 A fragment from the melody *Abiatu da bere bidean*[1] and its viewpoint representation. 18
2.5 Schema of a possible semiotic structure for the template piece *Erletxoak lorean*. 19
2.8 Score of template piece *Erletxoak lorean* and two melodies generated with it. 21
2.9 Suffix-based database for an eight event sequence. 23
2.11 Score of the piece *Erletxoak lorean* (top) and two melodies generated using it as template and iterative random. 27
2.12 Diagram of the music generation method presented. 28
2.13 Score of the melody *Abiatu da bere bidean* where the main rhythmic pattern is highlighted. 28
2.14 Score of the melody *Neure lagunak lagun zakidaz* where its most interesting pattern is highlighted. 29
2.15 Nested pattern (in purple) found within the principal pattern (in black) in the melody *Neure lagunak lagun zakidaz*. 30
2.16 Examples of pieces generated from template *Abiatu da bere bidean*. . . 31

3.1 Code matrices for OVO (top left), OVA (top right) and ECOC (bottom) 38
3.2 Example of how the competent sub-problems are selected for two new samples 42
3.3 Example of how the base classifiers are selected for each sub-problem . 43

4.1 Diagram of the method presented in this work. 54
4.2 Viewpoint representation of the first two bars of the melody *Abiatu da bere bidean*. 55
4.3 Example of an interval matrix extracted from a piece in the corpus. . . 55

4.4 Small fragment from a Dutch tune. Each line shows a different viewpoint and its transformation of the event sequence. Top: basic viewpoints, middle: derived viewpoints, bottom: linked viewpoints. 58
4.5 Example (in red) of a pattern found in the intref representation of two pieces of the corpus. 59
4.6 Diagram of covering examples of a query and two different target pieces. Different patterns are represented in each covering with shapes, where patterns found in different viewpoint representations of the pieces have different shape. 60
4.7 Diagram of a covering example of a query and a target pieces conserving collinearity of patterns. 62

List of Tables

1.1 Significant MIDI messages shown on Figure 1.1 4

2.1 Results obtained in the evaluation. The template piece was the third melody sung. 22
2.2 Comparison between coverings of the pieces in the corpus using only pitch and interval information and coverings that include contour patterns. 25
2.3 Comparison between covering using only pitch and interval information and covering that includes contour information, on the 247 pieces that use contour patterns on the covering. 26

3.1 Example of training data 35
3.2 Summary of the databases used in this work. 44
3.3 Classification accuracies obtained with different methods for each database. The best results are highlighted in bold. 46

4.1 Best results obtained for each template and cluster number with contour matrices. 57
4.2 Best results obtained for each template and cluster number with interval matrices. 57
4.3 Classification accuracy with different viewpoints. (*)Classification on 347 pieces done where the viewpoint c3i(level) is used and not merged with any other viewpoint, since it is undefined for 13 pieces of the corpus. 61
4.4 Classification accuracy with different viewpoints forcing collinearity of patterns on the covering. 63
4.5 Number of pieces of each composer used in this work. 64
4.6 Global feature collection used in the global_{12} representation. 64
4.7 Accuracy results of the classifications with each single classifier and OVO technique for both corpora. The best obtained results are shown in bold. (*) SMO appears only once in the table since the function included in Weka has OVO already applied. 65
4.8 Confusion matrix of the classification of the matrix representation of corpus_5 using Multilayer Perceptron. 66

1 Introduction

Music is defined as "vocal or instrumental sounds (or both) combined in such a way as to produce beauty of form, harmony, and expression of emotion" (Oxford Dictionary). It is an organized combination of sounds that has the ability of telling stories and producing feelings on people. All the cultures in the world have some form of music, and it has existed since prehistory, but how is music created? The creation of such sound combinations is a creative process that cannot be ruled: there are no specific rules for creating good music. In fact, in many occasions breaking composition rules has led to more interesting music. Considering this, how can we generate music automatically with no human intuition that leads the process? How could a machine emulate the process that big composers like Satie, Paganini or Piazzolla followed when composing?

On the attempt to understand how music works, several musicologists have tried to analyse the structures of different music genres or the styles of different composers. Some of these analysis are complex studies that intend to discover some features that the oeuvres of a certain style or composer have in common. These kind of studies can be used to place unknown pieces within a certain historical period or to guess their authors, but a vast musicological and historical knowledge might be necessary to do them.

On the other hand, with the development of computation sciences the field of Music Information Retrieval (MIR) started gaining popularity. It is an interdisciplinary science aimed to study the processes, systems and knowledge representations required for retrieving meaningful information from music. It has many applications, such as music retrieval, music recommendation systems or playlist generation, among many others that are helpful to understand some aspects of music without the need of extensive musicological knowledge.

In this work we use music information retrieval tools to try to answer questions like, how could we automatically generate understandable melodies? Can we use some of the tools created in the music generation process to also classify music? In this PhD dissertation several machine learning techniques and music representations are proposed to achieve some automatic music generation and classification tasks.

1.1 Motivation

This research was born in the context of the *Learning to create* Lrn2Cre8 EU project[1], which aimed to understand the relationship between learning and creativity, by means of practical engineering, theoretical study, and cognitive comparison. It began from the hypothesis that creativity is a function of memory, that generates new structures based on memorised ones, by processes which are essentially statistical. One of the goals of the project was to generate music automatically using machine learning techniques. But how can we apply those machine learning techniques to automatically generate music? How do we represent music in order to learn the necessary statistics to create new musical content?

In an effort to answer those questions new ones arose; can we use the techniques that we use to generate music to classify existing music? Which machine learning techniques can we use to improve the performance of these classification tasks? Some techniques used in the music generation task led us to automatic music classification, for which different approaches have been developed. In the same way, this task led us to try to improve the classification accuracies, by applying and optimizing class binarization methods.

1.2 Music representation

An important aspect to consider when working with music is the representation that is used, which many times depends on the available data. In music information retrieval two data types are mainly used: audio data and symbolic data. In all the music related tasks tackled in this work, only symbolic data has been used. Symbolic music representations comprise any kind of score representation with an explicit encoding of notes or other musical events. These include machine-readable data formats such as MIDI, which is the format used in the different tasks presented in this work.

1.2.1 MIDI

Musical Instrument Digital Interface (MIDI) is an a standard adopted by the electronic music industry for controlling devices, such as synthesizers, electronic pianos and sound cards, that emit music. It was originally intended to control one keyboard from another, but it was quickly adopted for the personal computer.

[1] https://web.archive.org/web/20160722213851/http://lrn2cre8.eu:80/

MIDI itself does not make sound, it is just a series of messages, like event messages that specify the instructions for music, including the pitch (MIDI note number), velocity (intensity) and clock signals (which set the tempo). When a musician plays a MIDI instrument, for each key he or she presses, a *note-on* event is registered, which includes which key it was and how hard it has been pressed, and the time of the event. When the performer releases the key, a *note-off* event is registered with the same information. In Figure 1.1 an example of MIDI stream can be seen. Each event has a track and channel parameter because each MIDI can include several tracks, where each track can consist of a maximum of 16 channels. In the example in the figure, all the events are in the track 0 and channel 1, because it is a one track file with a single channel.

```
<MidiEvent TIME_SIGNATURE, t=None, track=0, channel=None, data=b'\x03\x02\x18\x08'>
<MidiEvent DeltaTime, t=0, track=0, channel=None>
<MidiEvent SEQUENCE_TRACK_NAME, t=None, track=0, channel=None, data=b'Violin'>
<MidiEvent DeltaTime, t=0, track=0, channel=None>
<MidiEvent KEY_SIGNATURE, t=None, track=0, channel=None, data=b'\x01\x00'>
<MidiEvent DeltaTime, t=0, track=0, channel=None>
<MidiEvent INSTRUMENT_NAME, t=None, track=0, channel=None, data=b'Violin'>
<MidiEvent DeltaTime, t=0, track=0, channel=None>
<MidiEvent NOTE_ON, t=None, track=0, channel=1, pitch=64, velocity=85>
<MidiEvent DeltaTime, t=48, track=0, channel=None>
<MidiEvent NOTE_OFF, t=None, track=0, channel=1, pitch=64, velocity=85>
<MidiEvent DeltaTime, t=0, track=0, channel=None>
<MidiEvent NOTE_ON, t=None, track=0, channel=1, pitch=62, velocity=85>
<MidiEvent DeltaTime, t=48, track=0, channel=None>
<MidiEvent NOTE_OFF, t=None, track=0, channel=1, pitch=62, velocity=85>
<MidiEvent DeltaTime, t=0, track=0, channel=None>
<MidiEvent NOTE_ON, t=None, track=0, channel=1, pitch=64, velocity=85>
<MidiEvent DeltaTime, t=48, track=0, channel=None>
<MidiEvent NOTE_OFF, t=None, track=0, channel=1, pitch=64, velocity=85>
<MidiEvent DeltaTime, t=0, track=0, channel=None>
<MidiEvent NOTE_ON, t=None, track=0, channel=1, pitch=67, velocity=85>
<MidiEvent DeltaTime, t=48, track=0, channel=None>
<MidiEvent NOTE_OFF, t=None, track=0, channel=1, pitch=67, velocity=85>
<MidiEvent DeltaTime, t=0, track=0, channel=None>
```

Figure 1.1: Example MIDI data.

In Table 1.1 an explanation of the different messages of Figure 1.1 is shown. It can be seen that even though the pitch number of each note in the MIDI file is well specified in each NOTE_ON and NOTE_OFF event, what indicates their duration is the DeltaTime event (in ticks) that is introduced between the NOTE_ON and NOTE_OFF events. That is inconvenient, since the exact duration of the notes is not explicitly coded. However, some software libraries that work with MIDI use techniques to round these durations.

Several libraries exist to work with MIDI data. In this work different languages have been used, such as MIDI-Perl, Java sound and music21 (python).

Event	Function	Relevant Parameters
TIME_SIGNATURE	Indicates a change in the time signature	data: New time signature
SEQUENCE_TRACK_NAME	Gives a name to the track	data: Track name
KEY_SIGNATURE	Gives the key signature indicating the number of sharps and flats	data: Number of sharps Number of flats
INSTRUMENT_NAME	Name of the instrument used in the track	track: Track number, data: Name of the instrument
NOTE_ON / NOTE_OFF	Turns a note on or off	track:Number of track in which the note is added, channel: Number of channel within the track, pitch: Pitch of the added note velocity: Velocity or intensity of the note

Table 1.1: Significant MIDI messages shown on Figure 1.1

1.2.2 Viewpoint representation

In all the music related tasks presented in this dissertation a multiple viewpoint representation of the pieces [CW95] has been applied in order to represent different features of the events in the pieces. A *viewpoint* τ is a function that maps an event sequence $e_1, \ldots, e_\ell$ to a more abstract sequence $\tau(e_1), \ldots, \tau(e_\ell)$, comprising elements in the codomain of the function τ.

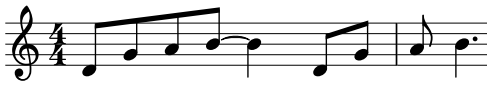
Three main viewpoint types exist: basic viewpoints, derived and viewpoints built using constructors.

- Basic viewpoints: Basic attribute domains such as pitch number or duration of the event.

- Derived viewpoints: Viewpoints that do not feature in the event space but are derived from one or more basic types, such as interval or melodic contour. They may be undefined ($\perp$) for some events.

- Viewpoints built using constructors: They are used to describe the interaction between different viewpoints. In this work only *linked* ones are used. These linked viewpoints represent every event as a pair the values of its constituent viewpoints.

In Figure 1.2 a short excerpt of a melody is shown with its viewpoint representation, where basic, derived and linked viewpoints can be seen. The concrete viewpoints shown in the figure are described later in Chapter 4.



pitch	62	67	69	71	62	67	69	71
dur	840	840	840	2520	840	840	840	2520
int	$\perp$	5	2	2	-9	5	2	2
ioi	$\perp$	840	840	840	2520	840	840	840
intref	P5	P1	M2	M3	P5	P1	M2	M3
c3(pitch)	$\perp$	$+$	$+$	$+$	$-$	$+$	$+$	$+$
c5(pitch, 3)	$\perp$	$++$	$+$	$+$	$--$	$++$	$+$	$+$
c3(dur)	$\perp$	$=$	$=$	$+$	$-$	$=$	$=$	$+$
c3i(level)	$\perp$	$+$	$-$	$+$	$-$	$+$	$-$	$+$
int $\otimes$ intref	$\perp$	5, P1	2, M2	2, M3	-9, P5	5, P1	2, M2	2, M3
intref $\otimes$ c3(pitch)	$\perp$	P1, $+$	M2, $+$	M3, $+$	P5, $-$	P1, $+$	M2, $+$	M3, $+$

Figure 1.2: Small fragment of a Dutch tune. Each line shows a different viewpoint and its transformation of the event sequence. Top: basic viewpoints, middle: derived viewpoints, bottom: linked viewpoints.

1.3 Contributions

This PhD project is based on the contributions compiled in Chapter 6 and some contributions that have not been published but are detailed in this manuscript. Such contributions include scientific contributions to the fields of Automatic Music Generation, Supervised Classification and Automatic Music Classification. The list of the main publications is presented below.

1.3.1 Automatic Music Generation

- [GC15] Izaro Goienetxea and Darrell Conklin. "Transformation of a bertso melody with coherence". In: *Proceedings of the 5th International Workshop on Folk Music Analysis (FMA 2015)*. Paris, France, 2015, pp. 56–58

  In this paper a semiotic structure that describes a template piece on the note level is presented. The defined semiotic structure is used to generate new melodies with a bigram statistical model built from a corpus of 100 bertso melodies and a stochastic hill climbing optimization method. A melodic reduction algorithm is applied not to constrain the least important notes and to allow a wider variety of generations.

- [GC18] Izaro Goienetxea and Darrell Conklin. “Melody Transformation with Semiotic Patterns”. In: *Music Technology with Swing*. Ed. by Mitsuko Aramaki et al. Cham: Springer International Publishing, 2018, pp. 477–488

  In this work a pattern discovery and ranking method is used to create a coherence structure of a template piece which describes the melodic relations between its segments. The coherence structure is used along with a trigram model of intervals built from a corpus of 1934 bertso melodies to generate new melodic lines. These melodic lines are sampled within the same rhythmic content of the template piece. An evaluation of some generated melodies has been carried out, with positive results.

- Statistics based music generation approach considering both rhythm and melody coherence. **Submitted** to IEEE Access journal

  This paper extends the work of [GC18], adding the analysis and generation of new rhythmic information in addition to the melodic line. An abstract template is created from a template piece, which includes its melodic and rhythmic coherence structures. Two statistical models are built from a corpus of 1934 bertso melodies, and a sampling strategy which generates rhythmic and melodic information is applied to generate new pieces. The generated melodies have been evaluated with good results.

1.3.2 Supervised Classification

- Problems Selection Under Dynamic selection of the best base classifier in One versus One: PSEUDOVO. **Submitted** to Applied Soft Computing journal

  This article proposes an optimization to the DYNOVO binarization technique, which tries to select the best classifier for each sub-problem of OVO dynamically for each new sample. In the presented optimization the sub-problems that are more relevant in the classification are selected, discarding the irrelevant ones. The accuracies of the method have been compared to other state-of-the-art methods with good results.

1.3.3 Automatic Music Classification

- [Goi+18b] Izaro Goienetxea et al. “Towards the use of similarity distances to music genre classification: A comparative study”. In: *PLOS ONE* 13.2 (2018), pp. 1–18

This work proposes a matrix based representation of the pieces of a corpus of bertso melodies, which is used along with several distances to create clusters of melodies that presumably belong to the same genre. New melodies are generated from each of the clusters using different pieces as templates. The generated pieces are supposed to belong to the same genres of the melodies of the clusters used to generate them. Those new melodies are then classified into the clusters using the same distances of the first step to validate both the representation and the cluster generation process.

- [Goi+16] Izaro Goienetxea et al. "Melody classification with pattern covering". In: *9th International Workshop on Music and Machine Learning (MML 2016)*. Riva del Garda, Italy, 2016

  This paper presents a tune family classification method that is based on the use of pattern discovery on the pieces of a corpus of Dutch folk tunes. The patterns discovered within different viewpoint representations of the pieces are merged to cover pairs of pieces following a leave-one-out strategy. The interests of these patterns are summed to measure the similarity between the pieces. The goal of the process is to find the piece that is most similar to each piece in the corpus. The presented method has obtained better results than most of the works that have faced the same classification task.

- [Goi+18a] Izaro Goienetxea et al. "On the Use of Matrix Based Representation to Deal with Automatic Composer Recognition". In: *AI 2018: Advances in Artificial Intelligence - 31st Australasian Joint Conference, Wellington, New Zealand, December 11-14, 2018, Proceedings*. 2018, pp. 531–536

  This work proposes a matrix based representation to classify polyphonic pieces of well known composers. A classification process has been carried out with different classifiers and the obtained classification accuracies have been compared to those obtained with a global feature representation used in a similar classification task that obtained good results. An OVO binarization has also been applied in the classification process to test whether it improves the accuracies. The results show that the matrix representation performs better than the global feature representation, and that the application of OVO is overall beneficial.

1.4 Thesis structure

This PhD dissertation is divided in six main chapters. Considering this introductory chapter, the remaining five are introduced below.

Chapter 2

In this chapter the work done on the field of automatic music generation is presented. The main idea of the chapter is to use an existing piece as template to generate new melodies that have the same coherence structure. Several ways to do that have been presented. The chapter is structured in four section. The first one provides an Introduction to the field, then a State of the Art describes the current state of the methods used in automatic music generation. The Contributions section enumerates the contributions to the field presented in this work, and finally, the Conclusions and future work are presented.

Chapter 3

This chapter describes the contribution made in this work to the field of supervised classification, and particularly to class binarization strategies. The chapter is divided into five sections, including (1) an Introduction; (2) Dynamic Classifier Selection, where these kind of methods are described; (3) Class binarization that explains the binarization types and specifies some optimizations done for One-vs-One; (4) Contributions to the topic and (5) Conclusions and future work.

Chapter 4

In this chapter the work done in the field of automatic music classification is presented. Three main classification tasks have been tackled: genre classification, tune family identification and composer recognition. The chapter is structured as follows. First an Introduction of the field is presented, then a State of the Art section describes the works that have been done in the different tasks that are faced in this work, followed by the Contributions section and Conclusions and future work.

Chapter 5

This chapter describes the main conclusions extracted from this PhD work as well as the paths that are envisaged for the future.

Chapter 6

The main publications that support this dissertation are presented in this chapter, listed in the same order as they have been listed in Section 1.3.

2 Automatic music generation

2.1 Introduction

The idea of generating music automatically is from even before the existence of computers. Some of the first attempts to create music without any knowledge on music composition are the *Musikalisches Würfelspiel* or music dice games [Hed78]. These games were popular throughout Western Europe in the 18th century and consisted in rolling a dice to get numbers that belonged to pre-composed segments, that were then joint together to create new musical pieces. An example of these games is the one published in 1792 that was attributed to Mozart, even though this attribution has not been authenticated.

The earliest automatically generated compositions are from the mid-1950s, around the same time as the concept *Artificial Intelligence* was coined. Among the first automatically generated compositions is the Illiac Suite, composed by a computer that had been programmed by Lejaren Hiller and Leonard Isaacson in 1957 [Edw11]. Around the same time, Martin L. Klein and Douglas Bolitho, engineers at Burroughs Corporation, programmed a computer nicknamed “Push-Button Bertha” to compose popular songs automatically [Hol08]. It composed 4000 pop tunes after being trained with 100 that were popular at that moment.

Another pioneer in the use of computers for algorithmic composition was Iannis Xenakis, who created a program that would produce data for his “stochastic” compositions. He wrote about them in detail in his book Formalized Music (1963) [Xen92].

After these first steps many composition methods have been developed, such as knowledge based systems, evolutionary and other population-based methods, fractals or statistical models [FV13]. Statistical models of symbolic music, for example, have been used to model several styles of music. The main advantage they offer is that they can reflect some features, melodic or rhythmic, of a corpus and can be learned from the corpus itself without the need of having any preprogrammed rules. These features could then be used to generate new musical sequences in the same style of the corpus [Dub+03; AW04].

When generating new pieces an important feature that needs to be taken into account is their coherence. Different theories have been developed on how the music should be structured in order to be comprehensible, but many of them agree on the need of segments that are repeated, or related in a more abstract level, through the pieces. Arnold Schoenberg [Zbi99] thought that listeners have to recognize musical figures and how they cohere in order to understand what they are listening. Other theories compare musical discourse and linguistics, as well as the mechanisms that the human brain has to understand them [Pat08; Ana97]. According to these theories, as well as in linguistics, relations between different segments of a musical piece are necessary to build a coherent discourse.

Different relations between segments can be found in music, repetition being the most obvious of them. It is a fact that almost all forms of music involve repetition [LF95], either of sequences of pitch of notes or at some higher structural grouping, and those repetitions impart a meaning to music [Mey57]. These repeated sequences are called *motifs*, and they are defined as "the smallest part of a piece or a section that, despite change and variation, is recognizable as present throughout" [Sch+06].

Motivic analysis of pieces was used by Cope in his work Experiments in Musical Intelligence (EMI) [Cop04], where he implemented a pattern matching process that found motifs repeated twice or more times within a piece, allowing the motifs to be transpositions or inexact repetitions. The information of these motifs was stored in order to be used later on the generation of new musical information. As Hofstadter described, EMI uses the relations between the discovered motifs within a piece as template to sample new notes in it, so that new pieces can be created which are identical on the template level, but sound truly novel on the note level. He coined this process with the term *templagiarism*, from "template plagiarism". The idea behind this was to create new pieces that would emulate the style of the author, who would tend to repeat these motifs through their oeuvre.

Even though Cope used templagiarism to create new pieces that emulated his composition style, following the idea of needing motifs in music to make it coherent or comprehensible, the same technique could be used to generate new pieces with coherence. To do so a structure that describes the coherence of a template piece should be built, which can be done in different ways, like the description of the acoustic structure, functional structure or semiotic structure. Semiotic structure is defined as the representation of similar segments by similar arbitrary symbols [Bim+12].

Following these ideas, in the works described in this chapter semiotic structures (also referred to as 'coherence structures' in this work) have been built to describe different relations between segments in a template piece, where a different semiotic

label is assigned to each different segment. Once a label is assigned to each segment in the template, that semiotic structure can be used to generate music that respects the relations between the segments, but with different melodic content.

In this chapter the main contributions made to the automatic music generation field are presented. Works with different semiotic structure definitions have been carried out, as well as a complete music generation method in which several contributions have been made. Bertso melodies have been used in the generation methods because of their suitability, since they are usually not complex and structured melodies.

2.2 State of the art

Many different approaches have been developed in the field of automatic music generation, and even though there is not a fixed taxonomy of automatic music composition methods, they are often classified as knowledge-based (or rule-based) methods, evolutionary methods, machine learning methods or hybrids.

Knowledge-based methods are based on pre-made sets of arguments or rules that are used to compose new music on a certain style or genre. Grammar models and rule learning methods are some examples of this type of generation. Grammar models produce musical pieces using rules, which expand high level symbols into detailed sequences of symbols (words). These rules can be hand coded by an expert or they can be learned from a corpus of melodies that share a genre or style. An example of the use of grammars for music generation is the one developed by Chemillier [Che04], which generates jazz chord sequences based on Steedman's grammar. This grammar was created from a set of modern jazz 12-bar chord sequences, which is considered a wide and representative range of permissible variations of the blues basic form.

Evolutionary methods are based on the improvement of a population by cycles of evaluation and reproduction with variation of its individuals. The process starts with the generation of the candidate solutions of the initial set, then in each cycle the candidates are changed by mutation or recombination and they are evaluated using a fitness function. Then, a selection method is applied which chooses the best solutions. These cycles are repeated until a stopping criteria is satisfied.

Evolutionary algorithms have been used in different tasks of music generation like in *GenJam* [Bil07], an interactive jazz improvisation system. GenJam uses a training process, in which the system plays a tune and a human mentor evaluates it as good or bad. These evaluations are then used to adjust the fitness function. Another example of the use of evolutionary algorithms for music composition is *MetaCompose*

[Sci+16], which is a component-based system for music generation that supports real-time improvisation. The composition process has three main steps: creation of a chord sequence, evolution of a melody fitting this chord sequence and creation of an accompaniment for the melody/chord sequence combination.

Machine learning methods extract the knowledge from a corpus instead of having it previously defined. Statistical models are an example of machine learning methods, in which different features of a corpus can be represented. These models are able to assign probabilities to automatically generated pieces, that indicate how similar the new piece is to those in the corpus.

Statistical models of music have been applied to the generation of melodies and harmonies in several works, and they go from the earliest Markov models [Bro+56] to new models based on deep learning [BL+12; Bri+17]. Whorley and Conklin [WC16] apply statistical models to generate four-part harmonizations using horizontal and vertical viewpoints of music and an iterative random walk sampling. Herremans et al. [Her+14] use a first order statistical model to capture the melodic and harmonic features of a first species counterpoint corpus and generate new musical content in the same style. To do so, they propose the use of a sampling method named Variable Neighborhood Search (VNS), which starts with a randomly generated fragment and optimize it making local changes to increase the probability of the fragment. They compare it to other sampling methods like random walk or Gibbs sampling.

Padilla and Conklin [PC18] have developed a method to compose Palestrina masses using a combination of statistical models, to capture the stylistic aspects of the music, and pattern discovery to extract the coherence structure of original pieces. The pattern discovery process is performed on a single viewpoint representation of the pieces, and the discovered patterns are used to build the coherence structure. That structure is then used to guide the generation of new musical material along with a first order Markov model. Collins et al. [Col+14] also consider the coherence problem for generating new music by defining a template from an existing polyphonic piece to sample new notes onto it. In the definition of the template, geometric patterns are discovered in a point representation of the notes in a pitch-time space for which a pattern discovery method named SIACT [Mer+02] is used. This pattern discovery strategy is able to discover exact repetitions and transposed segments.

Deep learning architectures are more and also more popular in music generation, and well known groups like Magenta[1] at Google are using them to generate new music. Deep learning is defined as a repertoire of machine learning techniques based on artificial neural networks which have multiple layers to process multiple abstraction layers of the data [BP17], and several approaches to automatically create music using

[1] https://magenta.tensorflow.org/

these techniques have been developed. For example Bretan et al. [Bre+16] apply a unit selection methodology to analyse if using only the units available in a library can be enough to generate a wide spectrum of new musical content. Then, they use a combination of a Deep Structured Semantic Model (DSSM) and Long Short Term Memory (LSTM) networks to predict the next unit in the generation model. Other generation works are based on the use of Generative Adversarial Networks (GANs) to create music, like MidiNet [Yan+17] and MuseGAN [Don+18].

Even though this is a growing area of research and interesting results are obtained using deep learning architectures for music generation, they still have some limitations, like the control (of tonality conformance, rhythm...), structure (giving direction to the generated music), creativity (versus imitation) and interactivity [BP17].

2.3 Contributions

The contributions of this work to the field of automatic music generation are related to the development of a method that is capable of generating new coherent melodies. The main idea is using a coherence structure extracted from an existing piece, which guides the generation process along with a statistical model based on a corpus of monophonic melodies. The corpus used in the generation has been analysed to look for any issues it might have, that could have negative consequences in the building of statistical models.

A first version of the generation method is presented, which is based on the semiotic description of the template piece on the note level; a different semiotic label is assigned to each different note of the piece used as template. These labels must then be respected when sampling new notes within them. Since this semiotic description can be too restrictive on the generation phase, a melodic reduction algorithm is applied to the labels, in order to get a wider variety of possible solutions.

The coherence structure definition has evolved to include relations between segments instead of describing relations between notes, and several abstraction levels are included into the coherence structure. This structure is used to generate new melodic lines with the same coherence and rhythmic information of the template piece. Finally, a method that also includes the rhythmic information into the coherence description and generation steps has been developed.

The papers that describe the mentioned methods are presented below, as well as some contributions related to the generation methods that are not described within the papers.

2.3.1 Corpus

The corpus used in the music generation methods presented herein is the *Bertso doinutegia*, a collection created by Joanito Dorronsoro and published for the first time on 1995 [Dor95]. It is maintained and updated every year by *Xenpelar Dokumentazio Zentroa*[2] with new melodies that are used in competitions and exhibitions. Entries in the collection have a melody name, the name or type of the strophe and type of the melody (genre), among other information. Since bertso melodies do not share a specific style, pieces of many genres can happen in the corpus. As a consequence no rule set can be defined to describe the corpus, and a statistical model should be used to capture some features, melodic or rhythmic, that the pieces have in common.

The scores included in the collection have been encoded in Finale and exported to MIDI. Currently the collection is composed of 2382 bertso melodies, which have a mean length of 60 notes. 85 of the melodies are polyphonic or have polyphonic parts. Since in this music generation method monophonic pieces are generated, all the pieces with polyphony are processed using a *skyline* method, which takes the event with highest pitch at unique onset times.

It has been detected that many pieces have long sequences of repeated notes, which could cause problems when used to build statistical models. Sequences of four repeated notes have been discovered occurring at least twice in 744 pieces (31% of the corpus), and at least once in 1064 (44.7% of the corpus). A model built from these sequences would assign high probability to long note repetitions, and even though these sequences exist in the corpus, it has been decided that the generation of such sequences should be avoided because of their lack of interest. Figure 2.1 shows the score of the melody with the highest number of sequences of repeated notes, which clearly is not musically interesting.

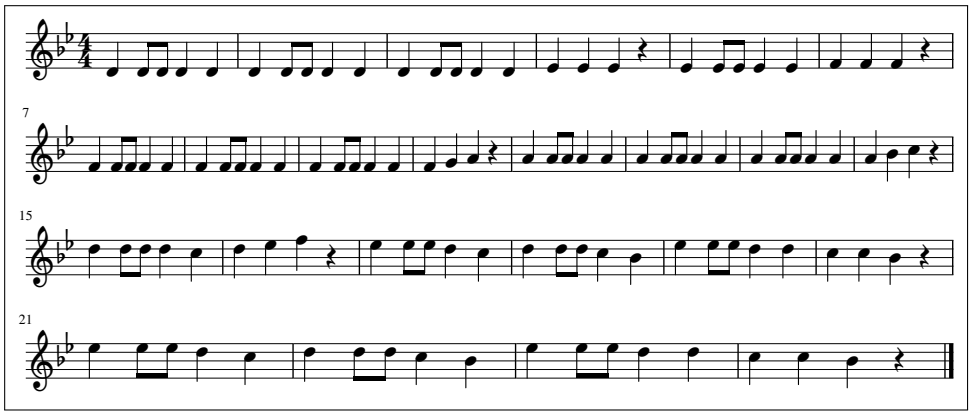


Figure 2.1: Score of the melody Neska zaharrak eta apaizak II [3]

[2] http://bdb.bertsozale.eus/es/
[3] http://bdb.bertsozale.eus/en/web/doinutegia/view/1648-neska-zaharrak-eta-apaizak-ii

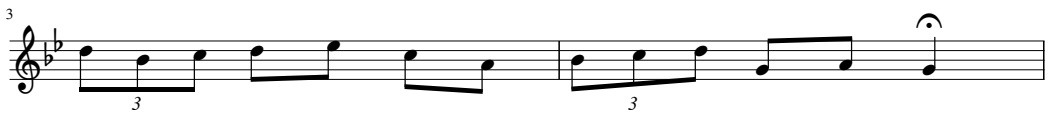
To avoid the negative effect that these pieces could have on the statistical model, the pieces with the highest proportion of repeated notes are removed from the corpus, reducing the size of the corpus to 1934 pieces (82.2% of the original corpus).

2.3.2 Note level coherence

The simplest way to do a semiotic description of a piece is describing it on the note level. A different semiotic label is assigned to each different note in the piece, and on the sampling step each different label is sampled with a different note. Even though it is a very simple solution, the use of these labels on the generation of new melodies, making sure that each different label is sampled with a different note, preserves the segment structure of the template in the new pieces.

Assigning semiotic labels to all the notes in a piece can be too restrictive on sampling, and in some cases it does not allow many different generations, specially when the allowed note vocabulary is not big. To solve this issue a melodic reduction method is applied which attempts to identify the most significant notes in the piece. Several methods exist to reduce melodies, like the time span reduction method [Jac87], or a grammar based reduction method developed by Groves [Gro16]. In this work a contour reduction method that identifies the notes where the melodic direction changes is proposed, similar to the contour reduction viewpoint [CA06]. Applying this method we assume that the notes in a piece where the melodic direction changes are more important than the passing notes. Once the most important notes are identified, a semiotic label X is assigned to the less significant notes in the segments, where X matches any note, and these positions are unconstrained on sampling.

An example of note level labelling can be seen in Figure 2.2, where the labelling of a segment of the melody *Abiatu da bere bidean* is shown. The note-level semiotic labelling is shown in the row *label*, where all the different notes have different labels. The row *reduction* shows the labelling after the contour reduction, where the notes where the contour direction changes are identified, and the passing notes are labelled with a label X, leaving them unconstrained.



pitch	74	70	72	74	75	72	69	70	72	74	67	69	67
label	A	B	C	A	D	C	E	B	C	A	F	E	F
reduction	A	B	X	X	D	X	E	X	X	A	F	E	F

Figure 2.2: A segment of a score from the corpus with the semiotic labels without any reduction algorithm (row Label) and with a contour reduction algorithm (row Reduction).

After labelling the notes of the piece a bigram model is built from a small selection of pieces of the corpus to sample new notes within the template.

The main publication related to the described work is the one below:

- Izaro Goienetxea and Darrell Conklin. "Transformation of a bertso melody with coherence". In: *Proceedings of the 5th International Workshop on Folk Music Analysis (FMA 2015)*. Paris, France, 2015, pp. 56–58 [GC15]

2.3.3 Music generation with a coherence structure with multiple abstraction levels

In this approach related segments are identified within a piece to use their relations as an abstract template to create new melodies. A coherence structure is built to describe melodic relations of various abstraction levels between segments of a piece. A statistical model is then built from the corpus of bertso melodies presented in Section 2.3.1 to sample new high probability sequences that respect the coherence structure. These new sequences conserve the rhythmic information of the template, because in this approach only melody is generated.

The most common relations between segments are repetition and transposition. For example, Collins [Col+14] uses these relations when describing the structure of a piece. In our approach new relations are added in order to consider sequences with more abstract relations, like contours, in the structure description. In Figure 2.3 a diagram that describes the proposed generation method is shown. This method has three main steps:

1. Building of the coherence structure
    a) Viewpoint representation of the template piece
    b) Pattern discovery in viewpoint representations
    c) Covering of the template piece
2. Building of the statistical model
3. Generation of the new pieces

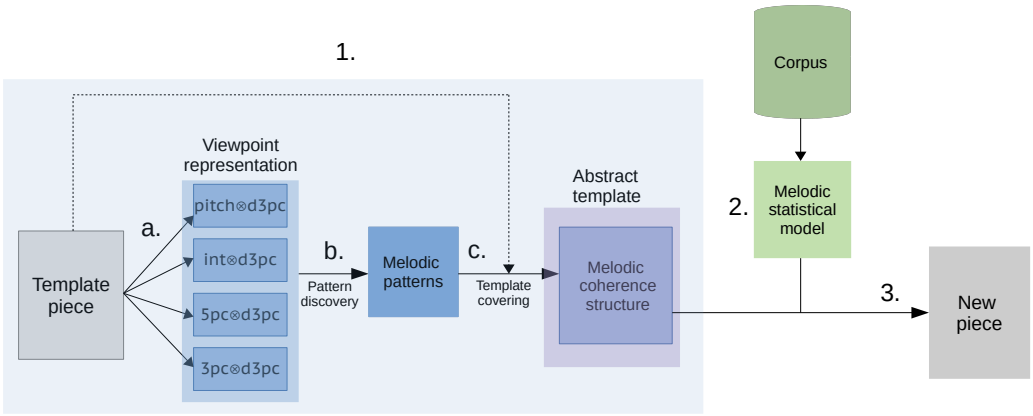


Figure 2.3: Diagram of the presented melodic generation method.

Each of these steps is better described below.

1. **Building of the coherence structure**

A coherence structure is built from the template piece, which is then used in the generation step. In the process of building this coherence structure three steps have been defined:

a) Viewpoint representation of the template piece
In addition to the repetition and transposition relations used in other works, more abstract representations are also added; contour representations. To represent these abstraction levels in a piece, a multiple viewpoint representation [CW95] is used. In Figure 2.4 a two bar fragment of a melody of the corpus can be seen, as well as its melodic viewpoint representation. The used viewpoints are pitch, interval and two contour viewpoints; 5pc and 3pc. Five point contour viewpoint 5pc computes whether the contour between two contiguous notes goes more than a scale step (one or two semitones) down (ld), goes one scale step down (sd), goes more than a scale step up (lu), goes one scale step up (su), or stays equal (eq), and three point contour viewpoint 3pc represents if the contour between two contiguous notes goes up (u), down (d) or stays equal (eq). A rhythmic viewpoint d3pc is also shown, which represents if the duration of two contiguous notes goes up (u), down (d) or stays equal (eq). A *linked* viewpoint 5pc $\otimes$ d3pc is also shown. These viewpoints represent the interaction between two viewpoints as pairs of values from its constituent viewpoints. In this case the interaction between all the melodic viewpoints and the rhythmic one is used to represent the template pieces.

b) Pattern discovery in viewpoint representations
In each viewpoint representation of the template piece a pattern discovery algorithm is run, which discovers all the patterns that happen twice or more times within each

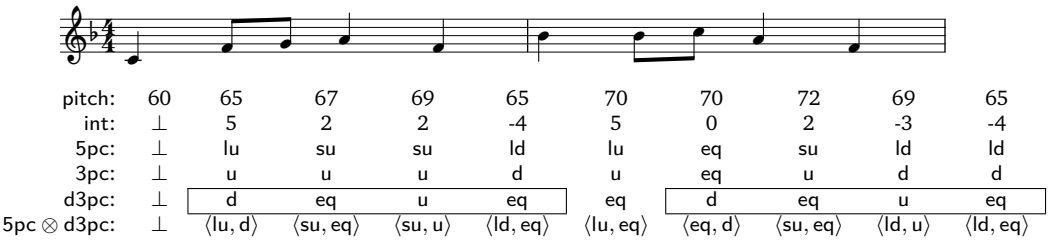


Figure 2.4: A fragment from the melody *Abiatu da bere bidean*[4] and its viewpoint representation.

representation. As an example, in the d3pc representation of Figure 2.4 a pattern of four elements has been highlighted.

It should be mentioned that not all the discovered patterns can be part of the coherence structure of a piece; only the most important ones should be included. Since the used pattern discovery method only returns the number of occurrences of each discovered pattern, and this is not enough information to decide if it is important or not, a new interest measure is computed for all the discovered patterns. This interest measure computes the probability of a pattern to happen in a piece of the corpus m or more times, then a pattern is interesting if it occurs more frequently than expected. This is a standard model for assessing discovered motifs in music informatics [CW16] and bioinformatics [Hel+98].

To compute the interest $\mathbb{I}$ of a pattern, first, we note that the background probability p of finding a pattern $P = \tau{:}(v_1, \ldots, v_m)$ in a segment of exactly m events can be computed using a zero-order model of the corpus:

$$p = \prod_{i=1}^{m} \frac{c(v_i)}{c},$$

where $c(v_i)$ is the total count of the feature $\tau : v_i$ and c is the total number of places in the corpus where the viewpoint τ is defined. Then the binomial distribution $\mathbb{B}(k; n, p)$ gives the probability of finding the pattern exactly k times in n events,

$$\mathbb{B}(k; n, p) = \binom{n}{k} p^k (1-p)^{n-k}$$

and therefore the negative log probability of finding k or more occurrences of the pattern in a template piece with ℓ events is

$$\mathbb{I}(P) = -\ln \mathbb{B}_{\geq}(k; n, p), \tag{2.1}$$

where $\mathbb{B}_{\geq}$ is the upper tail of the binomial distribution, with $n = \ell - m + 1$ being the maximum number of positions where the pattern could possibly occur in the template piece.

c) Covering of the template piece

The list of patterns discovered within the different representations of the template piece is sorted according to their interest, and they are used to cover the template piece, trying to use the patterns with the highest interest. In the covering phase no overlapping of contiguous patterns is allowed to make the sampling part easier. In Figure 2.5 a covering example for the melody of the corpus *Erletxoak lorean* is shown, where the used patterns, their type and interest value can be seen. In this covering two patterns have been used; 3pc $\otimes$ d3pc:(A) and pitch $\otimes$ d3pc:(B), which are the patterns that form the coherence structure that is then used on the sampling process.

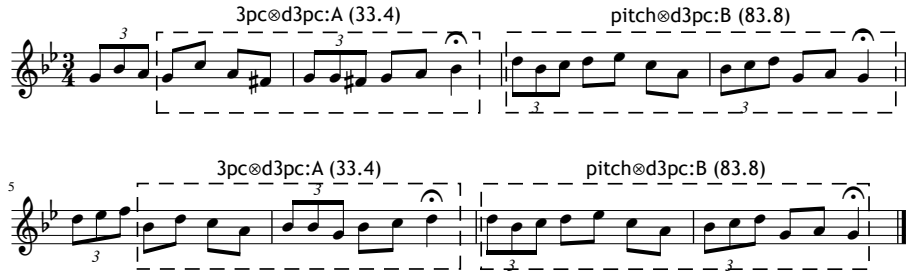


Figure 2.5: Schema of a possible semiotic structure for the template piece *Erletxoak lorean*.

After computing the interest value of all the discovered patterns an interest threshold should be set, to avoid considering patterns that are not important enough for the coherence structure. This threshold is set at 9.5, as a result of an analysis of the discovered patterns performed by hand, comparing different patterns, their significance in the score they are discovered in, and their interest value. As an example, in Figure 2.6 a score of the melody *Argi emaile txit diztiratsu* is shown, which is part of the corpus and where a three element pattern is highlighted. This is pattern pitch:$(64, 65, 67)$ which has an interest value of 9.2, and is not considered distinctive enough. The interest threshold is set at 9.5 to avoid this kind of patterns in the coverings.

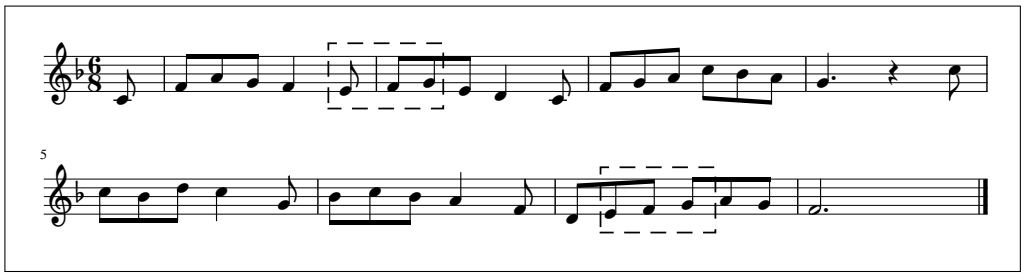


Figure 2.6: Score of the melody *Argi emaile txit diztiratsu*[5]with a not distinctive pattern highlighted.

[5]`http://bdb.bertsozale.eus/en/web/doinutegia/view/1087-argi-emaile-txit-diztiratsu`

In Figure 2.7 the score of the melody *Lagundurikan danoi I* is shown, where the pattern 3pc:$(u, u, d, d, eq, d, eq, d)$ is highlighted. The interest value of this pattern is 9.55, which is near the established interest threshold, but is considered a good enough pattern, considering the amount of information it represents.

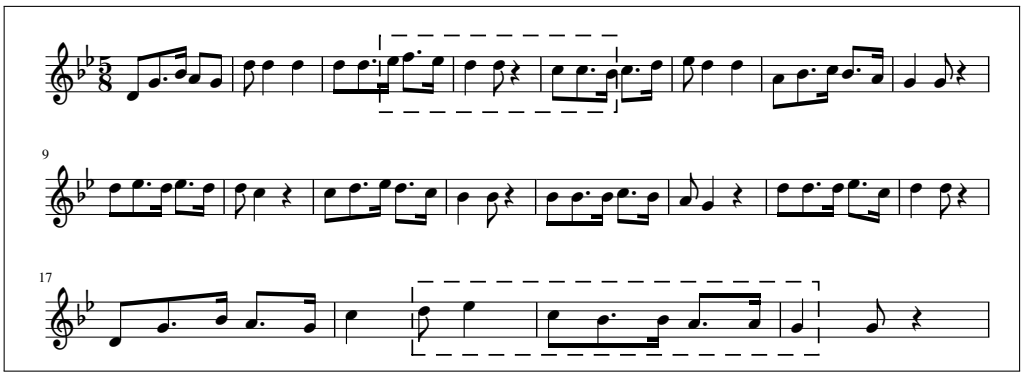


Figure 2.7: Score of the melody *Lagundurikan danoi I*[6]. A pattern considered interesting is highlighted.

2. **Building of the statistical model**

Once the covering of the template is done, the statistical model is built. It has been decided that using *n-grams* was the best option to describe the corpus, but the length of the model and the feature it describes have to be selected. After a study of how well each melodic viewpoint describes the corpus for different model lengths a trigram model of interval has been chosen to characterize the corpus.

The statistical model is used to assign probabilities to sequences of notes, where a higher probability means that the sequence is more similar to those in the corpus. The exact probability of a piece using a trigram interval model can be computed as described in [Con16]. Letting $v_i = \tau(e_i|e_{i-1})$ be the interval value of event e_i in the context of its preceding event e_{i-1}, the probability of a piece $\mathbf{e} = e_1, \ldots, e_\ell$ is computed as:

$$\mathbb{P}(\mathbf{e}) = \prod_{i=3}^{\ell} \mathbb{P}(v_i|v_{i-1}, v_{i-2}). \tag{2.2}$$

3. **Generation of pieces**

Once the coherence structure of the template piece and the statistical model of the corpus are built, a stochastic hill climbing optimization method is used to sample new notes into the coherence structure. The goal of the optimization is to generate high probability note sequences, since these sequences are supposed to retain more aspects of the melodies in the corpus. The process begins with a sequence of random notes that respects the coherence structure, and in each iteration of the optimization

[6] `http://bdb.bertsozale.eus/en/web/doinutegia/view/1016-lagundurikan-danoi-i`

process a random position is changed to try to get a higher probability piece. This process is iterated 10^4 times, generating pieces with the same coherence structure and rhythmic information of the template, but with different melodic information. The vocabulary of pitches allowed in the generation is obtained from the tonality and note range of the template piece.

In Figure 2.8 two melodies generated with this approach are shown. The piece presented in Figure 2.5 has been used as template, and it can be seen that even though the rhythmic content and the general structure are conserved in the generations, the melodies are different from the original.

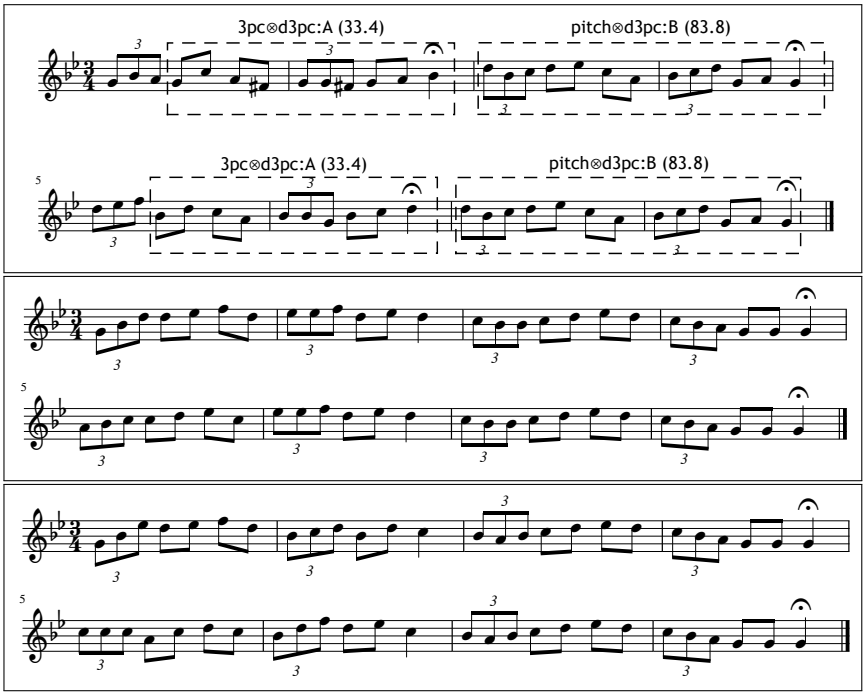


Figure 2.8: Score of template piece *Erletxoak lorean* and two melodies generated with it.

The generated melodies are overall smooth and they do not have big leaps that would sound weird or would make them difficult to sing. An evaluation of some melodies generated with this method was carried out in London in the final concert of the project Lrn2Cre8[7]. In this event three melodies were sung, one being the template piece and the other two melodies generated using that template. The audience was asked to fill a questionnaire where they would select which melody they thought was the template and how sure they were. In the top part of Table 2.1 the questionnaire they were given is shown. A total of 52 questionnaires (from approximately 100 distributed) was returned, and the obtained results can be seen in the bottom part of Table 2.1. The template piece was the third melody sung.

[7]https://web.archive.org/web/20160722213851/http://lrn2cre8.eu:80/

The results show that the generations were good enough to make the majority of the audience (55%) think they had been composed by a person.

Which piece is the original?
☐1 ☐2 ☐3

How confident are you on a scale of 1 to 5? (1=not confident, 5=very confident)
☐1 ☐2 ☐3 ☐4 ☐5

	transformation 1					transformation 2					original				
is original?	8 (15%)					21 (40%)					23 (44%)				
confidence	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
%	37.5	50	0	12.5	0	38	28.6	23.8	4.8	4.8	26	23.8	23.8	23.8	8.7

Table 2.1: Results obtained in the evaluation. The template piece was the third melody sung.

The paper that gives a full description of the method is listed below, along with some of the contributions made to the presented music generation method that are not described in the paper.

- [GC18] Izaro Goienetxea and Darrell Conklin. "Melody Transformation with Semiotic Patterns". In: *Music Technology with Swing*. Ed. by Mitsuko Aramaki et al. Cham: Springer International Publishing, 2018, pp. 477–488.

The following contributions have not been written in any paper, but they have been used to justify some of the decisions taken in the music generation process.

Adaptation of the pattern discovery method

Pattern discovery methods identify segments that are repeated through a symbolic representation of a musical piece or a corpus. Patterns are defined as sequences of event features, and a piece instantiates a pattern if the pattern occurs (one or multiple times) in the sequence: if the components of the pattern are instantiated by successive events in the sequence [CB08]. For the pattern discovery process needed in the presented work a pattern discovery algorithm named SPAM [Ayr+02] has been modified to identify similarities between and within segments. This is an algorithm that discovers all the frequent sequential patterns (patterns that occur more times than a given threshold) in a transactional database, specially efficient with large databases. Candidates are created with a depth-first search strategy and various pruning mechanisms reduce the search space.

To discover patterns within a single sequence, two modifications of the SPAM algorithm are necessary. First, SPAM is designed for multiple sequences, returning

support counts for the number of sequences matching the pattern, while in our work the repetitions need to be found within a single sequence. Second, SPAM allows discontiguous patterns, but contiguous patterns are preferred for the proposed process. To achieve both modifications first a suffix-based database from the multiple viewpoint representation of the piece is computed, as seen in Figure 2.9, where an eight event sequence is transformed into an eight entry database. Each s_i is the set of all viewpoint values of event e_i. This technique allows to effectively use SPAM looking for patterns only in the beginning of each database entry. The algorithm has also been modified to only look for contiguous patterns in the transformed suffix set. These modifications solve the issues that SPAM has for our pattern discovery tasks.

(1)	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8
(2)		s_2	s_3	s_4	s_5	s_6	s_7	s_8
(3)			s_3	s_4	s_5	s_6	s_7	s_8
(4)				s_4	s_5	s_6	s_7	s_8
(5)					s_5	s_6	s_7	s_8
(6)						s_6	s_7	s_8
(7)							s_7	s_8
(8)								s_8

Figure 2.9: Suffix-based database for an eight event sequence.

Effect of including contour patterns in coherence structures

In addition to the better known exact repetition and interval transposition relations, a contour relation is also introduced on the construction process of coherence structures. To justify this, an analysis has been performed to study the effect that the inclusion of these relations has on the created structures.

Contour transposition relations describe the relation between segments that share a similar melodic contour but do not instantiate the same interval pattern. Two types of contour relations are studied, five point contour and three point contour, defined by the viewpoints 5pc and 3pc respectively, where 5pc describes if the interval between two contiguous notes represents a leap, a scale step or it stays the same, and 3pc describes if the melodic contour between two notes goes up, down or stays the same. An example of this relation is shown on the score fragment of Figure 2.10. Though the two indicated phrases are clearly related, apparent in the score and to any listener, this is not by sharing an interval sequence, but rather an abstract contour sequence.

To analyse the convenience of using these kind of relations on coherence structures, the used corpus is studied to evaluate if they are able to improve the covering of the

[8] http://bdb.bertsozale.eus/en/web/doinutegia/view/137-begiztatua-nuen-euskaldun-makila

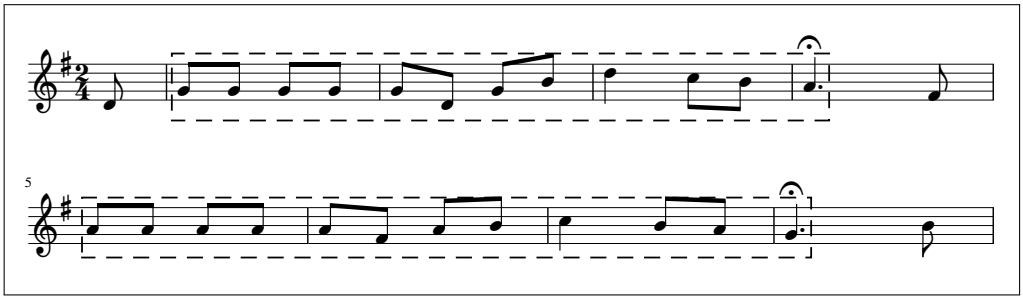


Figure 2.10: First two phrases of the melody *Begiztatua nuen*[8]. The two phrases are related by an abstract melodic contour relation and there is no transposition that carries one into the other.

pieces in it. To do so, two coherence structures have been built for each piece: one using only pitch and interval information and one also including contour information. The structures created from each piece have been compared to determine if the addition of contour information builds structures that actually cover the pieces better. To compare the coherence structures created with and without contour information, two features have been used: compression (which is described below) and covering percentage.

Compression

It is defined as the number of bits needed to represent each piece and the coherence structure built from it. It is computed using the following equation,

$$Comp = L(P) + L(D|P) + VT$$

where $L(P)$ is the number of bits needed to represent the patterns in the structure, $L(D|P)$ is the number of bits to specify the data with respect to the patterns and VT is the number of bits needed to specify the vocabulary of each piece. Lower costs mean better piece compression, thus, better coherence structure. $L(P)$ is computed with a zero-order model of the corpus, which gives the probability of finding each component of the pattern in a viewpoint representation of the corpus.

To compute the number of bits needed to specify the data with respect to the patterns ($L(D|P)$) different features have to be taken into account, like the events in the different pattern types and the events that are not part of any pattern. The events of the melody that are not covered by any pattern of the semiotic structure have a representation cost of $\log_2 N$ bits, where N is the number of different pitches in the vocabulary of the piece. For the events included within the patterns the cost change depending on the type of pattern they belong to. As pitch and int patterns define the pitch number of their events, representing them does not have any cost. On the other hand for 5pc patterns, the representation cost of the events change depending on the pattern features. For features su or sd the representation cost is $\log_2 2$, for lu

or ld the cost is $\log_2 k$, where k is the number of possible pitches for that contour. Since feature eq does not introduce a new note it has no representation cost. For the events on 3pc patterns, features u or d have a cost of $\log_2 k$ bits, where k is the number of possible pitches for that contour, and feature eq has no representation cost.

Once the patterns and the data with respect to the patterns are represented, the vocabulary of each piece (VT) must also be represented. It is computed as:

$$VT = \log_2 \binom{33}{N}$$

where N is the number of pitches on the vocabulary of the piece, and 33 is the number of different pitches in the corpus.

Structure comparison

Table 2.2 shows the values of the mean covering percentage and mean compression values of the coverings of all the pieces of the corpus, using only pitch and int patterns, and the values of the coverings including contour patterns. The number of pieces of the corpus that cannot be covered by any significant pattern is also shown.

Viewpoints	Covering Percentage	Compression	Uncovered Pieces
pitch&int	58.12%	2.72	46
All viewpoints	59.70%	2.71	34

Table 2.2: Comparison between coverings of the pieces in the corpus using only pitch and interval information and coverings that include contour patterns.

It can be seen that including contour patterns on the covering the mean compression of the corpus is lower than when only pitch and interval information is used. The mean covering percentage is also higher when all the pattern types are used, and in addition, less pieces are left uncovered.

On the other hand, it has to be mentioned that contour patterns are not used on the covering of all the pieces in the corpus; only 247 pieces have at least one contour pattern on its semiotic structure. This might be caused by the chosen covering strategy, which benefits pitch and interval patterns. To confirm that the amount of pieces that use contour patterns is indeed a consequence of the covering strategy, a pattern length based covering method has also been tried. In this covering process the list of patterns is sorted according to their length instead of computing their interest. Using the pattern length based covering, contour patterns are used on the coherence structure of 1251 pieces (52.59% of the corpus), proving that the chosen

covering method has a great impact on the type of patterns that are used to cover the pieces.

For the 247 pieces that use contour patterns on their covering a comparison between the coherence structures with only pitch and interval and the structures that include contour information has been made to study its effect in these pieces. The mean covering percentage and mean compression values are shown in Table 2.3.

Viewpoints	Covering Percentage	Compression
pitch&int	45.34%	2.93
All viewpoints	60.53%	2.84

Table 2.3: Comparison between covering using only pitch and interval information and covering that includes contour information, on the 247 pieces that use contour patterns on the covering.

It can be seen that including contour patterns on the semiotic structures significantly improves the covering percentage of the pieces, and the compression is also lower, which makes the semiotic structures better than those that only use pitch and interval information.

Sampling with Random Walk

In addition to stochastic hill climbing a second sampling method has been tried to sample new notes into the coherence structure. The chosen method is the iterative random walk, which has been used in other music generation works [WC16; Her+14]. The method consists in performing a random walk sampling process 10^6 times, measuring the probability of each generated melody with Equation 2.2, and saving the melody with the highest probability. The random walk process samples notes from left to right, and as in the first step of stochastic hill climbing method, every time a complete pattern is instantiated, all its future occurrences are also instantiated to preserve the coherence structure.

In Fig 2.11 two melodies generated with iterative random walk can be seen, as well as the used template piece. They are correct generations, they respect the coherence structure and have not melodically weird segments, even though they tend to have more interval leaps than the ones generated with stochastic hill climbing.

100 new melodies have been generated with both iterative random walk and stochastic hill climbing to compute the cross-entropies of the pieces generated with both methods. Cross-entropy is defined as $-\log_2 \mathbb{P}(e_1, \ldots, e_\ell)/\ell$, and it is used to work in a logarithmic scale and be able to compare the results better. Pieces with lower cross-entropy are more similar to the corpus. The pieces generated with iterative

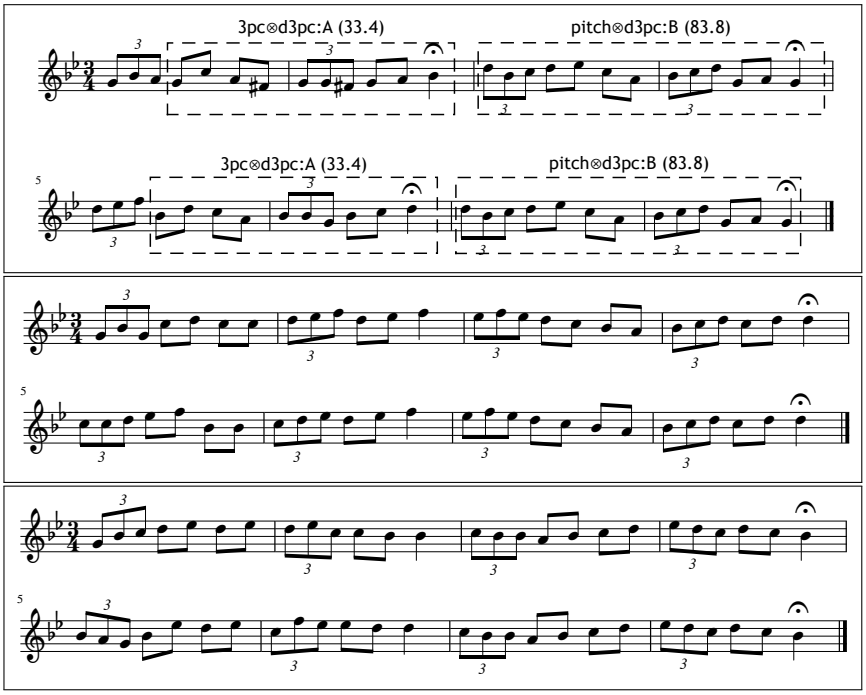


Figure 2.11: Score of the piece *Erletxoak lorean* (top) and two melodies generated using it as template and iterative random.

random walk have a mean cross-entropy of 2.22, while the melodies generated with stochastic hill climbing have a mean cross-entropy of 2.09.

In general, it can be concluded that using iterative random walk melodies similar to the ones created with stochastic hill climbing can be generated, even though the latter are slightly better according to their lower mean cross-entropy.

2.3.4 Rhythmic structure in template

An extension of the melody generation method of Section 2.3.3 has been proposed, where in addition to the melodic information, rhythm is also generated. In Figure 2.12 a diagram that describes the method is presented, where it can be seen that in besides the melodic structure and statistical model, a rhythmic coherence structure and a rhythmic model are also added.

In order to decide how to add the rhythmic information to the melodic generation method, the pieces in the corpus have been studied. It has been seen that in many of the pieces rhythmic repetitions happen in segments that are not completely related melodically. In Figure 2.13 an example of a piece can be seen where four segments have been highlighted. These segments have exactly the same rhythmic sequence, but no melodic viewpoint exists that is able to describe their melodic relation. Considering that in many pieces rhythmic and melodic relations do not

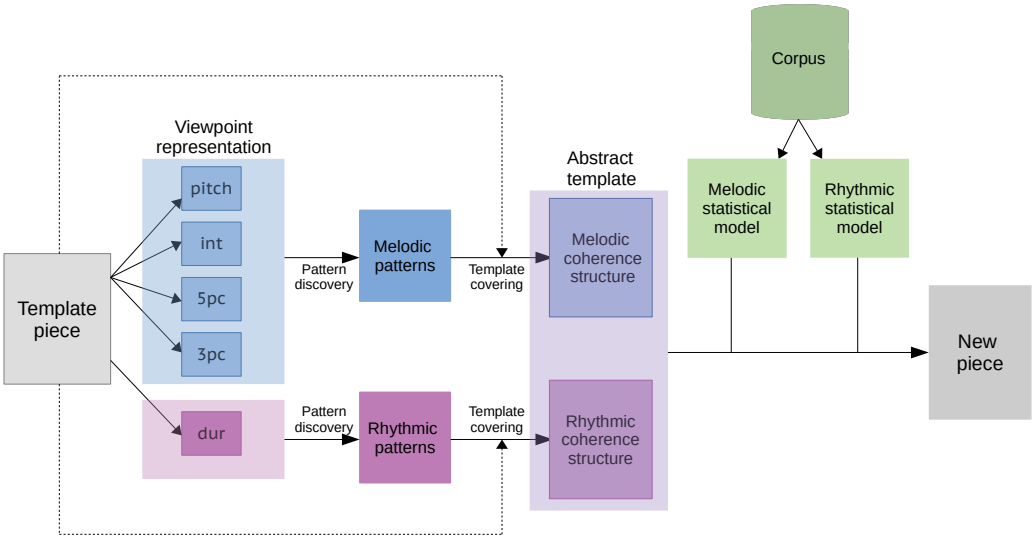


Figure 2.12: Diagram of the music generation method presented.

coincide, independent structures have been built for melodic and rhythmic relations of each template piece.

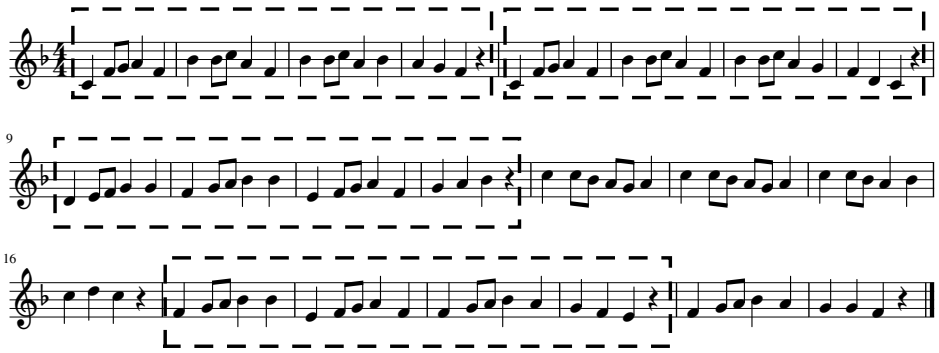


Figure 2.13: Score of the melody *Abiatu da bere bidean* where the main rhythmic pattern is highlighted.

To build the rhythmic coherence structure, patterns are discovered within the duration (dur) representation of the template piece, since only exact rhythmic repetitions have been considered in this work.

It has been noticed that in many of the pieces of the corpus the most significant rhythmic patterns are long patterns, which have other significant patterns within them. In Figure 2.14 an example of a long pattern which has shorter patterns within it can be seen. These so called nested patterns are also important when building the coherence structure of a piece, and have been considered in other music generation works [Col+14].

To discover this kind of nested patterns, the list of patterns that have been discovered in the template piece is sorted according to their interest, and the patterns in the

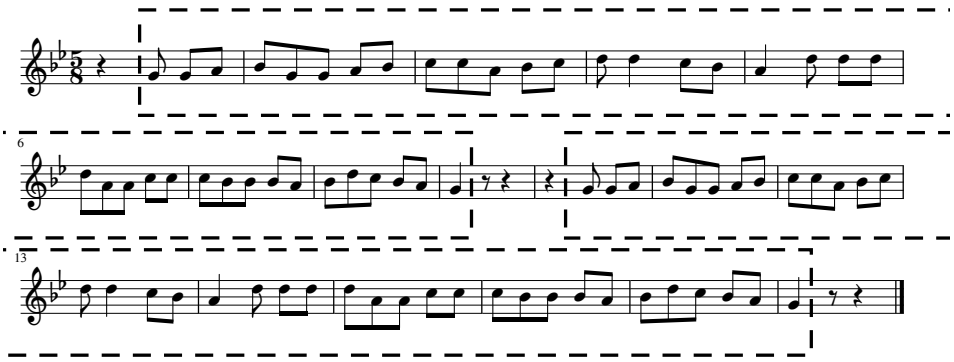


Figure 2.14: Score of the melody *Neure lagunak lagun zakidaz* where its most interesting pattern is highlighted.

list are used to cover the regions where the long patterns occur. The interest of rhythmic patterns is computed the same way as the interest of the melodic ones. This process is repeated until all the nested patterns in the piece are discovered, making it possible to nest patterns within other nested patterns, creating a pattern hierarchy.

For example, in Figure 2.15 the rhythmic structure of the piece *Neure lagunak lagun zakidaz* after the nested pattern discovery process is shown, where patterns of different nesting level are represented with different colours.

It can be seen that the piece has one main rhythmic pattern that is repeated twice in the piece, and even if the whole piece is covered by this pattern, it does not provide enough information about how the rhythm is structured through the piece. Considering that the only information that this pattern represents is that the piece has a long segment that is repeated twice, the patterns that occur within that principal pattern are also discovered and represented in the figure in purple.

The nested pattern discovery has also been applied in this purple pattern, in which another nested pattern of 7 elements has been discovered, represented in Figure 2.15 in green. In this example a pattern hierarchy of three levels has been created.

As melodic and rhythmic information are treated independently, a rhythmic statistical model is built in addition to the melodic model. As well as for the melodic model, a trigram model of duration is built.

Once the melodic and rhythmic coherence structures and both statistical models are built the sampling strategy needs to be chosen. Since in bertso melodies a fixed number of notes depends on the type of strophe, the melodies generated with this method should conserve the number of notes of the piece used as template. To make sure of that, a sampling strategy has been designed that is combined with

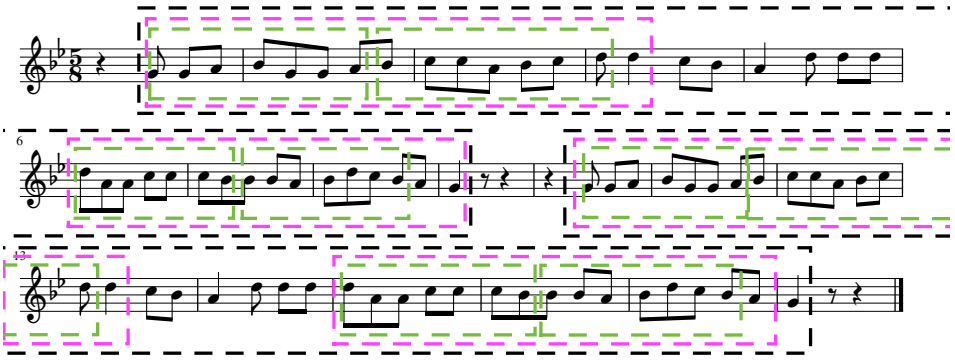


Figure 2.15: Nested pattern (in purple) found within the principal pattern (in black) in the melody *Neure lagunak lagun zakidaz*.

the stochastic hill climbing optimization method. The first step of stochastic hill climbing is to create a random initial piece that conserves the coherence structure of the template piece. In the proposed rhythmic sampling strategy, the duration of the notes of each pattern of the coherence structure is randomized instead of randomly sampled. This guarantees that the number of notes and total duration of the random piece are the same as those of the template piece.

In the rhythmic optimization steps, every time a random position is selected, the pattern that the position belongs to is identified, and a random value of the rhythmic vocabulary is selected to subtract its value to the note of that position. This vocabulary consists of the durations that are used in the template piece. In order to conserve the total duration of the piece, the subtracted value is then added to another random position within the same pattern. The changes need to be propagated to all the other instances of the pattern in order to conserve the rhythmic coherence. The probability of the rhythmic sequence is computed according to the rhythmic statistical model, and if the new probability is higher than the last saved one, the changes are allowed and the new sequence is used in the next iteration. If it is not higher the last changes are discarded. Since in many bertso melodies rests act as phrase boundaries, their positions and lengths have been fixed and they are not changed through the sampling phase.

To generate the final pieces, first the randomized rhythmic information and the random melodic line are created. On the optimization phase, first the rhythmic sampling process is iterated 10^6 times, getting a high probability rhythmic sequence, and then the melodic optimization is applied 10^4 times.

In Figure 2.16 an example of two pieces generated with this method are shown, for which the piece shown in Figure 2.13 has been used as template. It can be seen in both melodies that even though the same coherence structures and statistical models have been used, the generated pieces are different between them.

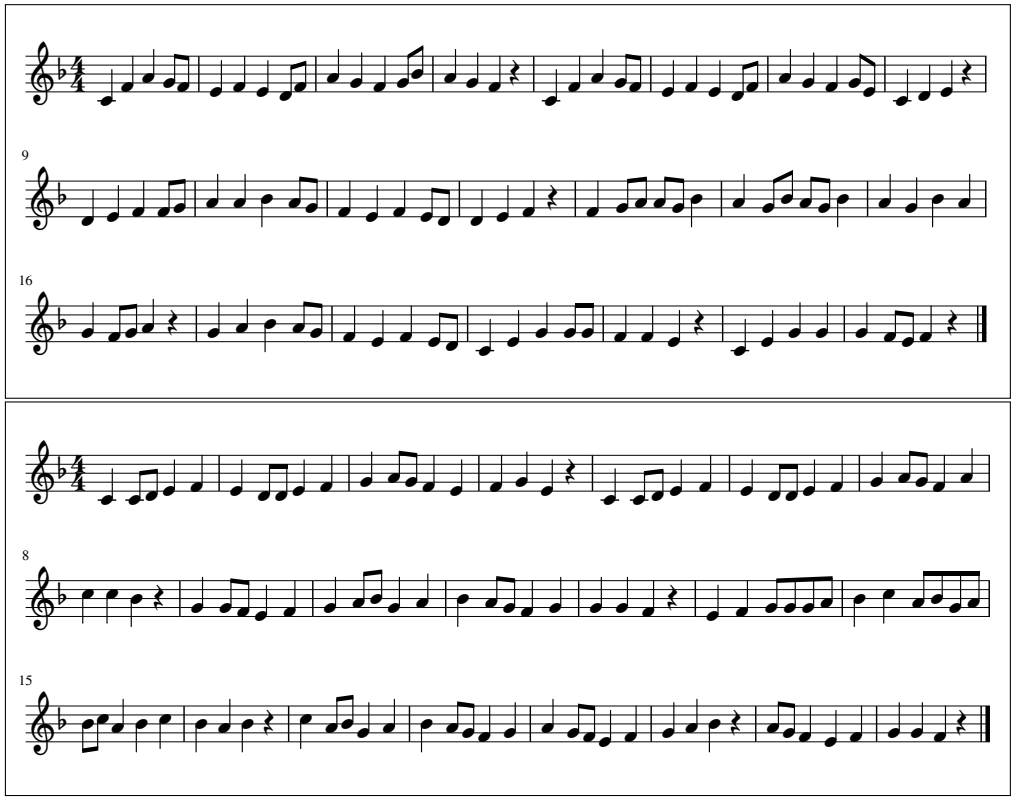


Figure 2.16: Examples of pieces generated from template *Abiatu da bere bidean*.

It should be mentioned that although both generations are different and respect the melodic and rhythmic coherence of the template piece, there is not a really big difference between the rhythmic content of the pieces. This happens because of the conservative rhythmic sampling strategy chosen to create pieces that conserve the number of notes and total duration of the template. Since the allowed rhythmic vocabulary is taken from the template, and in this case it is not very varied, there are not many possible combinations for the final sequences. On the other hand, both examples are correct melodies that could be used to sing bertsos.

An evaluation process has been carried out in order to evaluate if the pieces generated with this method are perceived as *natural* melodies, that is, melodies that could have been composed by a human. To do so, five pieces have been randomly selected from the corpus, and they have been used as template to generate a new melody from each one. Those ten pieces, the templates and the generations, have been sung and recorded (by me) using some lyrics that fit the metrics of the melodies. They have been distributed in two sets of melodies, where a generated melody and its template cannot be in the same set.

31 researchers and professors of the university have taken part in the evaluation process, where one of the two melody sets have been assigned to each participant. In the evaluation process, first a training phase was performed where the participants

listened to MIDI versions of all the pieces in their melody set, an then they listened to the five recordings. They were given a questionnaire where they needed to note, for each melody they heard, if they thought it was part of the corpus or generated, and how sure they were of their answer. Then, they had to listen only to the generated ones to evaluate how much they liked them from 1 (not at all) to 10 (a lot).

The results showed that even though some participants were able to identify some of the generated melodies as possibly generated, they were overall confused and not able to clearly distinguish between original and generated pieces. The results of the second part of the evaluation, where participants needed to give a score from 1 to 10 to the generated melodies of their set, show that in general they liked the new pieces.

The method and the evaluation results are described in detail in the paper below:

- Statistics based music generation approach considering both rhythm and melody coherence. **Submitted** to IEEE Access journal

2.4 Conclusions and future work

In this chapter the background of the work done in the field of automatic music generation is presented, as well as the contributions of our work to the field. Even though the topic is getting more popularity and research groups in big companies like Google and Spotify are working in it, there are still many challenges that need to be faced. The generation of structured coherent music is one of those challenges that this work tries to solve. The contributions presented in this chapter are mainly focused on the automatic generation of coherent bertso melodies. A corpus of these melodies is used to build a statistical model that guides the generation in order to get melodies that are similar to those in the corpus.

A first method of coherent music generation is presented based on the use of the semiotic description of the notes in a template piece to guide the generation. This method assigns different semiotic labels to different notes in the template piece, and these labels are then used to sample different notes in them. As the labelling can be too restrictive and there may not exist many different generations that respect the coherence structure, a melodic reduction method is applied to leave some of the notes of the piece unconstrained.

The method has evolved to describe the relations between segments of the template piece instead of the relations between notes. Relations of several abstraction levels have been considered, instead of only using the repetition and transposition relations.

A complete method for generating melodic content is presented, which includes the building of an abstract template for which the discovery of melodically related segments and selection of the most important ones are performed. This abstract template is then used to guide the sampling of new notes within the template with a statistical model built from the corpus.

The abstract template has been extended to also include the rhythmic information of the template piece. A rhythmic representation of the template piece is made and a pattern discovery method is applied to discover the repeated patterns in the representation. Since some very long patters were discovered, a method to discover nested patterns is also added. To sample new notes in the abstract template, the melodic and rhythmic coherence are considered and, in addition to the melodic statistical model, a rhythmic model is also built. Some of the pieces generated with these methods have been evaluated, and the results show that participants were overall confused and not able to clearly distinguish if a piece is an actual bertso melody or a generated one.

Several paths have been identified as future work. The pattern discovery method used in this work discovers patterns on contiguous elements, it does not allow gaps within the patterns. This should be fixed to be able to discover patterns that are not exact repetitions, but are clearly related. A new pattern discovery method that discovers feature set patterns would also be beneficial, to represent the template piece once using all the viewpoints and discover patterns that combine information of different abstraction levels.

Since in the method presented in this work only monophonic pieces have been generated, extending it to generate polyphonic pieces should be considered, as well as combining the use of coherence structures of a template with some recent deep learning methods to generate coherent music. Finally, in order to make the method 100% automatic, generating the coherence structures automatically instead of learning them from a template piece should also be considered.

The music generation method that includes the rhythmic generation is currently being used in a regional project named *Remembering Reminding*. The goal of the project is to create a mobile application that will include musical therapy tasks targeted towards people with dementia. Many studies have been carried through that suggest that activities related to music are beneficial for people with dementia. Music can improve their mood, it can make them recall past moments related to the music they are listening to... it can activate memories even on people with diseases like Alzheimer's. The activities included on the application will be related to the identification of similar melodies or a melody they already know that at some point has been changed using the methods presented in this work.

3 Supervised Classification

3.1 Introduction

Supervised classification is a machine learning task which is used to solve problems in a wide variety of domains. It is based on the use of *a priori* knowledge that is described by using a set of well labelled instances, which are referenced as *training data*. Each of those instances contain several relevant features that describe an object and its corresponding class.

Let training data TR=$\{x_i,\theta_i\}_{i=1}^{N}$ denote a set of N well-labelled instances, where x_i represents the i-th individual feature vector and θ_i represents the class the individual belongs to. In the particular case of the C-class problem, being $\theta \in \{1 \dots C\}$ the class label is commonly defined as an integer. Table 3.1 illustrates an example of training data.

Instance	X_1	X_2	...	X_L	Class
1	x_{11}	x_{21}	...	x_{L1}	θ_2
2	x_{12}	x_{22}	...	x_{L2}	θ_c
...	...	...	...	...	...
N	x_{1N}	x_{2N}	...	x_{LN}	θ_1

Table 3.1: Example of training data

Based on this training data a "general rule", also known as classifier, is created which is able to assign one of the previously defined C classes to new unknown instances.

Several ways exist to build these general rules. For example, distance-based methods are based on the idea that assume that the closer two instances are, the more similar they are, and K-Nearest Neighbour (K-NN) is the most popular example of this method. Decision trees divide the classification space into smaller and smaller areas and a class is assigned to the instances that belong to each area. Rule-based classifiers generate a set of rules to explain the feature space, concentrating on one class at a time, and disregarding what happens to the other classes. Bayesian classifiers use Bayesian reasoning in order to assign the most likely class value to each new instance. Neural Networks process information in a similar way that the human brain does, using networks composed of a large number of highly interconnected

processing elements(neurones) that work in parallel to solve a specific problem. Finally, Support Vector Machines (SVM) are the most common Kernel based method and perform classification by finding the hyperplane that maximizes the margin between two classes.

Once trained, one way to evaluate the classifiers is computing their accuracy, which indicates the percentage of instances that they classify well. To do so, a test set is used, which is composed of instances that are not part of the training set. In supervised classification problems where all the instances in the available data are used to train the classifier, there may not be any testing data to test its estimation error. To compute the accuracy that the classifiers would have in cases like these, some strategies known as Classification Error Estimators are used.

One of the Classification Error Estimator techniques is the k-fold cross-validation, where the dataset is randomly split into k mutually exclusive subsets. Each of these partitions is known as *fold*, and all the folds have approximately the same size. The classifier is trained and tested k times; each time, $k-1$ folds are used to as training data, while the remaining fold is used as testing set. The 5x2-fold cross-validation is also commonly applied as recommended by Dietterich [Die98], but in this work a 10-fold cross-validation is used as recommended by Kohavi [Koh95].

When a new supervised classification method is proposed, it is compared to other existing methods, but comparing only their accuracies is not enough to decide which one performs better. To make sure that actual statistical differences exist between classifiers, statistical tests are carried out. These tests proceed from the null hypothesis, which is the assumption that there is no statistical difference between the results obtained by the different classifiers. The rejection of this null hypothesis means that there exist statistical differences between the tested classifiers. There are different tests that can be used, but in this work only two are used: Wilcoxon test [Wil92], which is used to compare two classifiers over several datasets, as recommended by Demšar [Dem06], and the Friedman test and Iman-Davenport extension [ID80], used to compare multiple classifiers over multiple databases, as suggested by García et al. [Gar+10].

In this chapter a contribution named PSEUDOVO is presented. It is a supervised classification method which has two main components: Dynamic Classifier Selection and Class Binarization. Both components are described below, before the actual contribution is presented.

3.2 Dynamic Classifier Selection

Even though many different classifiers exist, not all of them are experts in classifying all the new instances they find; different classifiers usually make errors on different instances. The Classifier Selection strategies are based on this idea and they try to select the best classifier in a particular problem, and these classifier selection processes can be static or dynamic.

In the Static Classifier Selection (SCS) strategies, the local regions and the selected classifier for each problem are statically defined in the validation phase. Dynamic Classifier Selection (DCS) methods, on the other hand, try to dynamically select the best single classifier for each new instance to be classified, and they have been used in several real life problems. They are used in credit scoring problems [Xia+16] for example, where several DCS methods are applied. On the other hand, Tsymbal et al. [Tsy+08] proposed a dynamic ensemble method to deal with the antibiotic resistance problem.

Some of the most popular methods are Overall Local Accuracy (OLA) and Local Class Accuracy (LCA) presented by Woods [Woo+97]. The former is divided into two steps; validation and classification. The aim of the validation step is to see which base classifier gives correct or incorrect predictions to each training instance. To do so each training instance is classified by different base classifiers and the results are recorded. These results are then used to compute the local accuracy in the classification step, where every time a new instance arrives its local region is defined by obtaining its K nearest neighbours. The local accuracy of each base classifier for these neighbours is computed using the results recorded in the validation step and the classifier with the highest accuracy is selected to classify the new instance.

LCA is similar to OLA, and it also evaluates the competence level of each individual classifier in the local region, but the competence of each base classifier is calculated based on its local accuracy with respect to some output class.

Giacinto and Roli [GR99] also extended Woods' work proposing two new approaches: A Priori and A Posteriori. A Priori strategy considers the probability of correct classification of base classifiers in the local region, but instead of using only the labels assigned to each new sample, it considers the vector containing the posterior probabilities for each class. The second proposed approach, A Posteriori, is a mixture of LCA with A Priori.

3.3 Class binarization

Some classifiers, like SVM, are designed to classify between two classes. However, many real world problems are multi-class problems; the classifier needs to distinguish between more than two classes. In order to deal with multi-class classification, binarization techniques are widely used, which decompose the original multi-class problem into several easier to solve binary classification problems, that can be faced by binary classifiers.

Class binarization process consists of two main steps; decomposition and combination. In the decomposition step the original problem is decomposed into several binary sub-problems. Even though different decomposition techniques can be found in the literature, there are three main strategies:

- One-vs-One (OVO)[FÖ2a]: The problem is decomposed into a series of two-class problems, one problem for each pair of classes.

- One-vs-All (OVA)[Ana+95]: A sub-problem is created for each class, where the classifier learns to distinguish between that class and all the other classes.

- Error Correcting Output Codes (ECOC)[DB95]: The problem is decomposed into a larger number of binary sub-problems, where each classifier is trained on a two meta-class problem, and each meta-class consists of a combination of some of the original classes.

To represent how the classes are grouped within each sub-problem in each of the mentioned binarization methods, the code-matrix is used. Figure 3.1 shows the code matrices of each of the decomposition methods, where the binarization of a four class problem is shown. +1 and -1 represent the classes that are taken into account in a sub-problem and 0 represents a class that is not used in it.

$$\begin{array}{c} \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{array} \begin{array}{c} \begin{array}{cccccc} f_1 & f_2 & f_3 & f_4 & f_5 & f_6 \end{array} \\ \begin{pmatrix} +1 & +1 & +1 & 0 & 0 & 0 \\ -1 & 0 & 0 & +1 & +1 & 0 \\ 0 & -1 & 0 & -1 & 0 & +1 \\ 0 & 0 & -1 & 0 & -1 & -1 \end{pmatrix} \end{array} \qquad \begin{array}{c} \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{array} \begin{array}{c} \begin{array}{cccc} f_1 & f_2 & f_3 & f_4 \end{array} \\ \begin{pmatrix} +1 & -1 & -1 & -1 \\ -1 & +1 & -1 & -1 \\ -1 & -1 & +1 & -1 \\ -1 & -1 & -1 & +1 \end{pmatrix} \end{array}$$

$$\begin{array}{c} \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{array} \begin{array}{c} \begin{array}{ccccc} f_1 & f_2 & f_3 & f_4 & f_5 \end{array} \\ \begin{pmatrix} -1 & -1 & +1 & +1 & -1 \\ +1 & -1 & -1 & +1 & +1 \\ +1 & +1 & +1 & -1 & -1 \\ -1 & -1 & +1 & -1 & +1 \end{pmatrix} \end{array}$$

Figure 3.1: Code matrices for OVO (top left), OVA (top right) and ECOC (bottom)

3.3.1 OVO

In this work the One-vs-One binarization technique, also called Pairwise Classification has been used. This scheme divides a C class multi-class problem, $\theta_1 \dots \theta_C$, into $C(C-1)/2$ two-class sub-problems, where in each sub-problem only two classes are considered for classification, not taking into account the remaining $C-2$ classes.

When a new instance is being classified using this method, all the sub-problems return a prediction of its class, and all these outputs are then combined. To do so several strategies have been developed. The simplest one is the majority vote [FÖ2a; Fri96], where each sub-problem returns a vote, and the class with the largest amount of votes is predicted. An extension to this method is the Weighted Voting, where the confidence of the classifiers of each sub-problem is used to assign a weight to each output [HV10]. On the other hand, the method by Hastie and Tibshirani [HT98] tries to estimate the best approximation of the posterior probability of the class, starting from the posterior probabilities of the pairwise sub-problems.

Although class binarization techniques seem to be simple, they are effective dealing with multi-class problems, and they are applied in several real world classification problems. For example, Liu et al. [Liu+17] proposed a novel combination of OVO and OVA to classify intentions based on the brain signals obtained with a near-infrared spectroscopy, and Zhou et al. [Zho+17] applied OVO for predicting listing status of companies in China.

It should be mentioned that the OVO binarization technique has also some drawbacks:

1. Unclassifiable regions: Ties might happen between the classes that the different classifiers predict, not returning a winner case. Tie-breaking techniques would be necessary in these cases.

2. Number of classifiers: OVO creates a sub-problem for each pair of possible classes. Creating so many sub-problems may lead to having many classifiers that do not take into account the actual class of the instance, returning wrong predictions. On the other hand, the created sub-problems are simpler, hence, easier to build.

3. Weak classifiers: It is not easy choosing an optimal classifier for the dataset that is accurate in the classification of all the sub-problems.

Since these issues might have a negative effect in the classification accuracies, some approaches have been developed to try to solve them.

As a solution to the issue of unclassifiable regions Platt et al. [Pla+00a] proposed the use of Decision Directed Acyclic Graphs (DDAG). This method builds a rooted binary acyclic graph where in each node the classifier discriminates between two classes. Once the classifier gets to a leaf node, the class returned by the sub-problem that belongs to the leaf node is returned.

To tackle the issue of the high number of sub-problems that are produced in OVO different works have been developed. Some works proposed to combine OVA and OVO [GPOB06; KB03]; in these approaches, first OVA was applied to obtain the two classes with the highest confidence level, then OVO was applied with these two classes. In this way they reduced the number of sub-problems to $C + 1$. Other approaches proposed to dispose the sub-problems in a hierarchical structure such as graphs [Pla+00b; KU02; Men+16] or binary trees [FL06].

Several approaches have also been developed to try to solve the issue of the weak classifiers. Some works try to select the best base classifier in each sub-problem [Sze+09a; Arr+14] while others try to select the hyper-parameters of SVM classifier for each sub-problem of OVO. Many of these approaches use evolutionary algorithms, like Lebrun et al. [Leb+07] and Liepert [Lie03] that use Genetic Algorithms or Souza et al. [Sou+06], who use Particle Swarm Optimization methods.

3.3.2 Dynamic Classifier Selection and OVO

Several works try to solve the drawbacks of the number of classifiers and weak classifiers of OVO, combining it with Dynamic Classifier Selection methods.

Dynamic Sub-problem Selection in OVO

In order to reduce the number of sub-problems of OVO Galar et al. [Gal+13] and Bagheri et al. [Bag+12] proposed a new method, which intended to avoid the non-competent sub-problems. To do so, they applied Dynamic Classifier Selection methods: when an instance to be classified arrives, its K nearest neighbours are obtained, being the value of K three times the number of classes. Then OVO is applied considering only the classes in the neighbourhood, while the remaining classes are ignored. In this way for each new instance the sub-problems that take part in the final decision are selected dynamically.

Galar et al. [Gal+15] updated their sub-problem selection to Distance-based Relative Competence Weighting combination (DRCW-OVO). This method weighs the competence of the outputs of the selected classifiers depending on the distance of the new sample to the nearest neighbours of each class in the problem.

Dynamic Classifier Selection in OVO

Mendialdua et al. [Men+15] introduced a novel approach called DYNOVO that adapts the Dynamic Classifier Selection methods to OVO: when a new instance to be classified arrives, the multi-class problem is decomposed with OVO and for each sub-problem, independently, a Dynamic Classifier Selection method is applied to select the best base classifier.

Among the different Dynamic Classifier Selection methods they chose to use Overall Local Accuracy (OLA) and its extension Distance Weighted-Overall Local Accuracy (DW-OLA). Both methods work similarly: given an instance to be classified they select the classifier with the best behaviour in the local region of the instance. To get the local region of the instance the well known K-Nearest Neighbour algorithm is used.

They considered that OLA selected better classifiers when the class distribution of the local region were well balanced. That is why they proposed to use K-NNE instead of K-NN to obtain the local regions. K-NNE [Sie+11] is a version of K-NN, where the K nearest neighbours of each class are obtained independently.

Recently Zhang et al. [Zha+17] have proposed an extension of DYNOVO that combines OVO with an improved DES (Dynamic Ensemble Selection) procedure, dynamically selecting a group of competent heterogeneous classifiers in each sub-problem for each query sample.

3.4 Contributions

In this work an extension of DYNOVO [Men+15] has been presented as contribution, which reduces the number of sub-problems that take part in the classification of new instances. As mentioned before, DYNOVO tries to dynamically select the best base classifier for each sub-problem of OVO, using a Dynamic Classifier Selection method. Following Galar et al. [Gal+13] or Bagheri et al. [Bag+12], in this work the use of DCS is proposed to also reduce the number of sub-problems of OVO, avoiding the non-competent sub-problems. The new approach has been named PSEUDOVO, which stands for Problem SElection Under Dynamic One Versus One.

3.4.1 PSEUDOVO

The new approach is divided into two steps; for each test sample to be classified, first the sub-problems that take part in the final decision are selected, and then the base classifier for each of those sub-problems is chosen.

Sub-problem selection

To select the sub-problems that will be taken into account in the classification of each test sample, the approach proposed by Galar et al. [Gal+13] is used. When an instance to be classified arrives, its local region is obtained using K-NN (being the value of K three times the number of classes). Then the sub-problems with the classes within the local region take part in the final decision while the sub-problems with a class that is not in the local region are ignored.

Figure 3.2 illustrates how the sub-problems are selected to classify 2 new samples (▲and ●) in a 5-class problem. After their local regions are obtained selecting their 15 nearest neighbours, the sub-problems with class 5 are ignored to classify ▲ and the sub-problems with classes 2 and 4 are ignored to classify ●.

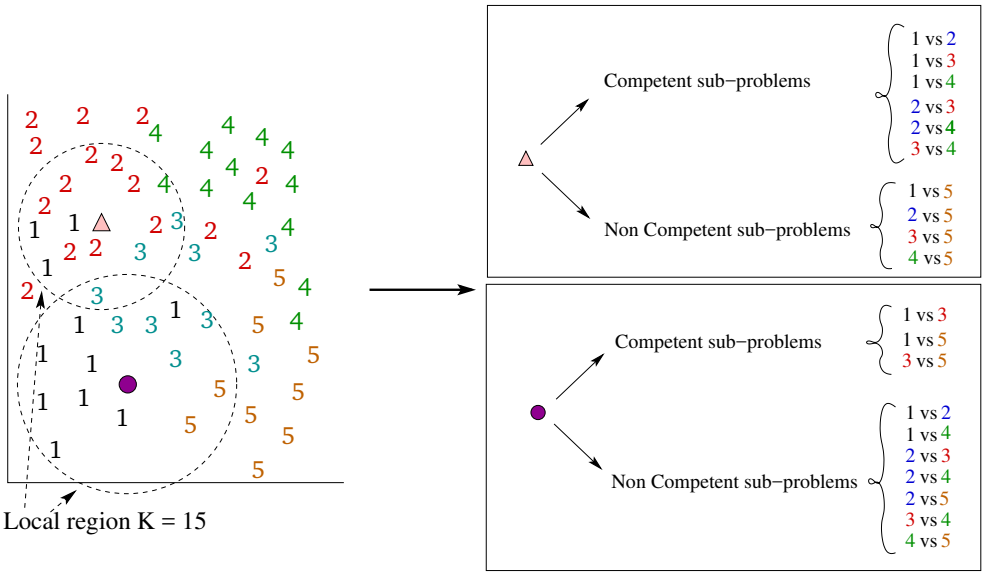


Figure 3.2: Example of how the competent sub-problems are selected for two new samples

Selection of the base classifiers for each sub-problem

Once the sub-problems that take part in the final decision are chosen, the base classifiers for each of those sub-problems are selected. To do so DYNOVO is applied. In each sub-problem, independently, Distance Weighted-Overall Local Accuracy (DW-OLA) is applied with a little change; instead of K-NN, K-NNE is used to get the local region. Then the local accuracy of each base classifier is calculated for the instances of the local region, where they receive weights depending on their distance to the test sample. Finally the base classifier with the highest local accuracy is used to build the classifier of the sub-problem.

Following with the previous example, Figure 3.3 illustrates how the base classifiers are selected in each sub-problem for candidate •. First the local region is obtained getting the 3 nearest neighbours of each class. Then the local accuracy of the different base classifiers is calculated. Finally the classifier with the highest local accuracy is selected. For example, in the sub-problem 1-vs-5 C_1 classifier is selected since it performs better in the closest instances of the test sample, even though C_2 classifies correctly more instances of the local region.

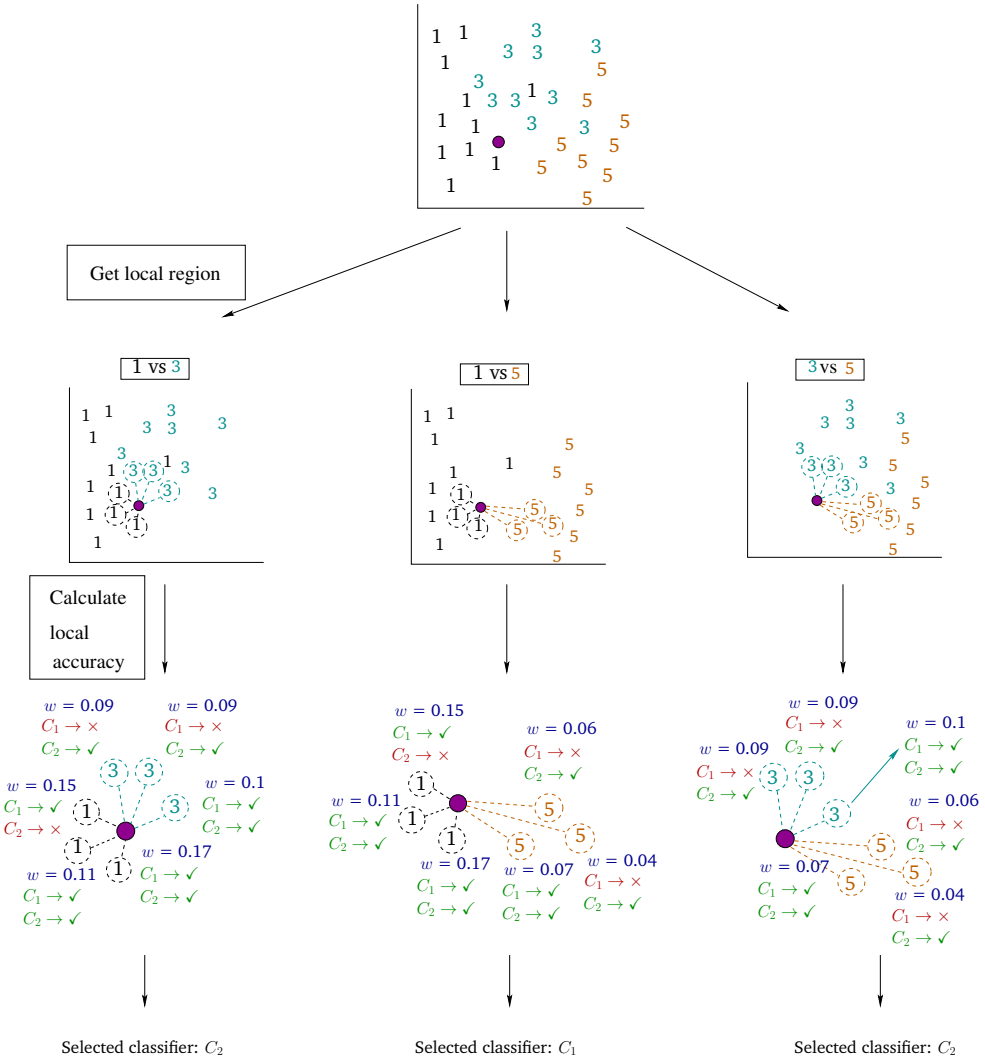


Figure 3.3: Example of how the base classifiers are selected for each sub-problem

To study the performance of the method it is compared to other state-of-the-art methods over a set of 25 datasets selected from the UCI repository [DKT17]. A database called *composers* has also been used. It is a database composed of a global feature representation of pieces of three classical composers, better described in Chapter 4.3.3.

A summary of the used datasets can be found on Table 3.2, where the number of instances (#Cases), the number of attributes (#Attributes) and number of classes (#Classes) of each database is presented.

Database	#Cases	#Attributes	#Clases
anneal	898	38	5
balance-scale	625	4	3
car	1728	6	4
cmc	1473	9	3
composers	1138	12	3
dermatology	366	33	6
ecoli	336	7	8
glass	214	9	7
iris	150	4	3
imgsegment	2310	19	7
lymph	148	18	4
nursery	12960	8	5
optdigits	5620	64	10
page-blocks	5473	10	5
pendigits	10992	16	10
satimg	6435	36	7
solar-flare1	323	12	6
solar-flare2	1066	12	6
soybean	683	35	19
vehicle	846	18	4
vowel	990	13	11
waveform-5000	5000	21	3
wine	178	13	3
winered	1599	10	6
yeast	1484	8	10
zoo	101	17	7

Table 3.2: Summary of the databases used in this work.

Five different base classifiers from a software package for machine learning called Weka [Hal+09] have been used in the classification steps: J48, SMO, JRip, Naive Bayes and Bayesian Networks. Classifiers of diverse natures have been selected to give variability and reliability to the experimental phase.

The classification accuracy of each binarization method is obtained by means of a stratified 10-fold cross-validation. Some of the methods also need a validation process that consists of a 5-fold cross-validation for each training fold.

In this work DCS methods are used in two phases of the process; the selection of sub-problems that participate in the final prediction and the dynamic selection of the base classifier for each sub-problem generated in OVO. The methods to select the base classifiers are based on K-NNE to define the local regions of the samples to be classified. Since the results may vary depending on the value of K, the methods have been tried with different values for K: 3, 6, 9, 12 and 15.

Table 3.3 presents the results obtained with both DYNOVO and the new method, as well as with some of the state-of-the-art methods. A brief description of the compared methods is presented below.

- Best single (OVO-BS) [Fri96]: every base classifier is used to classify each database using OVO decomposition. The results of the classifier with the highest accuracy are shown for each database.

- Static selection (OVO-ST) [Sze+09b]: for each sub-problem the base classifier that obtains the best results in a validation process is selected.

- Galar et al. (GALAR) [Gal+13]: the K nearest neighbours of the test instance are found and OVO is applied, only considering those classes in the neighbourhood. The value of K is established to 3 times the number of classes, and the result obtained with the best classifier is shown in each database.

- DYNOVO: the results obtained with DYNOVO and the K that gets the overall best accuracy are shown.

- PSEUDOVO: the results obtained with PSEUDOVO and the K that gets the overall best accuracy are shown.

The new approach (PSEUDOVO) obtains the best result among the used paradigms in 17 out of 26 datasets, being as well the classifier which shows the best mean, and the best ranking value. Regarding the differences, it is worth noticing that they are in general not very high, although some of them are remarkable. For instance, in the *wine* dataset, both PSEUDOVO and DYNOVO perform 2% worse than OVO and Galar; on the contrary, in *cmc*, *vehicle*, *winered* and *yeast* the new proposed approach obtains an accuracy 8% better than the previous approaches. In mean, PSEUDOVO outperforms all of the remaining paradigms, hence obtaining the best rank (1.90), followed by DYNOVO (2.19) and further from the other approaches. If the best result among all the paradigms is compared with each one individually, i.e., the maximum value is compared with the one obtained by a single approach, it is again PSEUDOVO which obtains the best results, obtaining a mean difference of -0.26%, while the other approaches have -0.41% (DYNOVO) -2.48% (Galar) -2.56% (OVO-BS) and -3.09% (OVO-ST); in this sense it can be stated that PSEUDOVO has proven to be the best in the performed experimental phase.

Iman–Davenport test has been used to detect statistical differences among the different algorithms. The results of the statistical analysis reject the null hypothesis that all the methods are equivalent, and the Shaffer post-hoc test has been run.

DB	OVO-BS	OVO-ST	GALAR	DYNOVO	PSEUDOVO
anneal	98.552	98.998	98.552	**99.443**	**99.443**
balance-scale	90.400	89.120	90.400	90.560	**90.720**
car	93.866	93.692	93.866	**96.470**	**96.470**
cmc	54.582	53.089	54.447	**63.069**	62.933
composers	85.589	85.677	86.292	**89.104**	**89.104**
dermatology	97.541	96.995	**98.361**	97.268	97.541
ecoli	86.607	85.417	86.905	89.583	**90.179**
glass	73.832	70.561	73.832	**75.701**	**75.701**
iris	**96.667**	**96.667**	**96.667**	95.333	95.333
imgsegment	97.186	97.186	97.013	**97.532**	**97.532**
lymph	**87.162**	86.486	**87.162**	**87.162**	**87.162**
nursery	97.238	97.824	97.130	**98.696**	**98.696**
optdigits	98.292	98.256	**98.523**	98.025	98.221
page-blocks	97.223	97.003	97.168	97.552	**97.734**
pendigits	98.026	98.426	98.299	98.472	**98.999**
satimg	88.283	88.454	88.361	90.070	**90.536**
solar-flare1	70.279	69.969	70.588	**73.684**	73.375
solar-flare2	75.516	75.141	75.516	**76.923**	76.829
soybean	94.583	93.997	93.704	**94.729**	94.290
vehicle	75.414	75.887	75.887	83.215	**83.924**
vowel	82.828	84.141	83.636	90.303	**90.707**
waveform-5000	86.700	86.680	**86.720**	85.000	85.300
wine	**98.876**	96.067	**98.876**	96.629	96.629
winered	61.789	58.974	60.788	69.543	**69.856**
yeast	59.434	58.895	59.771	67.318	**67.722**
zoo	96.040	95.050	96.040	**97.030**	**97.030**
Mean	86.250	85.717	86.327	88.401	**88.537**
Rank	3.46	4.15	3.29	2.19	**1.90**

Table 3.3: Classification accuracies obtained with different methods for each database. The best results are highlighted in bold.

The results show that PSEUDOVO is the most robust strategy since it outperforms significantly OVO-ST, OVO-BS and GALAR.

The method is explained in detail in the paper below:

- Problems Selection Under Dynamic selection of the best base classifier in One versus One: PSEUDOVO. **Submitted** to Applied Soft Computing journal

3.5 Conclusions

In this chapter some background on supervised classification has been presented, as well as a contribution to the field, in which a new version of DYNOVO called PSEUDOVO is presented. In this new method, in addition to the dynamic classifier selection that DYNOVO does for each sub-problem of the binarization step, a dynamic sub-problem selection proposed by Galar et al. [Gal+13] is also performed. The presented method has two main steps; in the first one the sub-problems that

presumably are competent in the classification of a new instance are selected, and in the second step the most competent classifier for each sub-problem is intended to be chosen.

The performance of the proposed method has been compared to some OVO versions, DYNOVO and the sub-problem selection technique proposed by Galar et al. [Gal+13]. Several databases from the UCI repository have been used to test the accuracies of the tested methods, as well as a new database of a global feature representation of musical pieces of some well known classical composers. PSEUDOVO has shown to be a good optimization of DYNOVO taking into account the competitive results that have been obtained.

As future work real applications in which PSEUDOVO could be applied are to be searched; on the other hand, there are some lines to investigate, such as the inclusion of new classifier algorithms which could adapt better to each considered two-class problem. Finally, the selection of the problems which would take part in the final decision can be tackled as a classification problem itself.

4 Music Classification

4.1 Introduction

Automatic music classification is a task within the field of Music Information Retrieval (MIR) which is getting more attention with the growth of the available information thanks to the digital media. Music Information Retrieval is a multidisciplinary field concerned with retrieving information from large music databases, in order to allow users to access and explore music in its different facets. The creation of huge collections coming from the restoration of existing archives and new content is constantly growing. These collections can consist of audio recordings, symbolic musical representation and other cultural information, and are in many cases unstructured data. In order to be able to retrieve data from these collections the instances in them need to have some category, or class, associated. These classes can make reference to their genre or author for example, and annotating them manually can be a heavy and difficult task.

Depending on the data that have to be organized, different types of classification tasks are identified in many works of topics like genre classification, tune family identification or composer recognition, and several different approaches have been developed to solve these problems, dealing with both symbolic and audio data.

Different taxonomies are used to classify music based on its genre, performer, composer or geographical or cultural point of origin, for example, but building such taxonomies is not always easy. Pachet and Cazarly [PC00] concluded by studying some of the genre taxonomies that are used in industry and on the Internet, that the hierarchical organization of genres is not an easy task, since they sometimes have fuzzy boundaries. However, even though it is not easy to define, genre is a crucial descriptor that is usually used to organize large collections of music, specially on the Internet, where it is often used in search queries.

Another popular task is the classification of folk tunes into tune families, which has long been a research topic in folk music studies and music data mining. Music, like languages, evolves through a process of transmission, variation and selection, and this evolution process results in contemporaneous groups of related melodies known as *tune families* [Bay50]. Generally documentary evidence of historic origin of tunes is unavailable, and tune families may be proposed based on melodic

similarity between tunes [VK12]. This is an important task, since folk music archives represent cultural heritage, and they need to be categorized in order to be consulted. Traditionally, analysis of tune families has been done by manually identifying and aligning groups of related melodies and comparing them, which is not feasible when dealing with large databases, and some computational methods are necessary.

In the case of the recognition of the composer of a musical piece a knowledge of music theory and historical context of the composer are needed when it is done manually. It is assumed that the compositions of an author share some stylistic elements, and these elements need to be identified in order to recognize who the author of a certain piece is. In order to try to do it automatically some features that will be used to characterize the pieces of each composer have to be defined, but since no theory exists for determining which features will be successful to characterize a style, different ones have been tried. Once a good representation of the pieces has been selected, a classification method is usually applied, which after being trained with already classified pieces, will be able to predict the composer of new unknown pieces.

An important element that should be taken into account when working on composer recognition is the influence that a composer could have had in the work of another one. Relations between some contemporary composers have been documented, and others might have existed that are not known. These relations could affect the classification of their pieces, since a composer could borrow some stylistic features from some works of other authors to use them in their own work, making the classification of these pieces more difficult.

In this chapter some contributions made to the field of automatic music classification are presented. Three classification tasks have been faced: genre classification, tune family identification and composer recognition. For the genre recognition an unsupervised method has been applied to a new matrix based representation of pieces to group similar bertso melodies into clusters. New melodies have been generated taking into account the melodic features of the pieces in each corpus, and the same matrix representation has been used to classify these new melodies into the clusters. A tune family classification method is also presented which has been applied on the Meertens Tune Collection, a digital collection of Dutch popular songs. Finally, the composer recognition task has been tackled with a work that proposes a matrix representation to classify pieces of some well known musical composers.

4.2 State of the art

Several approaches can be found in the literature that deal with music classification for different tasks, like tune family identification or automatic music genre classification. The idea behind many of them is to obtain a representation of the music and afterwards build a model which would be able to classify the characteristics of the music treated on the approach, namely genre, structure, artist, composer, and so forth. Different techniques are used depending on the type of data that is being used, symbolic or audio data, and several representations have been developed to obtain a good characterization of the pieces to classify them.

Some classification approaches distinguish two main phases: the representation of the pieces and their classification into one of the target classes. Several approaches can be used for representing melodies, such as the use of global features, n-gram model and its extension, multiple viewpoint models, or the discovery of distinctive sequential patterns. Some of the methods that can be found in the literature for different music classification task are specified below.

- **Genre classification**

Automatic music genre classification is a task that has attracted the interest of the music community for more than two decades, and several similarity methods and machine learning techniques have been studied in the literature to deal with it. Many of these works focus on finding a good representation of the data. Kotsifakos et al. [Kot+13] have dealt with music genre classification for symbolic music, and specifically MIDI, by combining the SMBGT similarity measure for sequences with the K-Nearest Neighbour (K-NN) classifier. For all MIDI songs they first extract all of their channels and then transform each channel into a sequence of 2D points, providing information for pitch and duration of their music notes. SMBGT similarity between all the pairs of the songs' channels is computed to then use this similarity score in the K-NN classification.

Valverde et al. [VR+14] present a novel feature vector obtained from a description of the musical structure described in MIDI files for music representation and use it to classify pieces in Classical, Brazilian Backcountry, Pop/Rock and Jazz genres. They compare the classification results obtained with the feature vectors to the results obtained with other features such as histograms of notes and statistical moments, obtaining promising results.

Some works apply genre recognition to folk music to distinguish between different sub-genres. Bassiou et al. [Bas+15] deal with Greek folk music genre classification. Hillewaere et al. [Hil+14] have worked on automatic classification of dances using

the *Dance-9* corpus studying the performance of several alignment methods. Chai and Vercoe [CV01] have worked on the classification of folk music pieces coming from different countries using monophonic melodies by means of hidden Markov models. In the cited paper, after trying several representations of the pieces, the authors state that "This shows that melodies of folk music do carry some statistical features to distinguish them".

Several works that deal with audio data have also been developed, like the one of Mandel and Ellis [ME05], which presents an approach based on support vector machines to classify songs based on global audio features. On the other hand, Bergstra et al. [Ber+06] present an algorithm based on Adaptive Boosting (ADABOOST) that is able to predict musical genre and artist from an audio waveform.

In the work of Lee et al. [Lee+15] the Bag of Words representation of modulation spectral analysis of spectral as well as cepstral features is constructed for music genre classification. This is an approach used as well in text classification [Zel+11b] which can be improved by means of a Singular Value Decomposition approach [Zel+15].

Recent success with deep neural network architectures on large-scale datasets has inspired numerous studies in the machine learning community for various pattern recognition and classification tasks such as automatic speech recognition, natural language processing, audio classification and computer vision [Mal16; Kar+16; Dor+16]. Work on music genre classification has been done as well; Rajann et al. [Raj+15] show that neural networks are comparable with classic learning models when the data is represented in a rich feature space. Zhang et al. [Zha+14] have proposed the use of computational deep learning modules for extracting invariant and discriminative audio representations which can then be used to classify music in different genres.

- **Tune family identification**

In computational studies of tune families different techniques like content-based retrieval [Kra+12] or predictive classification [Con13a] have been applied. Nearest neighbour methods have achieved good results in tune family identification compared to other tasks such as classification of tunes into folk music genres or geographical regions [Hil+14]. Applications for automatic tune family classification measure similarity between tunes, using methods like the measure of geometric distance on global-feature [Kra+13] or wavelet representations [Vel+13], edit distance on single-viewpoint string representations [Hil+14] or combined event features [Kra+13]. Compression distance on point-set [Mer14] or multiple-viewpoint representations [LM16] have also been used for tune family classification.

- **Composer recognition**

On the task of automatic composer classification, Herremans et al. [Her+15] propose a 12 global feature set to classify pieces of three classical composers with different classification methods, some of them descriptive, which are used to generate rules that are then applied in a generation method.

Dor and Reich [DR11] manually extract some pitch based features and use a classifier tool named CHECKUP to discover more features that they then use to classify pieces of nine composers. They also apply different classification methods like a K-Nearest Neighbour classifier, C4.5 decision tree classifier or support vector machines, and they compare their accuracies.

4.3 Contributions

The contributions of this work to the field of automatic music classification are related to three of the classification tasks mentioned in the introduction: genre classification, tune family identification and composer recognition.

Different approaches have been used to deal with each task, even though the works related to genre recognition and composer recognition both use a matrix based representation of the pieces of a corpus.

The description of each of the developed works and the publications related to them are presented below, as well as an improvement of the tune family identification method, which has not been published yet.

4.3.1 Unsupervised classification of bertso melodies

In this work an unsupervised genre classification of bertso melodies has been performed. To do so a new matrix based representation of pieces has been presented and several matrix distances have been tried to create clusters of similar bertso melodies. This corpus consists of melodies that have been used to sing bertsos, but that can belong to different genres. In this work we have assumed that similar melodies within that corpus have the same genre.

A subset of 100 melodies of the corpus described in Section 2.3.1 has been selected to be used in this study. The melodies in the corpus have a genre associated, but it has been confirmed with the administrator of the corpus that the descriptor is actually related to the lyrics that were used with the melodies when they were collected. Since there has not been any melody based genre analysis done, an unsupervised

classification method has been used to group them into clusters of similar melodies, which presumably have the same genre.

Once the clusters are created an early version of the music generation method of Chapter 2.3.3 is used to create 10 new melodies from each cluster. The hypothesis is that the new melodies that are generated from each cluster would belong to the same genre as the members of the cluster. To confirm that hypothesis the new melodies are then classified into the clusters. In Figure 4.1 a diagram of the method can be seen, where every step of the process is shown. Each of these steps is described below:

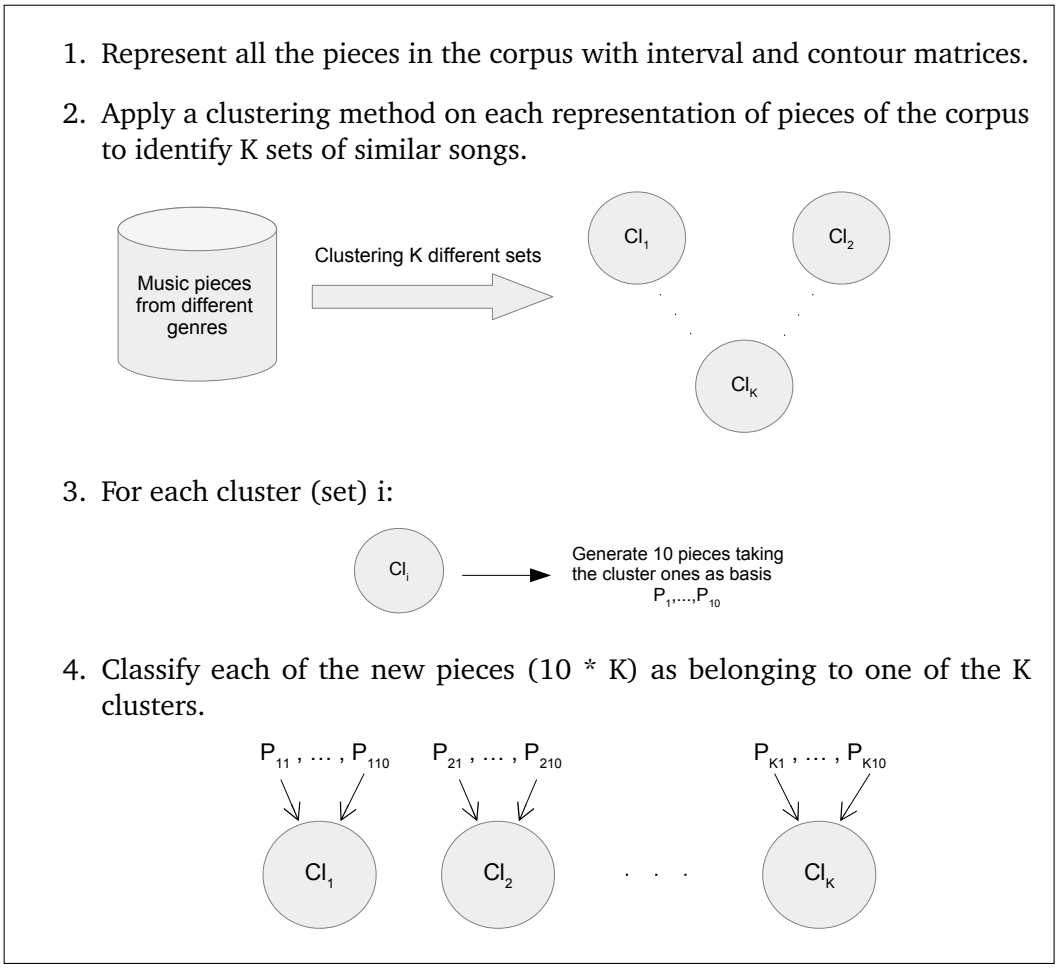


Figure 4.1: Diagram of the method presented in this work.

1. Representation of the pieces

In order to be able to group similar melodies they need to be somehow represented first. In this work a matrix representation has been presented, which is used to describe some melodic feature of the pieces.

Two melodic features have been proposed to characterize the melodies in the corpus: modulo 12 intervals of each pair of contiguous notes (intpc) and five point contour (5pc), where the melodic contour between two contiguous notes is represented with five possible values. In this viewpoint representation contours ld and lu record whether a note is approached by a leap of three or more semitones (down or up), sd and su represent a step (smaller than three semitones) approximation and eq represents a unison. In Figure 4.2 a short fragment of a score included in the corpus can be seen along with its intpc and 5pc representations.

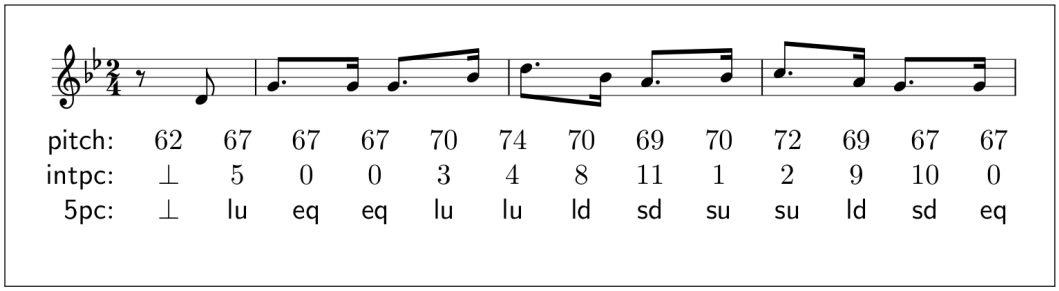


Figure 4.2: Viewpoint representation of the first two bars of the melody *Abiatu da bere bidean*.

Once the intpc and 5pc representations of all the pieces in the corpus have been obtained, the interval and contour matrices are built. Interval matrices are 12x12 matrices that describe the number of occurrences of each possible interval transition in each piece, while contour matrices are 5x5 matrices which count the number of transitions between all the contour pairs of each piece.

In Figure 4.3 an example of an interval matrix can be seen. What this matrix shows, for example, is that in the piece the matrix represents, a modulo 12 interval of 2 semitones followed by another 9 semitones happens 3 times (position (2,9)).

$interval$	0	1	2	3	4	5	6	7	8	9	10	11
0	9	0	0	4	0	1	0	0	0	0	2	0
1	0	0	5	0	0	0	0	0	0	0	0	0
2	2	0	1	1	0	0	0	0	0	3	0	0
3	0	0	1	0	2	0	0	0	0	0	0	3
4	0	0	0	0	0	0	0	0	2	0	0	0
5	2	0	0	0	0	0	0	0	0	0	1	0
6	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	1	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	2
9	0	1	0	0	0	0	0	0	0	0	3	0
10	3	0	0	1	0	0	0	1	0	1	1	3
11	0	4	0	0	0	0	0	0	0	0	4	0

Figure 4.3: Example of an interval matrix extracted from a piece in the corpus.

2. Apply a clustering method on each representation of the pieces of the corpus

Once the matrices are built, in order to discover similarity relations among the pieces, an agglomerative hierarchical clustering algorithm has been used in each matrix representation (Sequential Agglomerative Hierarchical Non-overlapping algorithm (SAHN)) [JD88]. SAHN needs a distance to measure how similar two matrices are, and several distances that can work with matrices have been tested:

- The distances induced by the following norms: 1-norm, ∞-norm, Frobenius norm, maximum modulus norm.
- Kullback-Leibler and Jeffrey divergences.
- Earth mover's, Manhattan and Intersect distances.

3. Generate 10 songs from each cluster

After applying the SAHN algorithm with the previous matrices distances to the pieces in the corpus, several clusters partitions are created, which are used to generate new melodies that are intended to have the same genre as the original pieces.

An early version of the generation method presented in Chapter 2.3.3 has been used to generate 10 new melodies from each one of the clusters. The generation method takes an existing piece where repeated and transposed segments are manually discovered, defining a simple coherence structure of the piece. The same coherence structure is used as template to sample new notes in it. To make sure that the sampled notes are "in the style" of the corpus, a statistical model is used to guide the generation of high probability pieces.

In this case, four pieces that are not part of the corpus of this work have been randomly chosen as templates, and 10 new melodies have been generated from each cluster and template piece. The statistical models needed to guide the generation of new pieces has been built from the pieces of each cluster. Since two matrix representations have been created for each piece in the clusters, two types of statistical models have been built; an interval model and a contour model. A whole generation process has been performed for each representation.

The sampling of new notes on the melodic coherence structure is done following the same stochastic hill climbing process used in the method of Chapter 2.3.3. 10 new melodies have been generated from each cluster and matrix type using each of the template melodies.

4. Classify each of the new pieces in one of the K clusters

As mentioned in the previous steps, four template pieces have been randomly chosen to generate 10 new pieces from each one of the clusters created using the different representations and distances. To test whether they have the same melodic features as the pieces in the clusters used to generate them, they also have been represented as matrices and classified within their closest cluster. To identify which of the clusters is the closest one, the same distance applied in the clustering phase is used. In Tables 4.1 and 4.2 the best accuracies obtained for each of the four templates and cluster number with contour and interval matrices are shown.

Contour matrices

Cluster Num	$Template_1$	$Template_2$	$Template_3$	$Template_4$
2	**0.875**	0.750	0.650	0.750
3	0.750	0.500	0.533	**0.800**
4	**0.500**	0.475	0.450	**0.500**
5	**0.550**	0.380	0.440	0.500
6	**0.583**	0.333	0.433	0.500

Table 4.1: Best results obtained for each template and cluster number with contour matrices.

Interval matrices

Cluster Num	$Template_1$	$Template_2$	$Template_3$	$Template_4$
2	0.875	0.750	0.800	**0.900**
3	0.667	0.667	**0.733**	0.600
4	0.625	0.475	0.550	**0.725**
5	0.400	0.360	0.440	**0.500**
6	**0.542**	0,350	0.350	0.483

Table 4.2: Best results obtained for each template and cluster number with interval matrices.

Heterogeneous classification results have been obtained with different distances and cluster numbers. In general interval representation is slightly better, although the best distance mean is obtained by M-norm with the contour representation. In general, different distances obtain the best result for different cluster numbers, which indicates that the appropriate one should be carefully selected for each considered case.

Obtained results indicate the appropriateness of the whole process: results over 0.5 can be considered encouraging, especially when the cluster number is 4 or more.

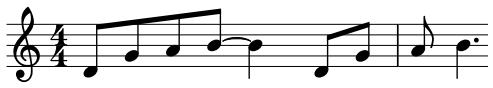
The work is presented in detail in the following publication:

- [Goi+18b] Izaro Goienetxea et al. “Towards the use of similarity distances to music genre classification: A comparative study”. In: *PLOS ONE* 13.2 (2018), pp. 1–18

4.3.2 Tune family classification with pattern covering

In this work a method for tune family classification is proposed, which is based on pattern sets and nearest neighbour classification. The method tries to find the most similar piece to each melody in a corpus, computing the similarity of two pieces with their shared statistically interesting patterns. The used corpus is a subset of the Meertens Folk Tune Collection: The Annotated Corpus[1]. It is a collection of 360 Dutch folk song melodies classified into 26 tune families, where the families contain between 8 and 27 melodies, each of them with a length of around 150 notes.

The first step to discover the patterns between the pieces of the corpus is to represent them using some feature, or combination of features, of the events in the pieces. In this approach a multiple viewpoint representation is used [Con13b]. Figure 4.4 presents a short tune fragment with different viewpoints used in this work, including interval from the previous note int or key note intref, 3-point contour c3 of pitch, duration or inverse metric level, 5-point contour of pitch c5(pitch, 3), which records whether the note was approached by a leap (three semitones or larger), a step (less than three semitones) or unison, and inter-onset-interval ioi. Another viewpoint used in this work is phrpos which records whether the note is first, last or inside a phrase [KC16].



pitch	62	67	69	71	62	67	69	71
dur	840	840	840	2520	840	840	840	2520
int	$\perp$	5	2	2	-9	5	2	2
ioi	$\perp$	840	840	840	2520	840	840	840
intref	P5	P1	M2	M3	P5	P1	M2	M3
c3(pitch)	$\perp$	$+$	$+$	$+$	$-$	$+$	$+$	$+$
c5(pitch, 3)	$\perp$	$++$	$+$	$+$	$--$	$++$	$+$	$+$
c3(dur)	$\perp$	$=$	$=$	$+$	$-$	$=$	$=$	$+$
c3i(level)	$\perp$	$+$	$-$	$+$	$-$	$+$	$-$	$+$
int $\otimes$ intref	$\perp$	5, P1	2, M2	2, M3	-9, P5	5, P1	2, M2	2, M3
intref $\otimes$ c3(pitch)	$\perp$	P1, $+$	M2, $+$	M3, $+$	P5, $-$	P1, $+$	M2, $+$	M3, $+$

Figure 4.4: Small fragment from a Dutch tune. Each line shows a different viewpoint and its transformation of the event sequence. Top: basic viewpoints, middle: derived viewpoints, bottom: linked viewpoints.

[1] http://www.liederenbank.nl/

Once the pieces in the corpus have been represented with different viewpoints, the patterns that occur between them are discovered with a sequential pattern discovery algorithm [Ayr+02]. A *pattern* is a sequence of event features described using viewpoints, and a piece instantiates a pattern if the pattern occurs one or more times in the piece. The number of pieces instantiating a pattern gives the *piece count* of the pattern. In Figure 4.5 an example of pattern occurring in the intref representation of two pieces can be seen -in red-.

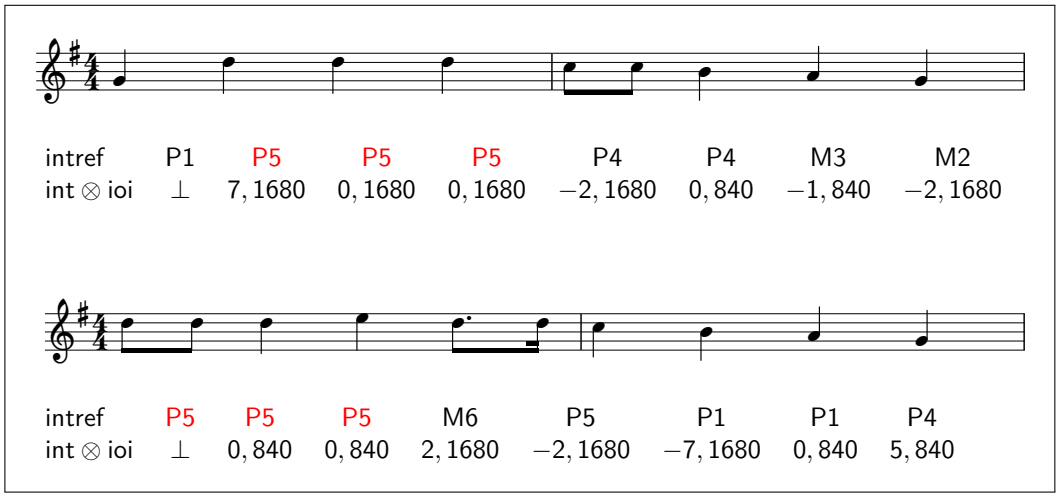


Figure 4.5: Example (in red) of a pattern found in the intref representation of two pieces of the corpus.

In the classification step, a leave-one-out strategy is followed, and each piece of the corpus is treated as a query, while all the remaining pieces are considered targets. The goal is to find the most similar target piece for each query. To do so all the patterns that happen between each query-target pair are sorted from most to least *interesting*: a pattern is considered the more interesting the more surprising its occurrence is in both the query and the current target tune. To compute the interest value of a pattern shared by the query and a target piece the process below is followed.

Using a binomial distribution the probability of a pattern P of length c occurring one or more times in a tune of length ℓ is $\mathbb{B}_{\geq}(1; \ell - c + 1; p)$, where p is the background probability of pattern P calculated from a zero-order model of the training corpus (excluding the statistics of the query). Then the expected piece count of the pattern in a query tune of length ℓ_q and a target tune of length ℓ_t is

$$\lambda = \mathbb{B}_{\geq}(1; \ell_q - c + 1; p) + \mathbb{B}_{\geq}(1; \ell_t - c + 1; p)$$

and the interest $\mathbb{I}$ of pattern P is the deviation between the expected piece count of the pattern (λ) and its actual piece count (always 2 in this case), computed using the negative logarithm of the Poisson approximation of the binomial distribution

$$\mathbb{I}(P) = \lambda + \ln(2) - 2\ln(\lambda)$$

with a higher value of $\mathbb{I}(P)$ indicating a more surprising pattern.

To compare two tunes, a covering method is applied to identify *sets of patterns* shared by both tunes. Candidate patterns are sorted by their interest $\mathbb{I}$, and the sorted pattern list is processed iteratively to choose the patterns that in each iteration fit best into the positions of the pieces that have not been yet covered by any pattern, not allowing overlapping between contiguous patterns. The similarity score for a pair of tunes is the summed interest of all patterns used in the covering of the two tunes.

In Figure 4.6 a diagram can be seen where the covering examples of a query and two different target pieces are shown. Patterns are represented with shapes, where patterns found in different viewpoint representations of the pieces have different shapes. The interest value and label are specified within each pattern in the figure. In this example the query-target$_2$ pair has a similarity score of 100.8, while the query-target$_1$ pair has a similarity score of 95.7, in consequence, the class of target$_2$ (Maadge) would be assigned to the query piece.

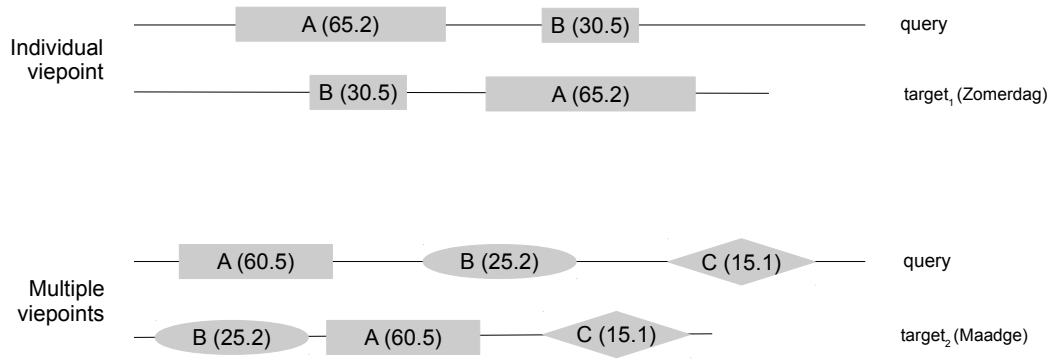


Figure 4.6: Diagram of covering examples of a query and two different target pieces. Different patterns are represented in each covering with shapes, where patterns found in different viewpoint representations of the pieces have different shape.

For the covering of the query-target pairs, we distinguish two situations:

(1) Individual viewpoints: patterns are discovered using one viewpoint representation at a time, and all patterns in the covering pattern set have the same individual viewpoint, where the viewpoint can be primitive or linked (e.g. int or int $\otimes$ ioi). An example of this type of covering can be seen in the top part of Fig 4.6, where a

covering of the query and target_1 pair can be seen, and all the used patterns, A and B, have the same type, and in consequence, the same shape.

(2) Multiple viewpoints with merging: patterns are discovered for each of the chosen viewpoints, and the set of patterns used in the covering of the tunes can include patterns of different viewpoints (e.g. both intref and c3(pitch) patterns). An example is shown in the bottom part of Fig 4.6, where three different pattern types are used in the covering.

In both cases, the covering results in one pattern set for each pair of tunes, from which the similarity score is computed. The unclassified tune is assigned the class of the most similar labelled tune.

The classification results obtained with this pattern covering and merging strategy are presented in Table 4.3, which shows the three best results obtained using single viewpoint (top), the best results obtained with multiple viewpoints (middle) and the results obtained with Fully Saturated Viewpoints (bottom). Fully Saturated and Linked Viewpoints contain all possible dyadic linked viewpoints formed from the following viewpoints: $\text{intref}, \text{c3}(\text{dur}), \text{c3}(\text{pitch}), \text{c5}(\text{pitch}, 3), \text{c3i}(\text{level}), \text{int}, \text{ioi}, \text{phrpos}$. Fully Saturated Viewpoints contain the single viewpoints in addition to all the linked viewpoints.

Viewpoints	Classification	Accuracy
intref	336/360	93.3%
intref $\otimes$ c3i(level) $\otimes$ phrpos	333/347*	96.0%
intref $\otimes$ c3i(level)	320/347*	92.2%
intref $\otimes$ c3i(level) $\otimes$ phrpos & intref $\otimes$ phrpos	**344/360**	**95.6%**
intref & int $\otimes$ ioi	**344/360**	**95.6%**
intref $\otimes$ c3i(level) $\otimes$ phrpos & int $\otimes$ ioi	340/360	94.4%
intref $\otimes$ c3i(level) & int $\otimes$ ioi	334/360	92.8%
Fully Saturated and Linked Viewpoints (28)	332/360	92.2%
Fully Saturated Viewpoints (36)	315/360	87.5%

Table 4.3: Classification accuracy with different viewpoints. (*)Classification on 347 pieces done where the viewpoint c3i(level) is used and not merged with any other viewpoint, since it is undefined for 13 pieces of the corpus.

The top results for linked and multiple viewpoints suggest that for this particular task and dataset metric and phrase information are important in addition to pitch or interval information. Similarly, van Kranenburg et al. [Kra+13] reported their highest classification accuracy for a combined edit distance on pitchband, metric weight and phrase position. Alternatively, combining intref and int $\otimes$ ioi also correctly classifies 344 out of 360 (95.6%) tunes.

All of the viewpoint selections listed in Table 4.3 achieve higher classification accuracies than earlier studies on the same corpus which used pitch-time representations and nearest neighbour classification (83.9% and 85.6%) [Mer14; Vel+13]. The best results are above the 94.4% accuracy for interval-based edit distance [Hil+14] and multiple viewpoint representation with corpus compression distance [LM16], but slightly lower than the 96.7% accuracy with multiple-viewpoint probabilistic classification [Con13a]. The classification accuracy of 98.9% reported by van Kranenburg et al. [Kra+13], using multiple-feature alignment and nearest neighbour classification, has not yet been achieved by any other method.

The results obtained in this work (average accuracies in leave-one-out) indicate that the presented pattern discovery, ranking and covering process is effective for tune family classification.

The method is described in the paper below:

- [Goi+16] Izaro Goienetxea et al. "Melody classification with pattern covering". In: *9th International Workshop on Music and Machine Learning (MML 2016)*. Riva del Garda, Italy, 2016

Improvement of the method

The presented method has been improved with a modification of the covering method that is not included in the paper above. In the first version of the method there was no constraint on the order of the patterns used in the covering. In the diagram of Figure 4.6 it can be seen that the order of the patterns used in the coverings of the queries and target pieces is not the same. In the query-target$_2$ pair the order of the used patterns is ABC in the query, while for target$_2$ the order is BAC. In the new covering method the collinearity of the patterns between query and target pieces is constrained, to make sure that not only both pieces share interesting patterns, but also the sequential relations between them. In Figure 4.7 an example of the allowed type of relation between patterns can be seen.

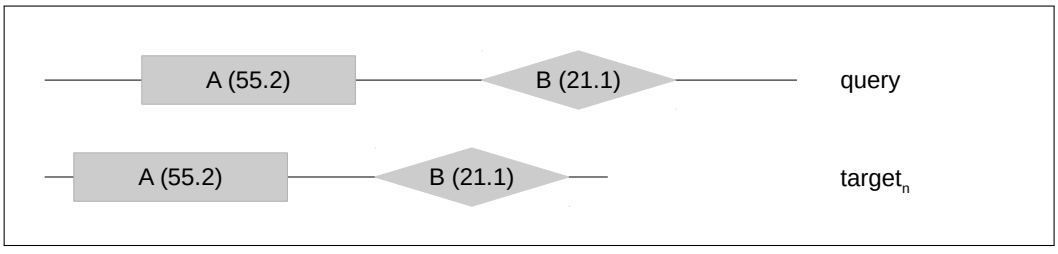


Figure 4.7: Diagram of a covering example of a query and a target pieces conserving collinearity of patterns.

The classification accuracies using different viewpoint representations and covering with and without collinearity are shown on Table 4.4.

Viewpoints	Classification Accuracy		Previous Accuracy
intref	339/360	94.2%	93.3%
intref ⊗ c3i(level) ⊗ phrpos	334/347	96.2%	96.0%
intref ⊗ c3i(level)	327/347	94.2%	92.2%
intref ⊗ c3i(level) ⊗ phrpos & intref ⊗ phrpos	**350/360**	**97.2%**	**95.6%**
intref & int ⊗ ioi	343/360	95.3%	95.6%
intref ⊗ c3i(level) ⊗ phrpos & int ⊗ ioi	344/360	95.5%	94.4%
intref ⊗ c3i(level)&int ⊗ ioi	334/360	92.8%	92.8%

Table 4.4: Classification accuracy with different viewpoints forcing collinearity of patterns on the covering.

It can be seen that the results obtained when forcing collinearity of the patterns in the covering are overall better than the results of the older version of the method. Even though for some representations better results are obtained with no collinearity, the highest accuracy is achieved when it is constrained.

4.3.3 Composer recognition with matrix representation

In this contribution the automatic composer recognition problem has been studied, using two corpora of pieces of well known composers. A matrix based representation has been used to represent the pieces of the corpora and several classifiers have been tried in order to get a high classification accuracy. Finally, the obtained accuracies have been compared to the accuracies obtained by a global feature representation used in a similar task and presented in [Her+15].

The first used corpus (corpus_3) is formed of polyphonic pieces of Bach, Beethoven and Haydn, and has a total of 1138 pieces. It is similar to the corpus used in [Her+15] and the distribution of the composers and pieces can be seen in the central part of Table 4.5. The second corpus (corpus_5) is an extension of the first one, but it also includes pieces of Mozart and Vivaldi. It contains 1586 pieces, and its composer/piece number distribution can be seen in Table 4.5.

All the pieces of both corpora have been downloaded in MIDI format from the KernScores website [Sap05], which was developed by the Center for Computer Assisted Research in the Humanities (CCARH), at Stanford University.

Two main steps have been defined in the composer identification process: representation and classification. The goal of the representation step is to find a way to characterize the pieces well enough to be used in the classification process, and in

Composer	Instances
Bach	694
Beethoven	190
Haydn	254
Mozart	313
Vivaldi	135

Table 4.5: Number of pieces of each composer used in this work.

this approach a matrix based representation similar to the interval matrix used in the unsupervised classification of bertso melodies of [Goi+18b] has been selected. In the cited work, 12x12 matrices that describe the number of transitions between all the modulo 12 interval pairs that occur in each piece are used, while the matrices used in this composer recognition task describe the probability of all the modulo 12 interval pairs of each piece .

The first step to build them is to compute the modulo 12 intervals of all the contiguous notes in each voice of each piece in the corpus. Since the pieces of the corpora are polyphonic, the intervals need to be computed for all the voices in each piece. Once the intervals are computed the 12x12 matrices are built, one for each piece, where the probabilities of transitions between intervals in all the voices are presented.

In order to test the validity of this representation, it is compared to a global feature representation used by Herremans et al. [Her+15] in a similar task. The list of the used global features is shown in Table 4.6.

Variable	Feature Description
x_1	Prevalence of most common pitch
x_2	Prevalence of most common pitch class
x_3	Relative prevalence of top pitches
x_4	Relative prevalence of top pitch classes
x_5	Prevalence of most common melodic interval
x_6	Relative prevalence of most common melodic intervals
x_7	Repeated notes
x_8	Chromatic motion
x_9	Stepwise motion
x_{10}	Melodic thirds
x_{11}	Melodic perfect fifths
x_{12}	Melodic octaves

Table 4.6: Global feature collection used in the global_{12} representation.

The classification step consists in assigning a class (in this case a composer) to a set of features that correspond to a piece. Since in the data set used in this work the composer of each piece is already known, a supervised learning technique is used to train the classifiers that then will be able to classify new unlabelled pieces.

Different base classifiers from the machine learning software Weka [Hal+09] have been tried in the classification steps: J48, SMO, JRip, Naive Bayes (NB), Bayesian Network (BNet), Random Forest (RF) and Multilayer Perceptron (MP). Since there is no test set available to test the performance of the classifiers, in each classification process a stratified 10 fold cross-validation has been used.

The classification task faced in this work is a multi-class problem, so class-binarization techniques can be applied in order to try to facilitate the classification. In this approach OVO binarization technique has been chosen to analyse the effect of class binarization in the classification. To combine the outputs of the sub-problems created on the decomposition step, the majority vote strategy [FÖ2b] is used, where each sub-problem returns a vote, and the class with the largest amount of votes is predicted.

The classification accuracies obtained for both corpora can be seen in Table 4.7. The accuracies for the matrix and global feature representations and all the base classifiers, with and without OVO are shown.

Corpus$_3$

	J48	J48-OVO	SMO*	JRIP	JRIP-OVO	NB	NB-OVO
global_{12}	84.007	83.568	86.028	82.074	83.655	66.784	66.872
$\text{matrix}_{\text{intpc}}$	81.459	82.1617	**89.982**	81.986	82.162	80.668	80.580

	BNet	BNet-OVO	RF	RF-OVO	MP	MP-OVO	Mean	Mean-OVO
global_{12}	78.647	78.735	87.171	87.786	87.346	87.434	***81.722***	***82.011***
$\text{matrix}_{\text{intpc}}$	82.513	81.986	86.907	88.401	**89.982**	89.631	***84.559***	***84.986***

Corpus$_5$

	J48	J48-OVO	SMO*	JRIP	JRIP-OVO	NB	NB-OVO
global_{12}	71.402	75.126	73.864	70.328	72.980	56.692	56.692
$\text{matrix}_{\text{intpc}}$	70.266	72.917	80.556	72.033	72.854	71.970	71.843

	BNet	BNet-OVO	RF	RF-OVO	MP	MP-OVO	Mean	Mean-OVO
global_{12}	65.530	67.361	77.904	79.104	74.432	76.641	***70.022***	***71.681***
$\text{matrix}_{\text{intpc}}$	72.096	73.106	79.419	80.556	**80.682**	80.177	***75.289***	***76.001***

Table 4.7: Accuracy results of the classifications with each single classifier and OVO technique for both corpora. The best obtained results are shown in bold. (*) SMO appears only once in the table since the function included in Weka has OVO already applied.

In both corpora the best results are obtained with the matrix representation and SMO or Multilayer Perceptron classifiers. Even though global feature representation improves the results obtained with matrices for some classifiers such as J48. The mean accuracies also show that overall, matrix representation obtains better results that the global feature representation, and that applying OVO is beneficial.

Some additional information can be extracted from the confusion matrix of the classification that obtains the highest accuracy (Multilayer Perceptron and matrix representation), which is shown in Table 4.8. It can be seen that pieces of Vivaldi and Bach are overall well classified, while most of the classification errors happen between Beethoven, Mozart and Haydn. Some of these classification errors could be caused by the relations that have been documented between these three composers, and the influence that they could have had on each other.

a	*b*	c	d	e	Classified as
667	8	3	8	8	a=Bach
11	119	23	30	7	b=Beethoven
9	30	209	61	3	c=Mozart
13	23	40	172	6	d=Haydn
15	2	3	3	111	e=Vivaldi

Table 4.8: Confusion matrix of the classification of the matrix representation of corpus_5 using Multilayer Perceptron.

In order to prove that there are statistical differences between global feature and matrix representations, a Wilcoxon signed-rank test has been applied. The result of the statistical analysis rejects the null hypothesis that both methods are equivalent.

Further statistical analysis has also been carried out in order to detect statistical differences between OVO and single classifier and, we have again applied Wilcoxon signed-rank test. The obtained results rejects the null hypothesis.

The classification accuracies and the results obtained from the statistical tests indicate that both the use of a matrix representation and OVO class binarization are beneficial for this kind of classification tasks.

The work is described in the paper below:

- [Goi+18a] Izaro Goienetxea et al. “On the Use of Matrix Based Representation to Deal with Automatic Composer Recognition”. In: *AI 2018: Advances in Artificial Intelligence - 31st Australasian Joint Conference, Wellington, New Zealand, December 11-14, 2018, Proceedings*. 2018, pp. 531–536

4.4 Conclusions

In this chapter some tasks within the field of automatic music classification have been described. Some related works of the literature have been presented, as well as our contributions to the field. Three main tasks have been tackled: genre classification, tune family recognition and composer identification. All the contributions of the

chapter to the field have been focused on the use of symbolic data, and pieces in MIDI format have been used in all the classification tasks.

The corpus used in the genre classification approach presented in this work is a 100 piece subset of the bertso melody corpus presented in Chapter 2.3.1. The entries in the catalogue have a genre descriptor, but it had been assigned according to the lyrics that were used with the melody when they were collected, making them unsuitable for a supervised classification method. An unsupervised classification method which uses a matrix representation of the pieces of the corpus has been presented and several distances to group them into clusters have been tested. Then, a music generation method has been used to generate 10 new melodies based on each of the clusters, which then have been classified again into the created clusters in order to evaluate the method used to create the clusters. The obtained results have shown that the process is appropriate for this kind of classification tasks.

An approach to tune family recognition has also been presented which is based on a pattern discovery and covering process. The Annotated Corpus subset of the Meertens Folk Collection has been used, which has 360 tunes classified into 26 tune families. Many viewpoint representations of the pieces in the corpus have been used to discover patterns occurring between the representations of different pieces. Then a covering method has been applied to cover pairs of pieces in a dense way, trying to find for each piece in the collection the piece with which it gets the best covering. An improvement of the method has also been presented which forces collinearity between patterns in the covering, improving the classification accuracies.

The last classification task that has been tackled in this chapter is the automatic composer recognition for which two corpora of polyphonic pieces of some well known classical composers have been compiled. A matrix representation similar to the one used in the unsupervised genre classification task has been tried to characterize the pieces in the corpora. To measure the suitability of this representation, the classification accuracies that are obtained with it are compared to the accuracies of a global feature representation from the literature which obtained good results in a similar task. Several base classifiers and a class binarization method have been applied in the classification of the pieces in both matrix and global feature representation, and promising results have been obtained, which indicate that the matrix representation of interval probabilities is suitable for this kind of task.

Since different classification tasks have been faced, several directions have been identified as future work. Matrix representations of interval or contour have been used for unsupervised genre classification and composer recognition in the presented works, and more complex matrices should be used in order to try to characterize the pieces better. This could be done including rhythmic information or combining

different melodic features, for example, by using linked viewpoints that would combine different viewpoints. Combining matrices with other type of representations like global features is also planned, in order to use information that comes from both representations and that have obtained acceptable classification accuracies when used independently. Some optimization of the One Versus One binarization used in the composer recognition task can also be applied to improve the classification accuracies.

In genre classification tasks, since the genre taxonomies are not easy to define, some cases might exist where instead of a single class it has several genres associated. A multi-label classification strategy [Zel+11a] should be tried in these cases.

The pattern discovery method used in the tune family identification task discovers patterns in a single viewpoint representation. This should be improved to include feature set patterns that would provide more information and would discover more complex patterns that would presumably improve the obtained results. Some work on the greedy covering strategy could also be done in order to make it in a *smarter* way.

5 Conclusions and future work

This dissertation presents the work done on three different fields: automatic music generation, supervised classification and automatic classification of music. These are three topics that are apparently independent but that have been related through this PhD.

Automatic music generation is a topic that interests many researchers and big companies like Google and Spotify, which are doing research on automatic music generation and have big projects like Magenta[1], and even Warner, who has bought an AI based music app which generates personalized sound to focus and relax. Other smaller companies like Melodrive[2] successfully apply music generation methods for creating music for video games among other tasks.

Our goal in this field was to be able to generate automatic and coherent melodies; melodies that sound *natural*. A good corpus of bertso melodies was offered to us by the Xenpelar documentation centre, which is specialized in bertsolaritza. We decided to use it to generate new pieces in the same "style", even though actually pieces of many different styles can be found in the corpus, since it includes Basque popular melodies and any melody that have been used to sing bertsos in competitions or exhibitions. In order to create pieces that share some melodic features with the melodies in the corpus, statistical models have been used in all the automatic generation approaches.

In order to endow the generated music with coherence, first the actual concept of coherence needed to be defined. Following the idea that music is interpreted like language by our brains, we concluded that in order to generate understandable music we would need to make sure of adding related segments through the new pieces. To generate music with related information, it was decided to use an existing piece as template, which is analysed in different ways to use its coherence structures to sample notes within them.

Supervised classification is a very popular machine learning approach widely used in real life problems because of the advantages it offers. In many cases, when dealing with multi-class classification class-binarization strategies are applied in order to be able to use classifiers initially designed for binary classification. The

[1] https://magenta.tensorflow.org/
[2] https://melodrive.com/

supervised classification chapter focuses on class binarization, which is widely used in real life to deal with multi-class classification problems. An optimization to the binarization method DYNOVO [Men+15] has been proposed, which has obtained good results compared to the results obtained with other state-of-the-art binarization methods. The proposed method has also been applied to a corpus of pieces of classical composers, proving that it is beneficial to classify music.

Automatic music classification is a field getting more and more attention with the growth of the amount of available data specially on the Internet. Many different techniques are applied nowadays in order to classify all the multimedia data in the Internet and the digitalised analogue data.

In all the works presented in this PhD dissertation symbolic data has been used, and some techniques from the music generation chapter are also applied in the tasks tackled within the music classification field. The pattern discovery and covering methods of the music generation method are applied to identify tune families of a corpus of Dutch folk tunes with good results. The statistical models built to characterize the corpora are used to characterize single pieces instead of a corpus, and used in the unsupervised classification of bertso melodies and in the composer recognition of polyphonic classical pieces. An OVO binarization technique is also applied in this last classification task. Good results have been obtained in all the music classification tasks that have been tackled.

As in this research project work in different fields has been done, many ideas have been identified as future work. An optimization of the pattern discovery method should be developed to include information of various viewpoints within the patterns. This optimization could improve both the coherence structures used in automatic music generation and the pattern based classification. The generation method should also be extended to generate polyphonic music, and the development of a method to automatically create coherence structures without using template pieces should also be considered. This would allow to completely automatize the generation process. Works with deep learning method are envisaged both for music generation and classification, and the matrices used in different classification tasks should be extended to include more features that would allow a better characterization of musical pieces.

Part I

Articles Related to Automatic Music Generation

6

Transformation of a bertso melody with coherence

Authors: Izaro Goienetxea and Darrell Conklin.

Booktitle: Proceedings of the 5th International Workshop on Folk Music Analysis (FMA 2015)

Year: 2015

TRANSFORMATION OF A BERTSO MELODY WITH COHERENCE

Izaro Goienetxea[1] **Darrell Conklin**[1,2]
[1] Department of Computer Science and Artificial Intelligence
University of the Basque Country UPV/EHU, San Sebastián, Spain
[2] IKERBASQUE, Basque Foundation for Science, Bilbao, Spain
{izaro.goienetxea, darrell.conklin} @ehu.eus

1. INTRODUCTION

The topic of automatic generation of music has existed for a long time, and many different approaches have been developed. One of these approaches is the use of statistical models. Statistical models of symbolic representations of music have been used in many works in the computational modeling of different music styles, like folk music. These models are able to capture some musical features, making it possible to generate new musical sequences that reflect an explicit musical style (Conklin, 2003; Dubnov et al., 2003).

Statistical models can also be used to transform pieces, by extracting the structure of a piece, describing it using semiotic labels, and sampling new pieces conserving that structure. This way a musical cohesion can be given to new productions (Conklin, 2003).

Bertsolaritza or bertsolarism is the art of singing improvised songs in Basque (bertsos), respecting various melodic and rhyming patterns. There is evidence of bertso singing and written bertso poem samples since the 15th century, and it is a very popular art nowadays in the Basque Country. Bertsos are sung in many different occasions, like informal lunches with friends, homage ceremonies or competitions and any topic can occur in a bertso. Many bertsolari competitions take place every year in the Basque Country, and every four years the national championship final is held, with around 15000 people in attendance.

In the work described herein a bertso melody is transformed to create new melodies, maintaining its original coherence structure.

2. METHODS

2.1 Bertso

Bertsolaritza is defined as a sung, rhymed and metered discourse by the book The art of bertsolaritza: improvised Basque verse singing, written by Garzia et al. (2001).

Bertso Doinutegia [1] is a collection of 3059 bertso melodies, created by Joanito Oiartzabal and published for the first time on 1995. It is updated every year by Xenpelar Dokumentazio Zentroa [2] with new melodies that are used on competitions or bertso exhibitions. Entries in the collection have a melody name, the name or type of the strophe, type of the melody (genre), creator, bertsolari who has used it, name and location of the person who has collected the melody and year of the collection. Melodies are classified into 17 different types or genres, and this classification is done based only on the melody. Some of the melodies in the collection have links to recordings of exhibitions or competitions where those melodies were used.

[1] http://http://bdb.bertsozale.eus/en/web/doinutegia/bilaketa
[2] http://bdb.bertsozale.eus/en/info/7-xenpelar-dokumentazio-zentroa

2.2 Semiotic Structure

Music structure description is currently a scientific challenge, and it can be approached in several ways. In this work semiotic structure is used to describe the structure of a bertso melody. Semiotic structure represents the similarities and internal relations of structural segments, using labels to identify similar segments (Bimbot et al., 2012). In the work described herein, the semiotic structure of a bertso melody is described by assigning a different semiotic label to each different note. For transformation, this label structure is taken as a constraint, specifying which notes must have the same pitch value.

Taking the whole structure as a constraint can be too restrictive when generating new melodies, since it does not allow a wide variety of different transformation. A melodic reduction strategy has been applied, to put the labels only on the more significant notes, allowing any pitch on the less important positions. These positions are labeled with a label X, where X matches any note. This method identifies the notes where the melodic direction changes, similar to what the contour reduction viewpoint of Conklin & Anagnostopoulou (2006), does but selecting the note previous to the position where the new contour is found, and puts label X on the other notes.

On the piece that is transformed in this work 42 Xs have been set out of 100 notes.

2.3 Sampling

Since in bertsos meter is a very important feature, in this work the rhythmic structure of the original piece is conserved, and a new melody line is created. A statistical model has been built, computing the probabilities of transitions between pitch contour values of 15 bertso melodies of the corpus described in Section 2.1. To do so, a five point contour (leap down, step down, repetition, step up, leap up) viewpoint has been computed, as in Conklin & Bergeron (2008), steps involving a motion of one or two

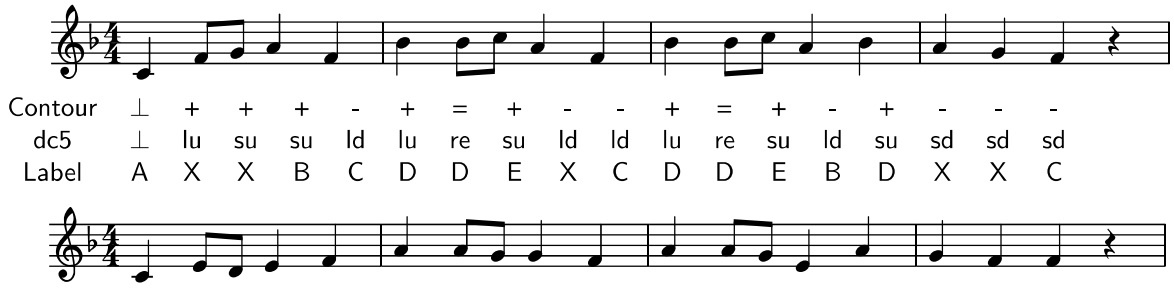


Figure 1: Segment of a transformation (bottom) of a bertso melody (top) with melodic contour, five point contour (dc5) and label of each note, determined by contour reduction.

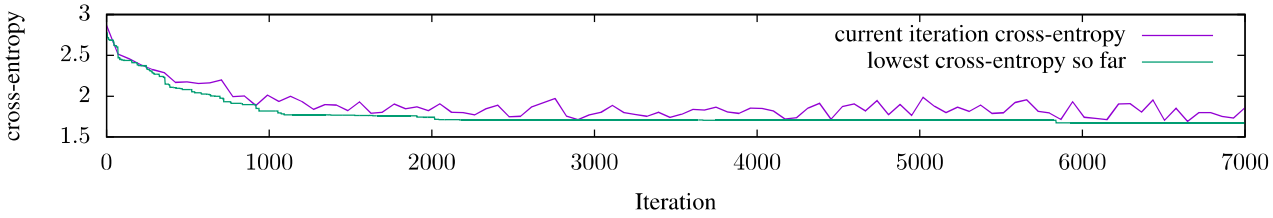


Figure 2: Cross-entropy of the piece through the iterations and the lowest cross-entropy reached.

semitones, and leaps a motion greater than two semitones.

A *stochastic hill climbing* method has been used for sampling, and this process has been iterated 10^4 times. A random piece with the same semiotic structure as the original song is taken as a starting point, and in each iteration a random location i in the piece is chosen. A random element $e_i \in \xi$, where ξ is an event space which describes the set of possible music notes, is substituted into that position. In the bertso melody transformed in this work a 9 element vocabulary ξ is used, with pitch numbers from 60 to 74. If the semiotic label that corresponds to position i is not X, the substitution must be done in all the positions in the piece that have the same semiotic label, producing a new piece $\mathbf{e} = e_1, ..., e_i, ...e_l$ with an updated probability. The probability of the piece is computed using the single viewpoint model described in Conklin (2013) and presented in the equation below. Letting $v_i = \tau(e_i|e_{i-1})$ be the contour feature of event e_i in the context of its preceding event e_{i-1}, the probability of the piece e is computed as:

$$P(\mathbf{e}) = \prod_{i=1}^{\ell} P(v_i) \times P(e_i|v_i, e_{i-1})$$

$$P(e_i|v_i, e_{i-1}) = |\{x : \tau(x|e_{i-1}) = v_i\}|^{-1}$$

The probability is used to measure the *cross-entropy* of the piece; the mean negative log probability of an event in the piece, defined by $-\log_2 P(\mathbf{e})/\ell$. If the new cross-entropy is lower than the last saved one, it is saved and next iteration is executed on the new piece.

3. RESULTS

A small section of a bertso melody transformation is shown on Figure 1. The contour sequence of the original piece is shown, as well as its five point contour (dc5) sequence, where ld represents a leap down, sd is a step down, re is a repetition, su is a step up and lu is a leap up. The semiotic structure is represented with labels from A to E. The graph on Figure 2 shows how the cross-entropy of the piece varies through the iterations, in purple. The line has been smoothed with splines, so the direction of the cross-entropy can be seen.The lowest cross-entropy is reached before iteration 6000. Although the stochastic hill climbing method does not guarantee to find the optimal piece, it does improve the initial piece as seen on the green line of Figure 2.

4. CONCLUSIONS

In this work we present a method to transform a bertso melody preserving its musical structure, described by its semiotic structure. Semiotic structure is a good way to describe the coherence of the original piece, since all the repetitions on the piece that is being transformed are captured, and are going to be present on the transformation. Since assigning labels to all the notes on a piece can be too restrictive on sampling and it does not allow a wide variety of transformations, a contour reduction algorithm is used to allow some of the notes on the piece not to have a constraint. Using this method many different resulting melodies have been obtained, keeping the structure of the original piece, but they do not necessarily keep its repetitions. This can be a problem for bertso melodies, since they should be simple in order to make the process of creating and singing the bertso easier to the bertsolari. To solve this issue, pattern discovery methods in conjunction with semiotic structure labels are being explored.

5. ACKNOWLEDGEMENTS

This research is supported by the project Lrn2Cre8 which is funded by the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 610859.

6. REFERENCES

Bimbot, F., Deruty, E., Sargent, G., & Vincent, E. (2012). Semiotic structure labeling of music pieces: Concepts, methods and annotation conventions. In *13th International Society for Music Information Retrieval Conference (ISMIR)*, (pp. 235–240)., Porto, Portugal.

Conklin, D. (2003). Music generation from statistical models. In *Proceedings of the AISB 2003 Symposium on Artificial Intelligence and Creativity in the Arts and Sciences*, (pp. 30–35).

Conklin, D. (2013). Multiple viewpoint systems for music classification. *Journal of New Music Research*, *42*(1), 19–26.

Conklin, D. & Anagnostopoulou, C. (2006). Segmental Pattern Discovery in Music. *Informs Journal on Computing*, *18*, 285–293.

Conklin, D. & Bergeron, M. (2008). Feature set patterns in music. *Comput. Music J.*, *32*(1), 60–70.

Dubnov, S., Assayag, G., Lartillot, O., & Bejerano, G. (2003). Using machine-learning methods for musical style modeling. *Computer*, *36*(10), 73–80.

Garzia, J., Egaña, A., & Sarasua, J. (2001). *The art of bertsolaritza: improvised Basque verse singing*. Donostia, Bertsozale Elkartea; Andoain, Bertsolari Liburuak, 2001.

7

Melody Transformation with Semiotic Patterns

Authors: Izaro Goienetxea and Darrell Conklin.

Booktitle: Music Technology with Swing.

Year: 2018

Publisher: Springer

Melody transformation with semiotic patterns

Izaro Goienetxea[1] Darrell Conklin[1,2]

[1] Department of Computer Science and Artificial Intelligence
University of the Basque Country UPV/EHU, San Sebastian, Spain
[2] IKERBASQUE, Basque Foundation for Science, Bilbao, Spain
{izaro.goienetxea,darrell.conklin}@ehu.eus

Abstract. This paper presents a music generation method based on the extraction of a semiotic structure from a template piece followed by generation into this semiotic structure using a statistical model of a corpus. To describe the semiotic structure of a template piece, a pattern discovery method is applied, covering the template piece with significant patterns using melodic viewpoints at varying levels of abstraction. Melodies are generated into this structure using a stochastic optimization method. A selection of melodies was performed in a public concert, and audience evaluation results show that the method generates good coherent melodies.

1 Introduction

In recent years the topic of computational music generation has experienced a dynamic renewal of interest, though automation of music composition has intrigued people for hundreds of years. Even before the age of computers the idea of automatic music composition existed. A classical example of the automatic composition idea is the *Musikalisches Würfelspiel* or musical dice game, like the one published in 1792 that was attributed to Mozart [16].

Statistical models of symbolic music have been prevalent in computational modelling of musical style, since they can easily capture local musical features by training on large corpora rather than hand coding of stylistic rules [1, 7, 12, 15]. The lasting impact of statistical models on the topic of music generation spans from the earliest Markov models [5] to new variants of statistical models based on deep learning [4] and grammatical methods [20].

An issue faced by all methods for music generation is the *coherence problem*: ensuring that music material repeats or recalls in a more abstract sense material presented earlier in the piece. Nearly all forms of music involve repetition [17], either at the surface or deeper structural levels, and repetition imparts meaning to music [18]. Though early knowledge-based methods [13] explicitly considered repetition, the problem of achieving coherence in music generated from machine learning models remains largely unsolved.

A natural way to describe the coherence of a piece of music is by constructing a *semiotic structure*, defined as a representation of similar segments by a limited set of arbitrary symbols, each symbol representing an equivalence class

of segments [3]. A key observation is that a semiotic structure can be "inverted", generating new music by instantiating the symbols and retaining the abstract equivalence structure though having completely new music material [9]. The procedure can therefore be seen as *generation by transformation*: retaining abstract aspects of a template piece while modifying specific material.

Progress on the coherence problem was made recently in the music generation method of Collins et al. [6], where similar segments are identified by patterns indicating transposed repetitions in Chopin mazurkas. These "geometric" patterns are only suitable for carefully selected examples, because repetition in music need not be restricted to rigid transpositions. Consider, as an illustration, the simple melodic fragment of Figure 1. Though the two indicated phrases are clearly related, apparent in the score and to any listener, this is not by sharing an interval sequence, but rather an abstract contour sequence. The method described in this paper is able to naturally handle such musical phenomena using heterogeneous patterns discovered automatically using various viewpoints.

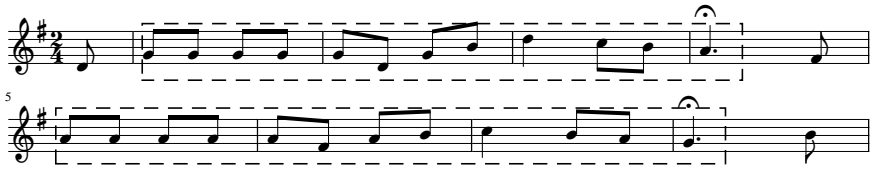


Fig. 1. First two phrases of the melody *Begiztatua nuen*[1]. The two phrases are related by an abstract melodic contour relation and there is no transposition that carries one into the other.

The style chosen to model is the folk style of *bertsos*. These are improvised Basque songs, sung by *bertsolaris*, that respect various melodic and rhyming patterns and which have fixed rhythmic structures. They can be classified into traditional folk melodies, new melodies, and melodies that are specifically composed. Bertso melodies usually have repeated and similar phrases, making them a challenge for statistical models and a good style for exploring the coherence problem. In this paper rhythmic aspects are conserved, so that the new melody can be used with lyrics created for the original melody.

The corpus used for this study is the *Bertso Doinutegia*, a collection of bertso melodies compiled by Joanito Dorronsoro and published for the first time in 1995 [14]. It currently contains 2379 melodies and is maintained and updated every year by Xenpelar Dokumentazio Zentroa[2] with new melodies that were used in competitions and exhibitions. Scores in the collection were encoded in Finale and exported to MIDI. Metadata associated with each song includes the

[1] http://bdb.bertsozale.eus/en/web/doinutegia/view/137-begiztatua-nuen-euskaldun-makila

[2] http://bdb.bertsozale.eus/es/

melody name, the name or type of the strophe, type of the melody, composer, bertsolari who has used it, name and location of the person who has collected the melody, and year of the collection. Some of the melodies in the collection have links to recordings of exhibitions or competitions where those melodies were used.

2 Methods

The transformation process presented in this paper has five main components: viewpoint representation; pattern discovery applied to a template piece to identify similar segments; pattern ranking and covering to form the semiotic structure; statistical model construction; and generation from the statistical model.

2.1 Viewpoint representation

To describe the template piece on different levels of abstraction a multiple viewpoint representation [9, 12] is used. A *viewpoint* τ is a function that maps an event sequence $e_1, \ldots, e_\ell$ to a more abstract derived sequence $\tau(e_1), \ldots, \tau(e_\ell)$, comprising elements in the codomain of the function τ.

viewpoint	codomain
pitch	$\{50, 52, 53, \ldots, 83\}$
dur	$\{1, 2, 3, \ldots\}$
onset	$\{0, 1, 2, \ldots\}$
intpc	$\{0, \ldots, 11\}$
int	$\{\ 14,\ \ 12,\ \ 11, \ldots, 14, 15, 17\}$
pc	{d, eq, u}
5pc	{ld, sd, eq, su, lu}
d pc	{d, eq, u}

Table 1. A specification for a small set of viewpoints.

Table 1 presents five melodic viewpoints pitch, int, intpc, pc and 5pc, and three rhythmic viewpoints dur, onset, and d pc. The viewpoint pitch represents the MIDI number of each event; the viewpoint int computes the interval between an event and the preceding one; the viewpoint intpc computes the pitch class interval (interval modulo 12) between an event and the previous one. A three-point contour viewpoint pc computes the melodic contour between two events: upward (u), downward (d) or equal (eq); and a five-point contour viewpoint 5pc computes whether the contour between two contiguous events is more than a scale step down (ld), is one scale step down (sd), is more than a scale step up (lu), is one scale step up (su), or stays equal (eq). The duration contour viewpoint

d pc computes if the duration of a note is shorter (d) than the previous one, longer (u) or equal (eq). The viewpoint representation of an example segment, using several viewpoints of Table 1, is shown in Figure 2.

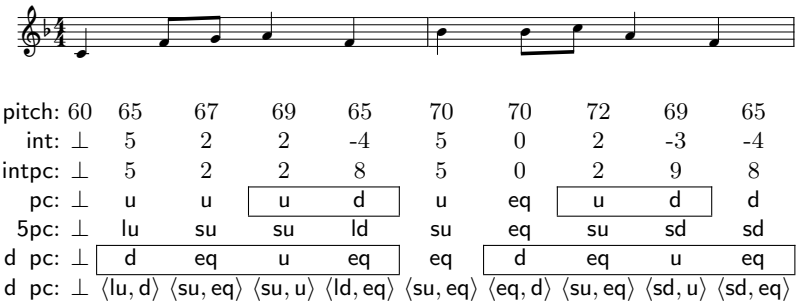


Fig. 2. A fragment from the melody *Abiatu da bere bidean*[3] and its viewpoint representation. Two patterns are highlighted.

To represent the interaction between melodic and rhythmic viewpoints, melodic viewpoints are *linked* with the rhythmic viewpoint d pc. A linked viewpoint $\tau_1 \otimes \tau_2$ represents events as pairs of values from its constituent viewpoints τ_1 and τ_2. Each new linked viewpoint is used to represent the template piece independently; using the four melodic viewpoints of Table 1 we get four different linked viewpoints: pitch ⊗ d pc, intpc ⊗ d pc, pc ⊗ d pc, and 5pc ⊗ d pc. An example representation of one of these (5pc ⊗ d pc) can be seen in Figure 2. To establish the semiotic structure, pattern discovery is performed on the template piece for each linked viewpoint independently.

2.2 Patterns and semiotic structure

To construct a semiotic structure of a template piece it is necessary to identify interesting repeated patterns which provide a dense covering of the template piece. Patterns are defined as sequences of event features described using viewpoints, and an event sequence instantiates a pattern if the components of the pattern are instantiated by successive events in the sequence. More precisely, a pattern of length m is a structure $\tau{:}(v_1, \ldots, v_m)$, where τ is a viewpoint and the v_i are elements of the codomain of τ. For example, in Figure 2 two simple patterns, each instantiated twice, are highlighted; pc:(u, d) and d pc:(d, eq, u, eq).

Patterns in a template piece can be found by applying a sequential pattern discovery method [2, 8] to each viewpoint representation of the template piece, identifying all patterns occurring more than once. This resulting list is then

[3] http://bdb.bertsozale.eus/en/web/doinutegia/view/2627-abiatu-da-bere-bidean

sorted according to an interestingness measure of patterns, and the ones that will form the coherence structure are chosen using a covering algorithm. These steps are now described in the remainder of this section.

Pattern distinctiveness and ranking Pattern interestingness is very important: in a given piece many patterns may exist but not all patterns are statistically or perceptually significant to a listener. For example, the pc pattern shown in Figure 2 would likely be instantiated many times in any template piece, but its occurrences (simply three notes with an up-down contour motion) are probably not structurally related or distinctive to the template piece, while the d pc pattern is more interesting. In order to build a good semiotic structure of the template piece, distinctive and interesting repetitions can be identified using a statistical method which provides the probability of seeing an indicated pattern at least the observed number of times in a template piece. Then a pattern is interesting if it occurs more frequently than expected. This is a standard model for assessing discovered motifs in music informatics [11] and bioinformatics [21].

More precisely, we derive a function $\mathbb{I}$ measuring the interest of a pattern. First, we note that the background probability p of finding a pattern $P = \tau : (v_1, \ldots, v_m)$ in a segment of exactly m events can be computed using a zero-order model of the corpus:

$$p = \prod_{i=1}^{m} \frac{c(v_i)}{c},$$

where $c(v_i)$ is the total count of the feature $\tau : v_i$ and c is the total number of places in the corpus where the viewpoint τ is defined. Then the binomial distribution $k; n, p\big)$ gives the probability of finding the pattern exactly k times in n events, and therefore the negative log probability of finding k or more occurrences of the pattern in a template piece with ℓ events is

$$\mathbb{I}(P) = \quad \ln \quad _{\geq} \; k; n, p\big), \tag{1}$$

where $_{\geq}$ is the upper tail of the binomial distribution, with $n = \ell \quad m+1$ being the maximum number of positions where the pattern could possibly occur in the template piece.

Template covering Following pattern discovery, the template piece is covered, trying to use the most interesting patterns but also striving for a dense covering. Though finding a covering jointly optimal in those requirements is intractable, a greedy method can be used to rapidly find a reasonable semiotic structure. In the greedy covering method, discovered patterns are sorted from most to least interesting using Equation 1, then this sorted list is processed to choose the patterns that fit into the positions of the template piece that have not been yet covered by any pattern, not allowing overlapping between contiguous patterns.

Example In Figure 3 the pattern structure of the template *Erletxoak lorean*[4] after the covering process is shown, patterns represented by the viewpoints pitch ⊗ d pc and pc ⊗ d pc. Above each pattern is the the viewpoint name, the pattern label, and the $\mathbb{I}$ value in brackets.

The template is a short piece with four phrases, having two sections in an overall $ABA'B$ structure. The music is syllabic with each phrase having 13 notes, in the key of Gm, briefly visiting B♭M in the third phrase (established at the high F♮). The B phrase is perfectly captured by a discovered pitch pattern, and though a few notes at the beginning of A and A' have not been covered by patterns, the discovered three-point contour pattern successfully captures the similarity between the second and fourth phrases. Note that there is no rigid transposition that relates these two phrases, but they have similar melodic contours that are captured by viewpoint patterns.

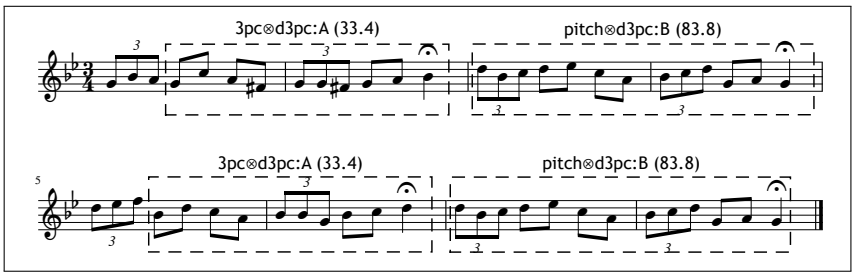


Fig. 3. Schema of a possible semiotic structure for the template piece *Erletxoak lorean*.

2.3 Statistical model

The semiotic structure defines the coherence within the template piece that will be conserved. To generate into the structure, stylistically coherent surface material is generated using a statistical model of the bertso corpus. In this work a trigram statistical model is built from a corpus to generate musical material into a template described by a semiotic structure. The exact probability of a piece using a trigram viewpoint model can be computed as described in [9]. Letting $v_i = \tau(e_i|e_{i\ \ 1})$ be the viewpoint τ value of event e_i in the context of its preceding event $e_{i\ \ 1}$, the probability of a piece $\mathbf{e} = e_1, \ldots, e_\ell$ is computed as:

$$\mathbb{P}(\mathbf{e}) = \prod_{i=3}^{\ell} \mathbb{P}(v_i|v_{i\ \ 1}, v_{i\ \ 2}) \times \mathbb{P}(e_i|v_i, e_{i\ \ 1}). \tag{2}$$

To elaborate, the product of all features in the sequence according to a trigram model is represented by the first term. Trigram probabilities of the viewpoint τ

[4] http://bdb.bertsozale.eus/en/web/doinutegia/view/241-erletxoak-lorean-orain-kantatuko-det-ii

are computed from the entire corpus. The second term is the probability of the particular event given the feature, defined as a uniform distribution over events having the property v_i:

$$\mathbb{P}(e_i|v_i, e_{i\ \ 1}) = |\{x \in \xi : \tau(x|e_{i\ \ 1}) = v_i\}|^{\ \ 1},$$

where ξ is the set of possible pitches (see Table 1).

The model above can be applied for any viewpoint. To select a viewpoint for modelling stylistic aspects of the bertso corpus in this study, every melodic viewpoint presented in Section 2.1 was evaluated with leave-one-out cross validation. Probabilities of every piece, according to Equation 2, were computed. Applied to the entire corpus of 2379 melodies, the product of all these probabilities gives a measure of the fit of the model to the corpus. The negative base-2 logarithm of this product is called the *cross-entropy* and lower cross-entropies are preferred. Every melodic viewpoint was tested, as were two linked melodic viewpoints intpc ⊗ 5pc and intpc ⊗ pc. The results of this procedure are shown in Table 2, which shows that the interval viewpoint int has the lowest cross-entropy on the corpus and is a good viewpoint to use for generation.

Viewpoint	Trigram Model
pc	4.45
pitch	2.62
int	2.55
intpc	3.83
5pc	3.38
intpc ⊗ 5pc	2.71
intpc ⊗ pc	3.13

Table 2. Cross-entropy of different viewpoints, determined by leave-one-out cross validation on the corpus.

2.4 Generation

In this work a semiotic structure is used along with the trigram statistical model to generate new melodies. Generated sequences having high probability are assumed to retain more aspects of the music style under consideration than sequences with low probability. The process of *optimization* is concerned with drawing high probability sequences from statistical models.

A stochastic hill climbing optimization method is used to obtain high probability melodies. The method starts with a random piece that respects the coherence structure extracted from the template piece, using pitches from a pitch set ξ' that defines the admissible pitches for the generated piece. This set is typically the scale defined by the desired tonality of the generated piece and will

be a subset of the complete pitch domain ξ. This initial piece is created with a left-to-right random walk, which samples a new note in every position of the template, and every time a complete pattern is instantiated, all of the future locations of the pattern are also instantiated, in this way conserving the original relation between them. The piece is then iteratively modified: in each iteration of the process a random location i in the current piece $\mathbf{e}$ is chosen. A pitch e_i is uniformly chosen from ξ' and is substituted into that position, producing a new piece $\mathbf{e}'$ with an updated probability $\mathbb{P}(\mathbf{e}')$. If $\mathbb{P}(\mathbf{e}') > \mathbb{P}(\mathbf{e})$, then $\mathbf{e}'$ is taken as the new current piece. Every time a position is changed, the pattern to which that note belongs is identified, and all other instances of that pattern are also updated. Thus at every iteration the generated piece conserves the semiotic structure. The optimization process is iterated up to 10^4 times, and after each update the probability of the new piece is computed using Equation 2. If the new probability is higher than the last saved one the change is retained.

3 Results

To illustrate the generality of the method, new melodies are generated using two different templates, and properties of generated melodies are discussed. For the second template, two songs were peformed and evaluated by an audience in a live concert setting in a jazz club in London.

3.1 Illustration on a full piece

The template used is *Erletxoak lorean*, which was discussed earlier in Figure 3. The pitch vocabulary used is $\xi' = \{66, 67, 69, 70, 72, 74, 75, 77\}$ and two different viewpoints were used for the statistical model (Equation 2): 5pc and int. The three transformations shown in Figure 4 conserve the semiotic structure shown in Figure 3. The first transformation contains within the B phrase a leap down by a diminished seventh, which though perhaps difficult to sing is interesting and is resolved properly by a step up. The A and A' phrases are somewhat reserved in their ambitus, though A contains an interesting ascending broken triad. The second transformation follows an overall smooth melodic contour and is a singable melody with internal coherence. Its shortcoming might be identified within the A' phrase which has a non-idiomatic leap which further exposes an F♯ and F♮ together in close proximity. This could be corrected by including another segmental viewpoint to ensure that the scale of each phrase is internally coherent. The final transformation of Figure 4, done with the int viewpoint as the statistical model, corrects to some extent the problems with excessive leaps with the general 5pc model, but is confined to a rather small ambitus.

3.2 London concert and listener evaluation

A small suite of pieces was performed live in a public concert named "Meet the Computer Composer" at the Vortex Jazz Club in London on September 28,

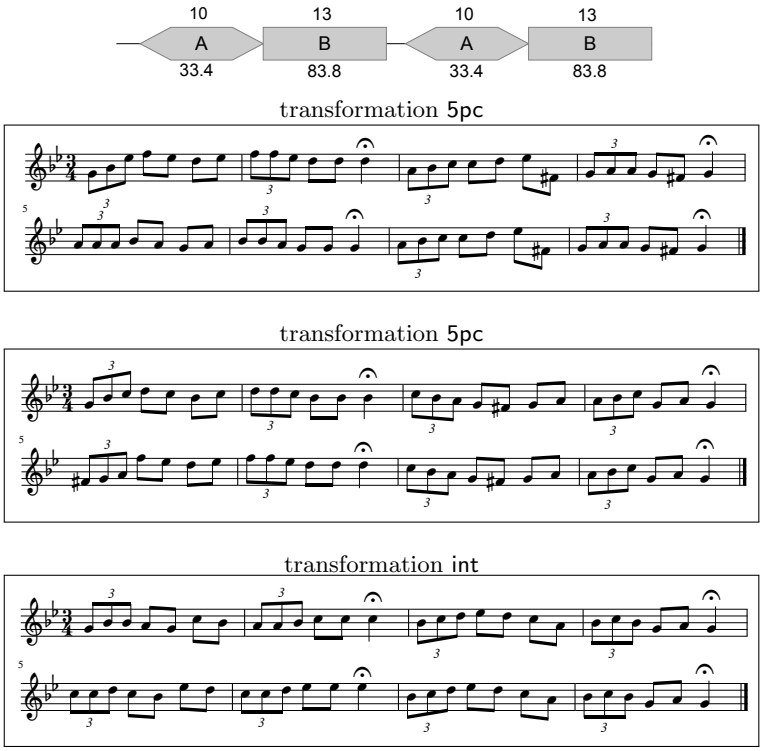


Fig. 4. Three transformations of the template piece *Erletxoak lorean*. Top: its semiotic structure with the number of notes in each pattern and their $\mathbb{I}$ value. The first two transformations use a 5pc statistical model and the bottom one uses a int model.

2016. A bertso melody *Txoriak eta txoriburuak* was sung (by the first author IG) along with two generations that used the original as a template. The full scores of all three melodies can be seen in Figure 5. Following the bertso tradition of new lyrics to existing melodies, the three melodies were sung each with the same new lyrics that were specially written for the concert.

An audience questionnaire (Table 3, top) was given at the beginning of the concert to all the members of the audience, where they would note which one of the three melodies they thought was the original, and how confident they were in their decision. A total of 52 questionnaires (from approximately 100 distributed) was returned. In Table 3 the results obtained from the questionnaires can be seen. The majority (55%) of respondents incorrectly identifed one of the two transformations as the original piece, though the 44% identifying correctly the original had overall higher confidence in their decision. Regarding transformation 1, it must be noted that this was the first of three pieces performed, and the singer had not yet achieved perfect intonation: this no doubt affected the

lower (15%, with 37.5% not confident in their response) audience result for that transformation.

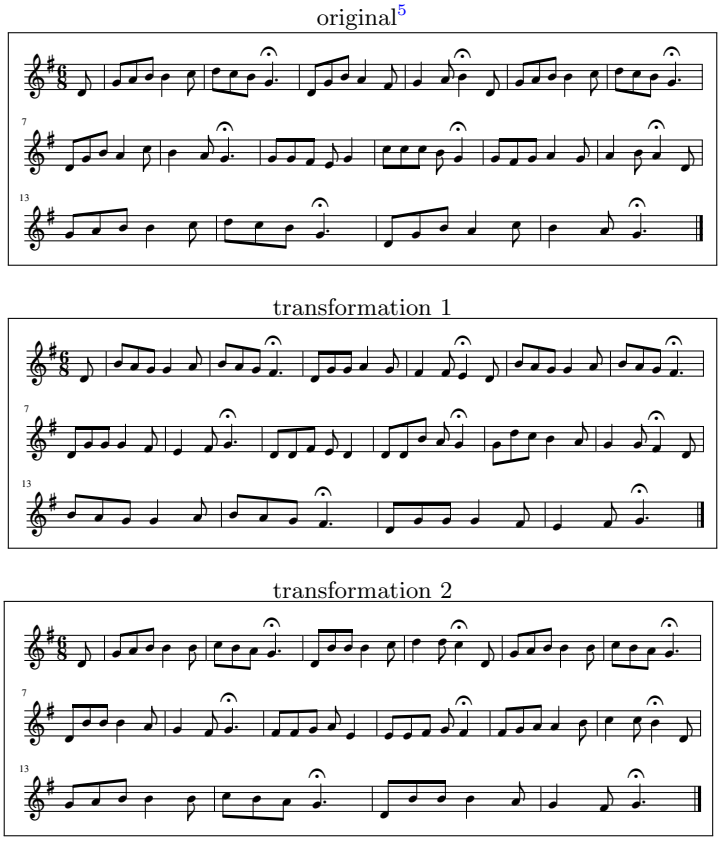


Fig. 5. Three pieces performed at the London concert.

4 Conclusions and future work

In this paper a method for transforming bertso melodies conserving the internal coherence of a template piece is presented. The basis of the method is a trigram statistical model combined with the strong constraints provided by a semiotic structure, which is identified using a sequential pattern discovery algorithm followed by a pattern ranking and covering method. New musical content is created

[5] http://bdb.bertsozale.eus/en/web/doinutegia/view/1564-txoriak-eta-txoriburuak

Which piece is the original?
1 2 3

How confident are you on a scale of 1 to 5? (1=not confident, 5=very confident)
1 2 3 4 5

	transformation 1					transformation 2					original				
is original?	8 (15%)					21 (40%)					23 (44%)				
confidence	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
%	37.5	50	0	12.5	0	38	28.6	23.8	4.8	4.8	26	23.8	23.8	23.8	8.7

Table 3. Top: the audience questionnaire distributed at the London concert. Bottom: results obtained. The original piece was the third melody sung.

using a statistical model which iteratively changes a template piece to improve the final result.

The generation method presented in this paper extends the method of Collins et al. [6] in some important ways. Not restricted to patterns conserving exact intervals, the method here allows a heterogeneous semiotic structure comprising a variety of abstract viewpoints. The generated pieces are not single random walks from a model, rather some effort is made to generate high probability solutions which are expected to be more stylistically valid. The method can be extended to polyphony and some initial work in those directions has been completed for counterpoint generation in the style of Palestrina [19] and multilayer textures in electronic dance music [10].

Acknowledgments This research was supported by the project Lrn2Cre8 (2013-2016) which was funded by the Future and Emerging Technologies (FET) programme within the Seventh Framework Programe for Research of the European Commission, under FET grant number 610859. The authors thank the Xenpelar Dokumentazio Zentroa for their enthusiasm in the project and for sharing the Bertso Doinutegia. Thanks to Kerstin Neubarth for valuable discussions on the research and the manuscript.

References

1. M. Allan and C. K. I. Williams. Harmonising chorales by probabilistic inference. In *Advances in Neural Information Processing Systems*, pages 25–32, 2004.
2. J. Ayres, J. Flannick, J. Gehrke, and T. Yiu. Sequential PAttern Mining Using a Bitmap Representation. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 429–435, Edmonton, Alberta, Canada, 2002.

3. F. Bimbot, E. Deruty, G. Sargent, and E. Vincent. Semiotic structure labeling of music pieces: Concepts, methods and annotation conventions. In *13th International Society for Music Information Retrieval Conference (ISMIR)*, pages 235–240, Porto, Portugal, 2012.
4. N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. In *International Conference on Machine Learning*, Edinburgh, Scotland, 2012.
5. F. P. Brooks, A. L. Hopkins Jr., P. G. Neumann, and W. V. Wright. An experiment in musical composition. *IRE Transactions on Electronic Computers*, EC–5:175–182, 1956.
6. T. Collins, R. Laney, A. Willis, and P. H. Garthwaite. Developing and evaluating computational models of musical style. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 30:16–43, 2016.
7. D. Conklin. Music generation from statistical models. In *Proceedings of the AISB 2003 Symposium on Artificial Intelligence and Creativity in the Arts and Sciences*, pages 30–35, Aberystwyth, Wales, 2003.
8. D. Conklin. Discovery of distinctive patterns in music. *Intelligent Data Analysis*, 14(5):547–554, 2010.
9. D. Conklin. Chord sequence generation with semiotic patterns. *Journal of Mathematics and Music*, 10(2):92–106, 2016.
10. D. Conklin and L. Bigo. Trance generation by transformation. In *8th International Workshop on Music and Machine Learning (MML 2015) at the 21st International Symposium on Electronic Art*, pages 4–6, Vancouver, 2015.
11. D. Conklin and S. Weisser. Pattern and antipattern discovery in Ethiopian bagana songs. In D. Meredith, editor, *Computational Music Analysis*, pages 425–443. Springer, 2016.
12. D. Conklin and I. H. Witten. Multiple viewpoint systems for music prediction. *Journal of New Music Research*, 24:51–73, 1995.
13. D. Cope. An expert system for computer–assisted composition. *Computer Music Journal*, 11(4):30–46, 1987.
14. J. Dorronsoro. *Bertso Doinutegia*. Euskal Herriko Bertsolari Elkartea, 1995.
15. S. Dubnov, G. Assayag, O. Lartillot, and G. Bejerano. Using machine-learning methods for musical style modeling. *Computer*, 36(10):73–80, 2003.
16. S. A. Hedges. Dice music in the eighteenth century. *Music & Letters*, 59(2):180–187, 1978.
17. J. Leach and J. Fitch. Nature, music, and algorithmic composition. *Computer Music Journal*, 19(2):23–33, 1995.
18. L. B. Meyer. Meaning in music and information theory. *Journal of Aesthetics and Art Criticism*, 15:412–424, 1957.
19. V. Padilla and D. Conklin. Statistical generation of two-voice florid counterpoint. In *Proceedings of the Sound and Music Computing Conference (SMC 2016)*, pages 380–387, Hamburg, Germany, 2016.
20. D. Quick and P. Hudak. Grammar-based automated music composition in Haskell. In *Proceedings of the 1st ACM SIGPLAN Workshop on Functional Art, Music, Modeling & Design*, FARM '13, pages 59–70, 2013.
21. J. van Helden, B. André, and J. Collado-Vides. Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. *Journal of Molecular Biology*, 281(5):827–842, 1998.

8

Statistics based music generation approach considering both rhythm and melody coherence

Authors: Izaro Goienetxea, Iñigo Mendialdua, Igor Rodríguez and Basilio Sierra.

Journal: Submitted to IEEE Access

Submitted in Ju ly 2019

Year: 2019

Publisher: IEEE

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Statistics based music generation approach considering both rhythm and melody coherence

IZARO GOIENETXEA[1], IÑIGO MENDIALDUA[2], IGOR RODRÍGUEZ[1], AND BASILIO SIERRA.[1]
[1]Department of Computer Science and Artificial Intelligence, University of the Basque Country UPV/EHU, San Sebastian, Spain
[2]Department of Computer Languages and Systems, University of the Basque Country UPV/EHU, San Sebastian, Spain

Corresponding author: Izaro Goienetxea (e-mail: izaro.goienetxea@ehu.eus).

This work has been partially supported by the Basque Government Research Teams grant (IT900-16) and the Spanish Ministry of Economy and Competitiveness RTI2018-093337-B-I00 grant.

ABSTRACT This paper presents a music generation method based on the use of coherence structures extracted from a template piece and statistical models built from a corpus of monophonic melodies. Independent coherence structures are created to describe the most interesting melodic and rhythmic relations between segments in the template piece. To do so a pattern discovery and ranking method is applied in different abstractions of the template piece, and the most interesting ones are selected to form the coherence structure. Two statistical models are also built from the corpus: a statistical model that characterizes the melodic features of the corpus and a rhythmic one that characterizes its rhythmic features. New melodies are created by sampling new notes into the coherence structures according to the statistical models. A stochastic optimization method is used to generate high probability melodies. An evaluation of some of the generated melodies was carried out and the results show that the method generates good coherent melodies that could be original pieces.

INDEX TERMS coherence, computer generated music, rhythm generation, statistical models,

I. INTRODUCTION

Automatic generation of music has a long history of research since the creation of the first computer in 1840 by Lovelace and Babbage [1], but the idea of composing music automatically has existed even before the existence of computers. Some examples of this idea are the *Musikalisches Würfelspiel* or musical dice games, like the one published in 1792 that was attributed to Mozart [2].

The earliest automatically generated compositions are from the mid-1950s, around the same time as the concept of Artificial Intelligence was coined. Among the first automatically generated compositions are those of Lejaren Hiller and Leonard Isaacson from 1955-56 [3]. Since these first steps many different algorithms have been developed to compose music automatically, such as knowledge based systems, evolutionary and other population-based methods, fractals or statistical models [4].

Statistical models of symbolic music have been used in computational modelling of several musical styles, for which many computational approaches have been developed [5]–[8]. The main advantage they offer is that they can be learned from a corpus of music to extract its musical features. These features can be then used to generate new musical sequences that reflect an explicit musical style [9]–[11].

An important issue that needs to be taken into consideration when generating music automatically is the coherence of the generated pieces. New pieces should contain material that is related (by repetition or a more abstract relation) to segments seen earlier in the piece, in order to endow it with some musical meaning. Different theories have been developed on how the music should be structured in order to be comprehensible. Arnold Schoenberg [12] believed that laws are needed to write music; acoustic laws and laws that result from the combination of time and sound. According to him listeners have to recognize musical figures and how they cohere in order to comprehend what they are listening.

Some theories compare musical discourse and linguistics, as well as the mechanisms the human brain has to understand them [13], [14]. These works suggest that, as in linguistics, relations between different segments in musical pieces are necessary to build a coherent discourse. The most obvious relation between musical segments is repetition. It is a fact

that almost all forms of music involve repetition [15], either of sequences of pitch of notes or at some higher level of structural grouping, and that repetition imparts a sense of meaning to music [16]. These repeated segments are named *motives*, where a motif is defined as "the smallest part of a piece or a section of a piece that, despite change and variation, is recognizable as present throughout" [17]. Though early knowledge-based methods [18] explicitly considered repetition, the problem of achieving coherence in music generated from machine learning models remains largely unsolved. Some approaches have been developed to deal with the coherence of the generated music, like the description of its acoustic structure, functional structure or semiotic structure. Semiotic structure is defined as the representation of similar segments by similar arbitrary symbols [19]. Once the semiotic structure of a piece is described, the process can be "inverted", to generate new music by instantiating the symbols of the structure, getting pieces with new music material but the same coherence structure of the original piece.

In this work the music generation method presented in [20] is extended. It is a work based on the use of the coherence structure of a template piece along with a statistical model of a corpus to generate new melodies that have the same coherence structure as the template piece and that also share some melodic features with the corpus. In this extension of the method, in addition to the melodic transformation a rhythmic generation step is added, following the same idea of using a coherence structure of a template and statistical models.

As well as in [20], the style chosen to model in this work is the folk style of bertsos. The art of singing these improvised Basque songs is called *bertsolaritza* or *bertsolarism*, and they are sung by *bertsolaris*. Bertsos must respect various melodic and rhyming patterns, and their rhythmic structure has to fit in one of the many accepted metrics. They are defined as sung, rhymed and metered discourses by the book *The Art of Bertsolaritza: Improvised Basque Verse Singing* [21]. There is evidence of bertso singing and written bertso poem samples since the 15th century, and it is a very popular art nowadays in the Basque Country. Bertsos are sung in many different occasions, like informal lunches with friends, homage ceremonies or competitions, and any topic can occur in a bertso. Many bertsolarism competitions take place every year in the Basque Country, and every four years the national championship final is held, with around 15000 people in attendance.

Experts say the chosen melody for singing a bertso and the manner in which it is sung can be the key for the communicative success of the bertsolari, since the chosen melody must be able to combine with the created lyrics to transmit what the bertsolari wants to express with the bertso. These melodies can be traditional folk melodies, new melodies that have an appropriate rhythmic structure and melodies that are specifically composed. Bertso melodies usually have repeated and similar phrases, making them a challenge for statistical models and a good style for exploring the coherence problem.

The rest of the paper is organized as follows. Section II gives an overview of the related work of the field of automatic music generation, Section III describes the corpus used in this work and Section IV gives a complete description of the presented generation method. In Section V some of the obtained results are shown and in Section VI how they have been evaluated is explained. Finally, in Section VII the extracted conclusions and the identified future work are presented.

II. RELATED WORK

Several approaches have been developed in automatic music generation that, even though there is not a fixed taxonomy of these methods, are often classified as knowledge-based (or rule-based), evolutionary methods, machine learning methods or hybrids.

Knowledge-based methods use pre-made sets of arguments or rules that describe a style or genre, to compose music on the same style or genre. Some examples of this generation type are the grammar models and the rule learning methods. Grammar models produce musical pieces using rules, which expand high level symbols into detailed sequences of symbols (words). These rules can be hand coded by an expert or they can be learned from a corpus of melodies that share a genre or style. An example of the use of grammars for music generation is the one developed by Chemillier [22], which generates jazz chord sequences based on Steedman's grammar. This grammar was created from a set of modern jazz 12-bar chord sequences, which is considered a wide and representative range of permissible variations of the blues basic form.

Evolutionary methods are based on the improvement of a population by cycles of evaluation and reproduction with variation of its individuals. The process starts with the generation of the candidate solutions of the initial set, then in each cycle the candidates are changed by mutation or recombination and they are evaluated using a fitness function. These cycles are repeated until a stopping criteria is satisfied.

Evolutionary algorithms have been used in different tasks of music generation like in *GenJam* [23], an interactive jazz improvisation system. GenJam uses a training process, in which the system plays a tune and a mentor human evaluates it as good or bad. These evaluations are then used to adjust the fitness function. Another example of the use of evolutionary algorithms for music composition is *MetaCompose* [24], which is a component-based system for music generation that supports real-time improvisation. The composition process has three main steps: creation of a chord sequence, evolution of a melody fitting this chord sequence and creation of an accompaniment for the melody/chord sequence combination.

Machine learning methods extract the knowledge from a corpus instead of having it previously defined. Statistical models are an example of machine learning methods, in which different features of a corpus can be represented in

a model that will be able to assign probabilities to automatically generated melodies.

Statistical models of music have been used for generation of melodies and harmonies in several works, and they go from the earliest Markov models [25] to new models based on deep learning [26], [27]. Whorley and Conklin [28] use statistical models to generate four-part harmonizations using horizontal and vertical viewpoints of music and an iterative random walk sampling. Herremans et al. [29] use a first order statistical model to capture the melodic and harmonic features of a first species counterpoint corpus and generate new musical content. To do so, they use a sampling method named Variable Neighborhood Search (VNS), which starts with a randomly generated fragment and optimize it making local changes to increase the probability of the fragment, and they compare it to other sampling methods like random walk or Gibbs sampling. Padilla and Conklin [30] have developed a method to compose Palestrina masses using a combination of statistical models, to capture the stylistic aspects of the music, and pattern discovery to extract the coherence structure of original melodies. The pattern discovery process is performed on a single viewpoint representation of the melodies, and the patterns used to build the coherence structure are used to guide the generation of new musical material. Collins et al. [31] consider the coherence problem for generating new melodies by defining a template from an existing polyphonic piece to sample new notes onto it. Geometric patterns are discovered in a point representation of the notes in a pitch-time space for which a pattern discovery method named SIACT [32] is used, which is able to discover exact repetitions and transposed segments.

Deep learning architectures are used more and more in music generation, and well known groups like Magenta[1] at Google are using them to generate new music. Deep learning is defined as a repertoire of machine learning techniques based on artificial neural networks which have multiple layers to process multiple abstraction layers of the data [33], and several approaches to automatically create music using these techniques have been developed. Some works [34] use a unit selection methodology to analyse if using only the units available in a library can be enough to generate a wide spectrum of new musical content. They then use a combination of a Deep Structured Semantic Model (DSSM) and an Long Short-Term Memory (LSTM) to predict the next unit in the generation model. Other generation works are based on the use of GAN (Generative Adversarial Network) to create music, like MidiNet [35] and MuseGAN [36]. Even if this is a growing area of research and interesting results are obtained using these architectures for music generation, they still have some limitations, like the control (of tonality conformance, rhythm...), structure (giving direction to the generated music), creativity (versus imitation) and interactivity [33].

[1]https://magenta.tensorflow.org/

III. CORPUS

The corpus used in this work is the *Bertso doinutegia*, a collection created by Joanito Dorronsoro and published for the first time on 1995 [37]. It is maintained and updated every year by *Xenpelar Dokumentazio Zentroa*[2] with new melodies that are used in competitions and exhibitions. Entries in the collection have a melody name, the name or type of the strophe and type of the melody (genre), among other information. Melodies are classified into 17 different types or genres, and 381 of the melodies in the collection have links to recordings of exhibitions or competitions where those melodies were used.

The scores included in the collection have been encoded in Finale and exported to MIDI. Currently the collection is composed of 2382 bertso melodies, which have a mean length of 60 notes. 85 of the melodies are polyphonic or have polyphonic parts. Since in this work monophonic pieces are generated, all the pieces with polyphony are processed using a *skyline* method, which takes the event with highest pitch at unique onset times. Three of the 85 polyphonic MIDIs have some format error that makes them unusable, so they have been removed from the corpus.

Since in this work statistical models are built to capture different aspects of the corpus, it has been studied in order to detect anomalies within the pieces that could affect on the creation or use of the statistical models. It has been found that many pieces have long sequences of repeated notes. Sequences of four repeated notes have been discovered occurring at least twice in 744 pieces (31% of the corpus), and at least once in 1064 (44.7% of the corpus). A model built from these sequences would assign high probability to long note repetitions, and even though these sequences exist in the corpus of the style that is being replicated, it has been decided that the generation of such sequences should be avoided because of their lack of interest. Figure 1 shows the score of the melody with the highest number of sequences of repeated notes, which clearly is not musically interesting.

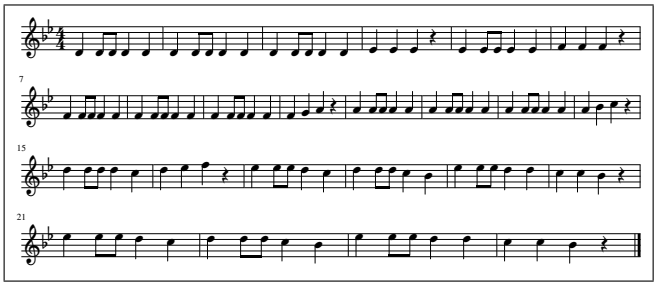


FIGURE 1: Score of the melody Neska zaharrak eta apaizak II

To avoid the negative effect that these pieces could have on the statistical model, the pieces with the highest proportion of repeated notes are removed from the corpus, reducing the size of the corpus to 1934 pieces (82.2% of the original corpus).

[2]http://bdb.bertsozale.eus/es/

IV. PROPOSED METHOD

The method presented in this paper is an extension of the work presented in [20], which is able to generate new melodic information maintaining the rhythm of an existing piece. In this approach, in addition to the melodic information, the rhythmic part is also generated. The generation process is based on the use of an abstract template that consists of a melodic coherence structure and a rhythmic coherence structure, both extracted from an existing piece, and two statistical models, a melodic model and a rhythmic one. The coherence structures ensure that the final pieces have related segments within them, while the statistical models are able to capture certain melodic and rhythmic features of a corpus that are then reflected in the generated material. In Figure 2 a diagram of the presented method can be seen. It can be seen that the melodic and the rhythmic information are treated independently, both for building the coherence structures and the statistical models and for generating new musical content. Each of the components of the figure are described in more detail below.

A. COHERENCE STRUCTURES

The aim of the coherence structure is to describe the relations between similar segments of a piece. In this work, since melodic and rhythmic relations are considered, the coherence of the template is described by two independent components; the melodic coherence structure and the rhythmic coherence structure. It has been decided to create independent structures to describe melodic and rhythmic coherence, because in bertso melodies many rhythm repetitions can occur where the melodic content is different between the repetitions. In Figure 3 an example is shown where a rhythmic pattern is highlighted. The pattern has a length of 19 components and is repeated four times through the piece, but there is no any melodic relation between all the occurrences of the pattern. It has been considered that these cases make it necessary to have independent coherence structures for the melodic relations and the rhythmic ones.

The melodic coherence structure needs to capture the most relevant melodic similarity relations between segments in the piece, while the rhythmic coherence structure must describe the rhythmic repetitions between segments. Similarity relations of various abstraction level are considered to be included in the melodic coherence structure, in order to be able to capture not only the more obvious relations like repetition or transposition, but also the more abstract ones. Since in bertso melodies exact rhythmic repetitions are very common, only this type of relation is represented in the rhythmic coherence structure of a piece.

The process to build both of the coherence structures has three main steps: viewpoint representation, pattern discovery and pattern ranking and covering. They are explained below.

1) Viewpoint representation

In order to analyse the template piece on different abstraction levels a multiple viewpoint representation [38] is used. A viewpoint τ is a function that maps an event sequence $e_1, \ldots, e_\ell$ to a more abstract sequence $\tau(e_1), \ldots, \tau(e_\ell)$, comprising elements in the codomain of the function τ. In the building process of the melodic coherence structure notes are the only events that are taken into account.

In Table 1 some melodic viewpoints (pitch, int, intpc, 3pc and 5pc), and three rhythmic viewpoints (dur, intDur, and d3pc) are presented. The viewpoint pitch represents the MIDI number of each note; the viewpoint int computes the interval between a note and the preceding one; the viewpoint intpc computes the pitch class interval (interval modulo 12) between a note and the previous one. A three-point contour viewpoint 3pc computes the melodic contour between two notes: upward (u), downward (d) or equal (eq); and a five-point contour viewpoint 5pc computes whether the contour between two contiguous notes goes more than a scale step down (ld), goes one scale step down (sd), goes more than a scale step up (lu), goes one scale step up (su), or stays equal (eq). The duration viewpoint dur represents the duration of each note, while durInt represents the relation between the durations of two contiguous events. Contour viewpoint d3pc computes if the duration of a note is shorter (d) than the previous one, longer (u) or equal (eq). The representation of an example segment, using several viewpoints of Table 1, is shown in Figure 4.

viewpoint	codomain
pitch	$\{50, 52, 53, \ldots, 83\}$
dur	$\{1, 2, 3, \ldots\}$
int	$\{-14, -12, -11, \ldots, 14, 15, 17\}$
3pc	{d, eq, u}
5pc	{ld, sd, eq, su, lu}
d3pc	{d, eq, u}
durInt	$\{1/16, 1/8, 1/4 \ldots\}$

TABLE 1: A specification for a small set of viewpoints. Top: two basic attributes of notes; bottom: derived viewpoints.

Melodic viewpoints

In this work melodic and rhythmic coherence are independently analysed, so they have been independently represented. The template pieces are represented using pitch, int, 3pc and 5pc melodic viewpoints, to be able to capture relations between segments in different abstraction levels.

Rhythmic viewpoints

Since in this approach of the generation method only exact rhythmic repetitions are considered, dur is the only viewpoint used to represent the template piece, which indicates the duration in ticks of each event in the piece.

2) Pattern discovery

To construct the coherence structure of a template piece it is necessary to discover and join the interesting patterns that are repeated through every viewpoint representation of the

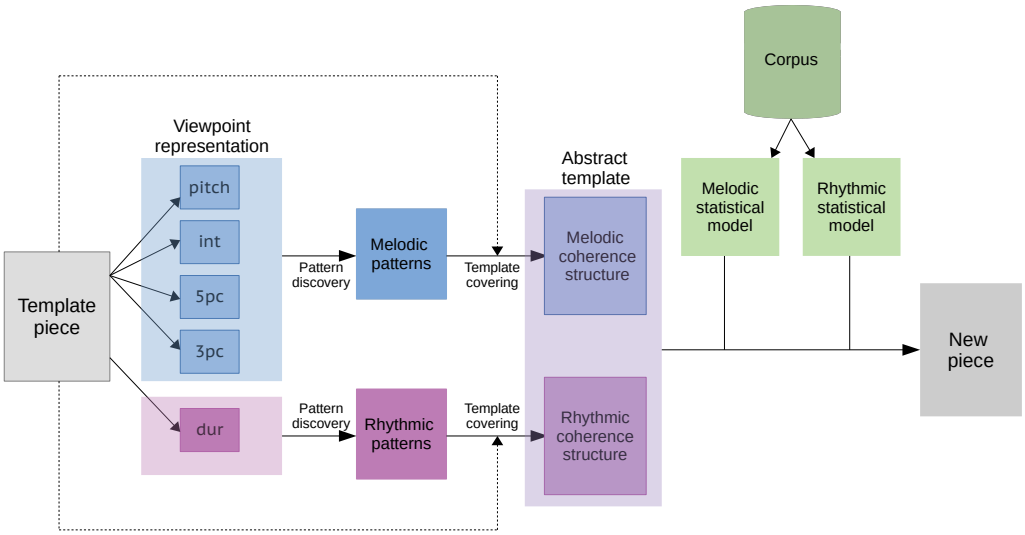


FIGURE 2: Diagram of the method proposed in this work.

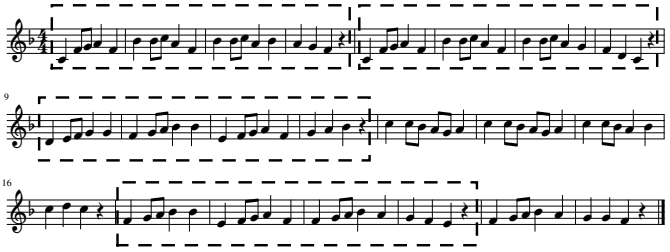
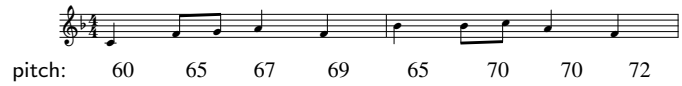


FIGURE 3: Score of the melody *Abiatu da bere bidean* where the main rhythmic pattern is highlighted.



pitch:	60	65	67	69	65	70	70	72
int:	⊥	5	2	2	-4	5	0	2
3pc:	⊥	u	u	u	d	u	eq	u
5pc:	⊥	lu	su	su	ld	su	eq	su
dur:	1024	512	512	1024	1024	1024	512	512
durInt:	⊥	1/2	1	2	1	1	1	1/2
d3pc:	⊥	d	eq	u	eq	eq	d	eq

FIGURE 4: A fragment from the melody *Abiatu da bere bidean*and its viewpoint representation. A pattern of length two is highlighted.

piece and cover it in the most dense way possible. Patterns are defined as sequences of event features (or viewpoints), and a piece instantiates a pattern if the pattern occurs (one or multiple times) in the sequence: if the components of the pattern are instantiated by successive events in the sequence [39]. More precisely, a pattern of length m is a structure $\tau : (v_1, \ldots, v_m)$, where τ is a viewpoint and the v_i are elements of the codomain of τ. For example, in Figure 4 a simple pattern $\mathsf{d3pc}{:}(\mathsf{u}, \mathsf{d})$ is highlighted.

Pattern discovery methods identify segments that are repeated through a symbolic representation of a musical piece or a corpus. In this work a pattern discovery algorithm named SPAM [40] has been used to identify similar segments. This is an algorithm that finds all the frequent sequential patterns (patterns that occur more times than a given threshold) in a transactional database, specially efficient with large databases, but the method has been adapted to be used to discover patterns within a single sequence. Candidates are created with a depth-first search strategy and various pruning mechanisms are used to reduce the search space.

3) Pattern ranking and covering

The pattern discovery algorithm produces all the patterns that appear more times than a given threshold and it does not have a more sophisticated method to rank the discovered patterns. One way to do this is by measuring the distinctiveness of each pattern [41], but in this work the priority is not to find patterns that are over-represented in a piece with respect to an anticorpus. The priority is to find patterns that are significant in the piece, in terms of occurrence and length.

Measuring the interest of the discovered patterns is a very important task; many patterns can occur in a piece, but not all of them are important enough to be used on the building of the coherence structure. For example, the d3pc pattern shown in Figure 4 would likely be instantiated many times in any piece, but its occurrences (simply three notes with an up-down duration contour motion) are probably not structurally related or distinctive to the template piece. In order to build a good coherence structure of the template piece, distinctive and interesting repetitions should be identified using a statistical method which provides the probability of seeing an indicated pattern at least the observed number of times in a template piece. Then a pattern is considered interesting if it occurs more frequently than expected. This is a standard model for assessing discovered motifs in music informatics [42] and bioinformatics [43].

We derive a function $\mathbb{I}$ measuring the interest of a pattern.

First, we note that the background probability p of finding a pattern $P = \tau{:}(v_1, \ldots, v_m)$ in a segment of exactly m events can be computed using a zero-order model of the corpus:

$$p = \prod_{i=1}^{m} \frac{c(v_i)}{c},$$

where $c(v_i)$ is the total count of the feature $\tau : v_i$ and c is the total number of places in the corpus where the viewpoint τ is defined. Then the binomial distribution $\mathbb{B}\big(k; n, p\big)$ gives the probability of finding the pattern exactly k times in n events,

$$\mathbb{B}(k; n; p) = \binom{n}{k} p^k (1-p)^{n-k}$$

and therefore the negative log probability of finding k or more occurrences of the pattern in a template piece with ℓ events is

$$\mathbb{I}(P) = -\ln \mathbb{B}_{\geq}\big(k; n, p\big), \tag{1}$$

where $\mathbb{B}_{\geq}$ is the upper tail of the binomial distribution, with $n = \ell - m + 1$ being the maximum number of positions where the pattern could possibly occur in the template piece.

The interest of all patterns discovered within all the viewpoint representations is computed and, in order to discard patterns that are not important enough for the coherence structure, an interest threshold is set. This threshold is set at 9.5, as a result of an analysis of the discovered patterns performed by hand, comparing different patterns, their significance in the score they are discovered in, and their interest value. As an example, in Figure 5 a score of the melody *Argi emaile txit diztiratsu* is shown, which is part of the corpus and where a three element pattern is highlighted. It is the pattern $\mathsf{pitch}{:}(64, 65, 67)$ which has an interest value of 9.2 and, given its length and information it represents, is not considered distinctive enough. The interest threshold is set at 9.5 to avoid this kind of patterns in the coherence structures.

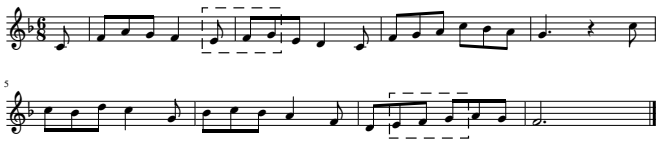


FIGURE 5: Score of the melody *Argi emaile txit diztiratsu*.

In Figure 6 the score of the melody *Lagundurikan danoi I* is shown, where the melodic contour pattern $\mathsf{3pc} : (u, u, d, d, eq, d, eq, d)$ is highlighted. The interest value of this pattern is 9.55, it is near the established interest threshold, but it is considered a good enough pattern, taking into account the amount of information it represents, showing that the chosen threshold is acceptable.

Once all the patterns in the different viewpoint representations of the template are discovered and their interest values are computed, they are used to cover the piece, trying to use the most interesting patterns but also striving for a dense covering. Since finding a covering that optimally fulfils both

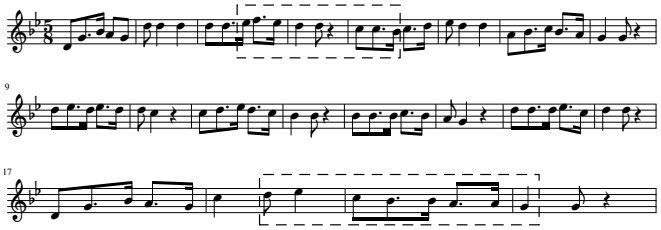


FIGURE 6: Score of the melody *Lagundurikan danoi I*.

requirements is not easy, a greedy method can be used to rapidly find a reasonable semiotic structure. In the greedy covering method, discovered patterns are sorted from most to least interesting using Equation 1, then this sorted list is processed to choose the patterns that fit into the positions of the template piece that have not been yet covered by any pattern, not allowing overlapping between contiguous patterns. As two different coherence structures are built, two independent template coverings are performed; one considering all the patterns discovered in the different melodic representations of the piece and one only considering the rhythmic ones.

Every time a pattern is used on the template covering, the interest value of the patterns that remain in the sorted list must be recomputed. Their number of occurrences must be updated to consider only those that happen on the positions of the piece that are still uncovered. Once the interest value is recomputed for all the patterns in the list, it is resorted.

An example of the recomputing process is shown on Figure 7, where a stave of a melody of the corpus can be seen, and patterns P and P′ are highlighted. Pattern P occurs four times in a piece and has an interest value of 41.2. Pattern P′ is instantiated twice in the piece and has an interest value of 30.2, so P will have a higher position on the pattern list and would presumably be used on the covering of the segment shown on Figure 7. When the piece covering process begins, a higher interest pattern covers the positions where two of the occurrences of P happen (not shown in the figure), making its number of occurrences lower to two. The interest values of the patterns that remain in the list are recomputed and the interest of pattern P drops to 19.7, which is lower than the interest value of P′, that will be used on the covering of the segment.

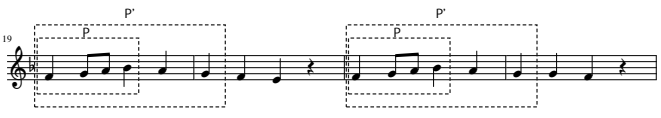


FIGURE 7: Example of the effect of computing the interest value of the patterns with the actual number of occurrences used on the covering. P represents the pattern that would be covered using the number of occurrences of the template piece, and P′ is the actual pattern used.

The covering of the piece ends when there is no free sequences in the piece that can be covered by any pattern

in the list.

Melodic coherence structure

In the building of the melodic coherence structure the patterns discovered in the representations of the four melodic viewpoints presented in Table 1 are used. Even though the patterns are discovered in different representations, all of them are gathered in a single pattern list that is used in the covering of the template piece in order to create the melodic coherence structure.

Rhythmic coherence structure

To build the rhythmic coherence structure, patterns are discovered within the dur representation of the template piece, where significant patterns are intended to be identified. It has been noticed that in many of the bertso melodies the most significant rhythmic patterns are long patterns, which can have other significant patterns within them. These so called nested patterns are also important when building the coherence structure of a piece, and have been considered in other music generation works [31]. In Figure 8 an example of a long pattern can be seen which has shorter patterns within it.

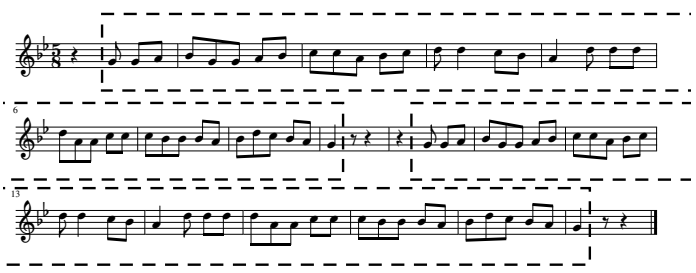


FIGURE 8: Score of the melody *Neure lagunak lagun zakidaz* where its most interesting pattern is highlighted.

To discover this kind of nested patterns, once the most significant patterns are selected to cover the template piece, the sorted list of all discovered patterns is processed again, to look for patterns that happen, twice or more, within those significant ones. This process is repeated until all the nested patterns in the piece are discovered, making it possible to nest pattern within other nested patterns, creating a pattern hierarchy.

Example

In Figure 9 the rhythmic structure of the piece *Neure lagunak lagun zakidaz* after the nested pattern discovery process is shown, where patterns of different nesting level are represented with different colours. It can be seen that the piece has one main rhythmic pattern that is repeated twice in the piece, and even if the whole piece is covered by this pattern, it does not provide enough information about how the rhythm is structured through the piece. Since the only information that this pattern represents is that the piece has a long segment that is repeated twice, the patterns that occur within that principal pattern are discovered and represented in the figure in purple.

The nested pattern discovery has also been applied in this purple pattern, in which another nested pattern of 7 elements has been discovered, represented in Figure 9 in green. In this example a pattern hierarchy of three levels is created.

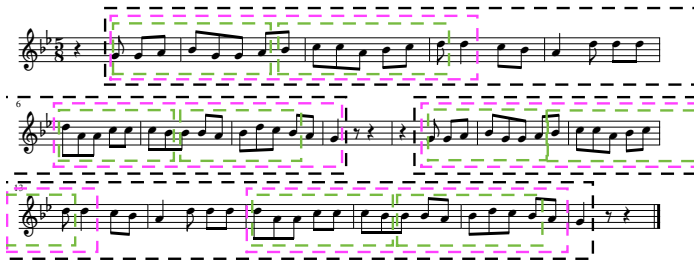


FIGURE 9: Nested pattern (in purple) found within the principal pattern (in black) in the melody *Neure lagunak lagun zakidaz*.

B. STATISTICAL MODELS

Once the coherence structures of the template piece are built, the new notes that will be sampled within them need to be chosen, in order to get new pieces that are stylistically similar to the ones in the bertso corpus. To sample these notes statistical models of the corpus are used, which assign probabilities to sequences of events, where high probability sequences are assumed to retain more aspects of the music style of the corpus than sequences with low probability. Since in this work a rhythmic generation process is presented in addition to the melodic generation, two different statistical models have been built; one to capture the melodic aspects of the corpus and one to capture the rhythmic ones. An *n-gram* model is used for both, for which a length and the feature it describes must be chosen, taking into account that different features are needed for each model.

1) Melodic model

In order to decide which melodic feature model fits the corpus best, models based on every melodic viewpoint presented in Section IV-A1 have been evaluated with leave-one-out cross validation. The cross-entropy of each model has been computed, where lower cross entropies are preferred. The cross-entropy of a model is defined as the negative base-2 logarithm of the product of the probabilities of all the 1934 pieces of the corpus.

To compute the probabilities of the pieces Equation 2 is used, which depends on the length of the n-gram. Letting $v_i = \tau(e_i|e_{i-1})$ be the viewpoint τ value of event e_i in the context of its preceding event e_{i-1}, the probability of a piece $\mathbf{e} = e_1, \ldots, e_\ell$ is computed as:

$$\mathbb{P}(\mathbf{e}) = \prod_{i=n}^{\ell} \mathbb{P}(v_i|v_{i-n+1}, \ldots, v_{i-1}) \times \mathbb{P}(e_i|v_i, e_{i-1}). \quad (2)$$

To elaborate, the product of all features in the sequence according to a n-gram model is represented by the first term. N-gram probabilities of the viewpoint τ are computed from the reduced corpus. The second term is the probability of the particular event given the feature, defined as a uniform distribution over events having the property v_i:

$$\mathbb{P}(e_i|v_i, e_{i-1}) = |\{x \in \xi : \tau(x|e_{i-1}) = v_i\}|^{-1},$$

where ξ is the set of possible pitches (see Table 1). Since this model can be applied for any viewpoint, it has been tried with different melodic viewpoints presented earlier in this work and different n-gram lengths, and the cross-entropies of the corpus represented with each viewpoint have been presented in Table 2. It can be seen that the interval viewpoint int has

Viewpoint	Unigram	Bigram	Trigram	Tetragram	Pentagram
3pc	4.63	4.53	4.45	4.35	4.26
pitch	3.72	2.87	2.62	2.68	3.05
int	3.17	2.74	2.55	2.56	2.81
intpc	5.19	4.06	3.83	3.7	3.7
5pc	3.65	3.47	3.38	3.29	3.22

TABLE 2: Cross-entropy of different viewpoints and different models, determined by leave-one-out cross validation on the corpus.

the lowest cross-entropy value in all the models, and how the value goes down when the length of the model increases, just until the trigram model is reached. After that the cross-entropy value starts going up. The lowest value indicates that a trigram int model fits the corpus best. Once the int trigram model is created Equation 3 is used to compute the probabilities of the generated event sequences,

$$\mathbb{P}(\mathbf{e}) = \prod_{i=3}^{\ell} \mathbb{P}(v_i|v_{i-2}, v_{i-1}). \quad (3)$$

since for this viewpoint $\mathbb{P}(e_i|v_i, e_{i-1}) = 1$.

2) Rhythmic model

As in the melodic model, to capture the rhythmic features of the corpus a trigram model has also been chosen. The same rhythmic viewpoint dur used in the viewpoint representation has been chosen to be used of the building of the rhythmic statistical model. To compute the probability of a rhythmic sequence Equation 3 is used.

C. GENERATION

The created statistical models are used to sample new high probability sequences into the semiotic structures, in order to generate pieces that retain more aspects of the pieces in the corpus. To do so, as two independent coherence structures and statistical models are created for the generation of new pieces, their melodic and rhythmic information are generated independently. For both generations a stochastic hill climbing optimization process has been used, which starts with a random sequence and iteratively changes random positions to improve its probability according to a statistical model, always respecting the coherence structures.

The vocabularies for both generations are fixed to ensure that the pitch and the duration of the generated notes are within a fixed range. The melodic vocabulary ξ'_m is defined by choosing the pitches from the tonality of the template piece, limiting them to the range of it, and which defines the admissible pitches for the generated pieces. The rhythmic vocabulary (ξ'_r) that is initially used in the generation is defined by taking the duration of the notes of the template piece. When the vocabularies are not big enough to create diverse generations, some entries can be added by hand, to generate more diverse pieces. The generation process starts with the rhythmic generation, and once it is finished the melodic generation step begins.

Rhythmic generation

When generating rhythmic sequences different constraints should be considered. As the generated pieces are bertso melodies, their note number and total duration should be conserved, in order to fit them into one of the accepted metrics and be able to use them to sing existing or new bertso lyrics. Taking that into account, when creating the initial random sequence for the stochastic hill climbing process, the rhythmic information that is contained within each pattern is randomized instead of sampling a random duration in each of its positions. Since rests act like phrase boundaries in many bertso melodies, their positions have been fixed, and they are never moved.

When randomizing patterns that contain nested patterns within them, first the patterns that are nested deeper in the hierarchy are treated, and the process then goes up until the patterns that are in the top level of the hierarchy are randomized. Every time a pattern is randomized all its other occurrences need also to be sampled, in order to conserve the rhythmic coherence of the piece defined in Section IV-A. When a segment of notes is not covered by any pattern of the coherence structure, it is treated like a one occurrence pseudo-pattern.

An example of randomizing patterns is shown in Figure 10, where a segment of the melody *Abiatu da bere bidean* is shown before and after the randomization. The segment is covered by a pattern that has a nested pattern repeated three times, highlighted in green. In the top stave the segment before the randomization is shown, while in the bottom stave the same pattern is shown after the rhythmic randomization and the random melodic sampling. The occurrences of the nested pattern cover almost all the notes in the main pattern, except the last three quarter notes and the rest, and the randomization can not have any effect in the duration of these last notes.

In each iteration of the optimization step a random position j of the piece is chosen, and the pattern that the position belongs to is identified. When j is part of a pattern that is nested within a larger one, the nested one is taken and a random element $r \in \xi'_r$ is chosen to subtract its value to the

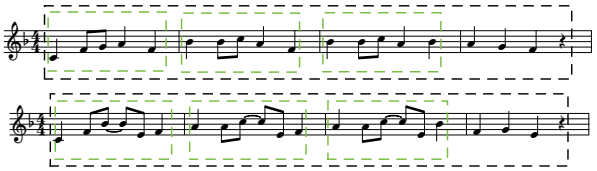


FIGURE 10: A rhythmic pattern of the melody *Abiatu da bere bidean* before (up) and after (bottom) the randomization step.

duration of j. The value of r is added to the duration of a position z randomly selected within the same pattern in the same pattern level. Once a pattern is updated, all its other occurrences are also updated, and the probability of the new sequence is computed with Equation 3. If the new probability is higher than the last saved one, the change is conserved. In Figure 11 the rhythmic pattern shown in Figure 10 can be seen after the optimization process, where it can be seen that the coherence of the nested patterns is conserved.

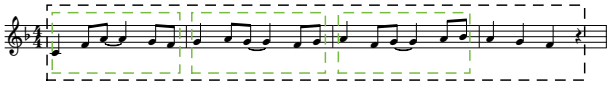


FIGURE 11: Rhythmic pattern of Figure 10 after the optimization phase.

In this step it is very important ensuring that no duration is set to 0 or negative values, so if this happened in an iteration of the optimization step, the change would not be valid. The rhythmic optimization process is run 10^6 times total.

Melodic generation

The initial random information of the melodic generation process is created with a left-to-right random walk, which must respect the coherence structure extracted from the template piece; a new note is sampled in every position of the template, and every time a complete pattern is instantiated, all of the future locations of the pattern are also instantiated. The piece is then iteratively modified: in each iteration of the process a random location i in the current piece $\mathbf{e}$ is chosen. A random element $e_i \in \xi'_m$ is substituted into that position, and the pattern to which that position belongs is identified, to also update all the other instances of the pattern, producing a new piece $\mathbf{e}'$. Thus the pieces generated at every iteration conserve the semiotic structure. The probability ($\mathbb{P}(\mathbf{e}')$) of the new piece is computed using Equation 3, and if it is higher than the last saved one, ($\mathbb{P}(\mathbf{e}') > \mathbb{P}(\mathbf{e})$), then the change is retained and piece $\mathbf{e}'$ is taken as the new current piece. This optimization process is iterated up to 10^4 times.

V. RESULTS

To illustrate the method, several melodies have been created using various pieces as template. Some examples of generated melodies using two different template pieces are shown below. The shown examples are high probability melodies according to the melodic and rhythmic models created from the corpus.

A. TEMPLATE *ABIATU DA BERE BIDEAN*

The first piece used as template piece is *Abiatu da bere bidean*. It is a piece with a length of 100 notes with a smooth melody and a pretty simple rhythm, and it has six phrases, all delimited by a rest.

The melodic structure that has been created from this template piece can be seen in Figure 12. Five patterns have been used in the covering of the piece, where four of them are pitch patterns and pattern E is an interval pattern. It can be seen that the second occurrence of pattern E has a rest in the middle, covering the end of a phrase and the beginning of the next one. This happens because no rhythmic information is used in the melodic coherence structure generation, and rests are not considered as events. Even if many melodic contour patterns have been identified in the piece, their interest value is not high enough to be included in the structure of this piece.

The rhythmic coherence structure of the template can be seen in Figure 13, where three rhythmic patterns are highlighted in black. A five element nested pattern has also been discovered within pattern A, which is repeated three times within each occurrence of A. All the events in the piece are covered by some rhythmic pattern except two rests that cannot be covered with any pattern, since the last rest in the score is not part of the viewpoint representation of the midi file.

In Figure 14 two pieces generated using this template can be seen. In both cases the generated melodies are smooth, with no big leaps. It can be seen that both melodies respect the melodic and rhythmic coherence structures extracted from the template piece.

Their melodic information is quite different, but it is noticeable that the rhythm is similar in both, even if it is not the same. Since in the original piece only quarter notes and quavers are used, the rhythmic vocabulary for the generations is limited. Even though semiquavers have been added by hand to the rhythmic vocabulary in order to have more options in the generation, they are not used in the shown generated pieces.

In the top piece of Figure 14 it can be seen that the interval int:E pattern has been sampled as an exact repetition, instead of a transposition. This is allowed in this approach of the method, since repetition is considered a particular case of a transposition.

B. EGUNTTO BATEZ NINDAGUELARIK

The second template used to illustrate the method is *Eguntto batez nindaguelarik*. It is a piece with 69 notes with no big leaps but a more diverse rhythm than the template *Abiatu da bere bidean*, and it has four phrases.

In Figure 15 the score of the melody can be seen where the melodic coherence structure that has been built for this template is highlighted. Only a pitch pattern has been chosen in the covering strategy, since, even if other melodic patterns

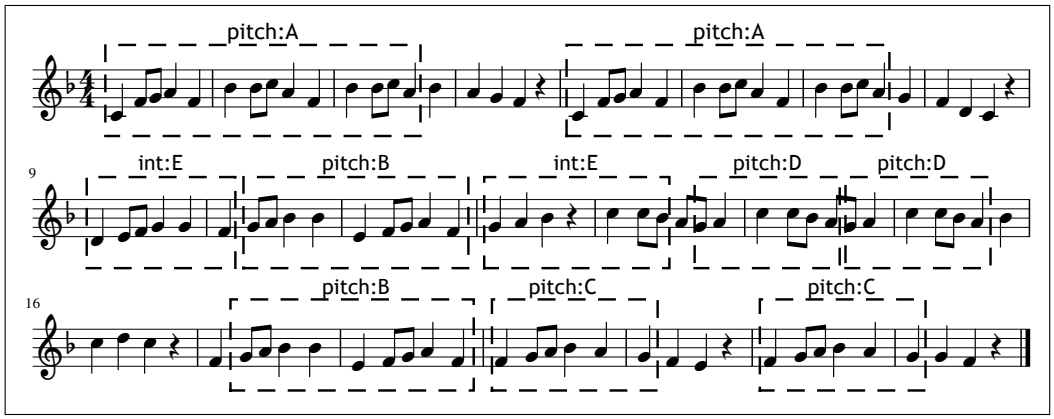


FIGURE 12: Score of the piece *Abiatu da bere bidean* with its melodic coherence structure highlighted. The type and label are shown above each pattern in the structure.

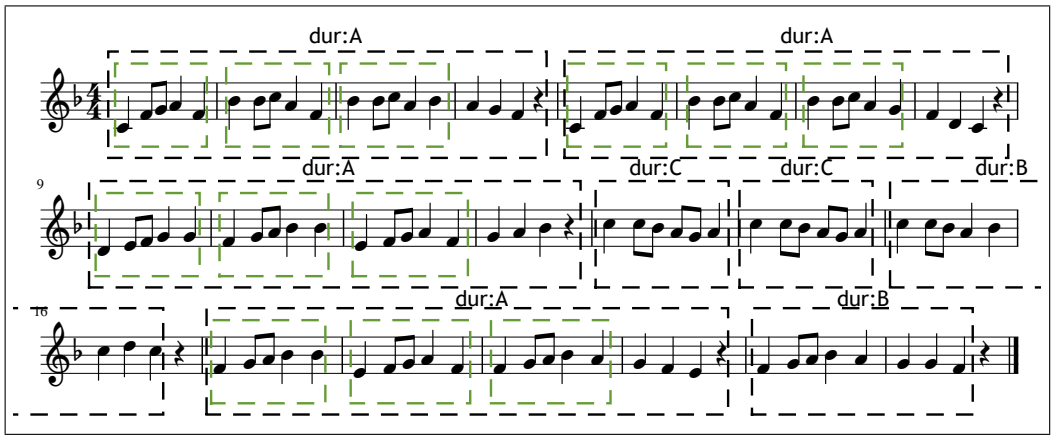


FIGURE 13: Score of the piece *Abiatu da bere bidean* with its rhythmic coherence structure highlighted. The label of each main pattern (in black) is shown, as well as the nested patterns (in green).

can be found that cover free spaces of the score, their interest is not high enough to be considered. Even though the melodic coherence structure is very simple in this template, it has been chosen because its rhythmic content is more interesting than the rhythm in template *Abiatu da bere bidean*.

The rhythmic coherence structure of the piece is shown in Figure 16, where four main patterns are highlighted. Duration pattern A has a nested pattern that happens twice in each occurrence of A. It can be seen that in this case not all of the notes are covered by a pattern in the rhythmic coherence structure, and as mentioned before, the segments that are not part of any pattern are treated like one occurrence patterns.

In Figure 17 the scores of two melodies generated with this template are shown. The first generated melody is pretty smooth with a nice rhythmic sequence which respects the rhythmic structure of Figure 16. The second generation, however, is an example of a melody that even if it has a high probability according to the statistical models (both melodic and rhythmic ones) has some rhythmic issues in the fourth and the 16th bars. The rhythm in these bars is not usual for bertso melodies and it would make them difficult to sing. Apart from these bars, both the rhythm and the melodic line in the piece are smooth and coherent. Since in this template more different measures are used compared to the first template, the generated pieces have more rhythmic differences between them.

C. DISCUSSION

The obtained results show that the method presented in this work can be used to generate new acceptable and coherent melodies. The rhythmic generation process that has been followed is conservative in order to generate pieces that maintain the original number of notes of each template, as well as its total duration, but in some cases it can be too conservative when the number of allowed rhythmic generations is low. This issue can be solved by allowing more duration values in the vocabulary of the piece or by using a model based on duration intervals instead of concrete duration values.

It also has been noticed that even if the statistical models assure that the generations will have certain melodic and rhythmic features, this is not always enough to guarantee that the generations will be musically pleasant. Since bertso melodies have no defined stylistic rules, rule sets cannot be used to evaluate the generations, but maybe some abstract

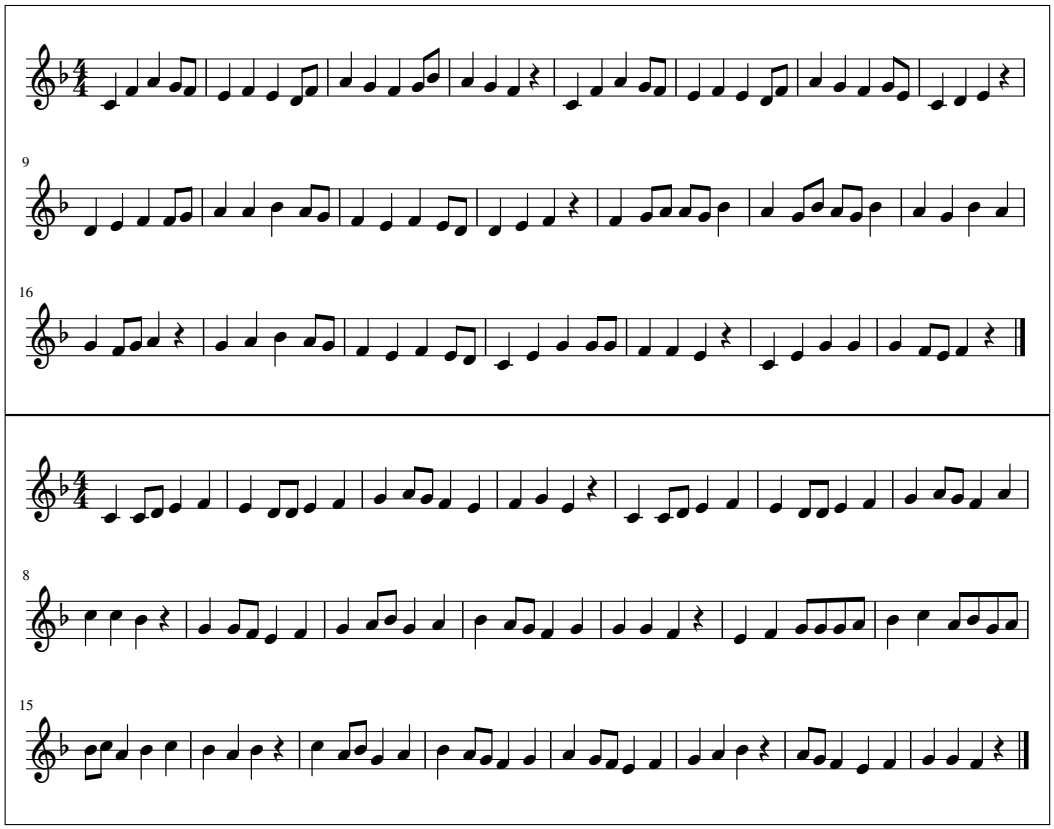


FIGURE 14: Examples of pieces generated from template *Abiatu da bere bidean*.

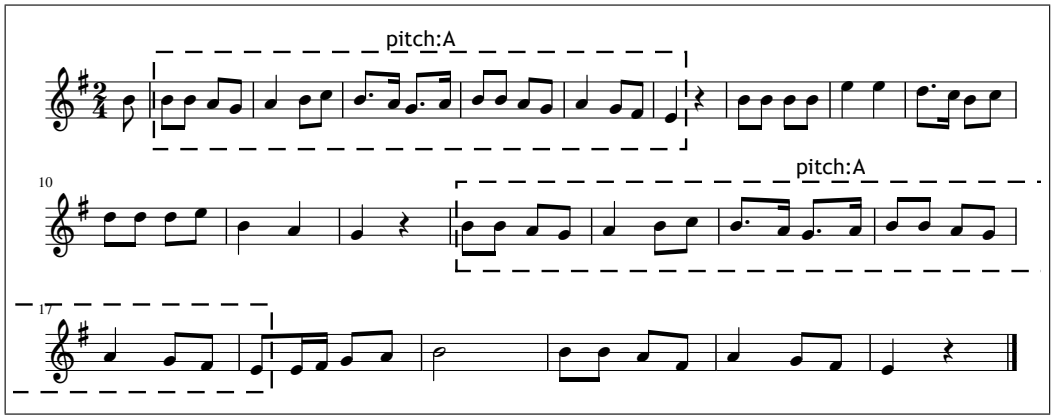


FIGURE 15: Score of the piece *Eguntto batez nindaguelarik* with its melodic coherence structure highlighted. The type and label are shown above each pattern in the structure.

rules could be learned from the corpus that would improve the musical quality of the generations.

VI. EVALUATION

To evaluate the results of the method presented in this paper an evaluation process has been carried out, for which five pieces have been randomly selected from the corpus, and they have been used as template to generate a new melody from each of them. Two sets of melodies were created from the ten initial pieces, making sure that a generated melody was never in the same set as the piece used as template to generate it. The scores in each melody set can be seen in Appendix A. Each melody was sung using some lyrics that fitted the metrics of the melodies downloaded from the website of the centre of documentation Xenpelar[3] and recorded. 31 professors and researchers from the Faculty of Informatics of the University of the Basque Country participated in the evaluation process, where one of the two sets of melodies was assigned to each participant.

The first step of the evaluation process was a training phase, in which the participants listened to the MIDI files

[3]https://bdb.bertsozale.eus/en/web/bertsoa/bilaketa

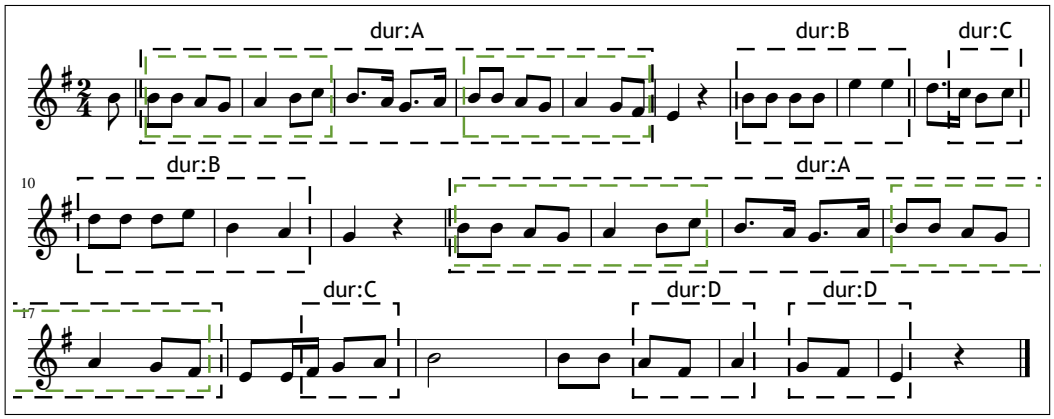


FIGURE 16: Score of the piece *Eguntto batez nindaguelarik* with its rhythmic coherence structure highlighted. The label of each main pattern (in black) is shown, as well as the nested patterns (in green).

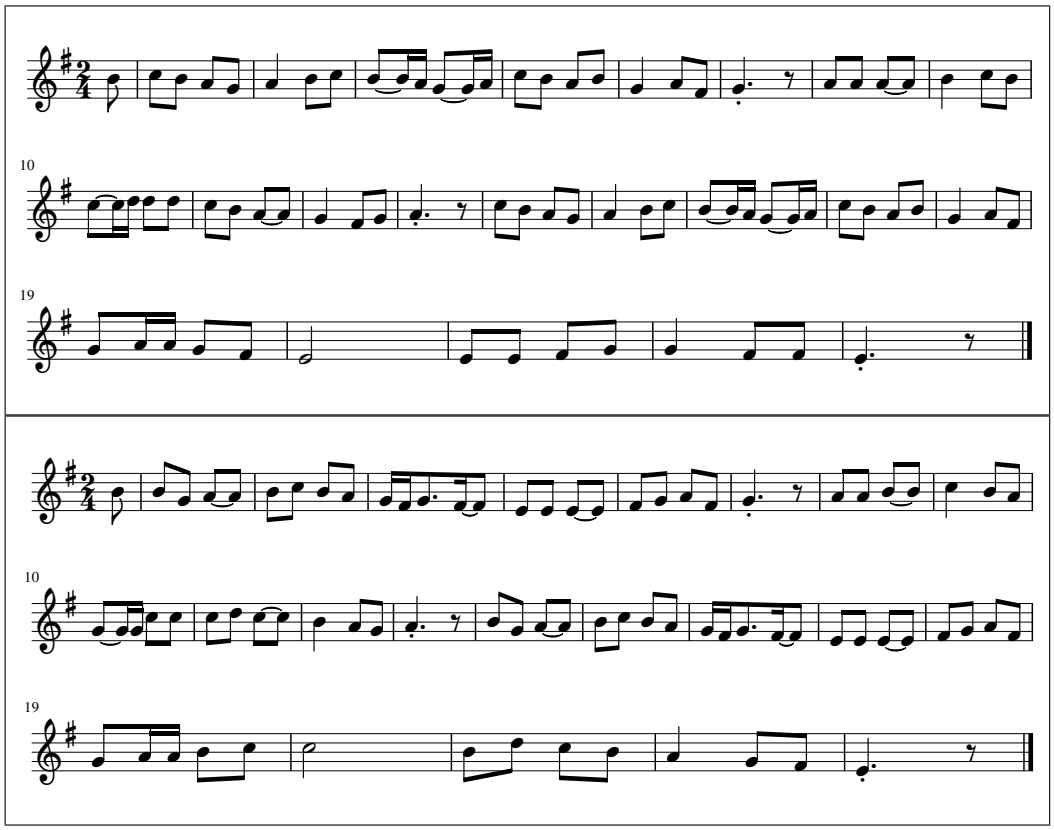


FIGURE 17: Examples of pieces generated from template *Eguntto batez nindaguelarik*.

of the melodies in their set once. After this process the participants were given a questionnaire in which they needed to mark if they thought that the recordings they would listen were original bertso melodies or generated ones.

In Table 3 an excerpt of the questionnaire given to the participants can be seen, where for a melody eight options are given. The first seven options indicate if the participants think that the melody is an original piece or a generated one, and in which degree they are certain of that. The eighth option has been added to mark the melodies that the participants knew, since it is possible that some of the melodies used as template are well known for some participants.

They listened to each of the five recordings of their set once to fill the questionnaire, and after finishing, they were asked to give a score from 1 to 10 to each of the generated melodies from their set. This evaluation was subjective and the participants were not given any more information about what they needed to take into account to evaluate the melodies. They just needed to evaluate them as they usually decide if they like a song or not. In Table 4 the obtained results are shown.The pieces generated with the presented method are shown in red, and the options that obtained more votes are

1: original 100%	2: quite sure original	3: I think original	4: I do not know	5: I think generated	6: quite sure generated	7: 100% generated
MELODY N:						
Original or generated?	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐ I know it

TABLE 3: Excerpt of the questionnaire that was given in the evaluation. Each participant needed to evaluate five melodies.

presented in bold.

The results of set A show that both $piece_3$ and $piece_5$ were perceived as generated, even if only two people were 100% sure. However, $piece_2$ was also perceived as a possibly generated piece even though it is an original bertso melody. The only clear result in this set is the one obtained with $piece_4$, which is a well known melody and 9 people knew it.

The results of set B show that $piece_1$ was perceived as possibly generated, even though, in the cases of $piece_2$ and $piece_4$ the results are not clear, and the responses are distributed between possibly original and possibly generated.

The evaluation of the generated pieces show that even if they obtain a mean score from 6.06 to 7.63, the scores are quite different depending on the listener as some of the standard deviation values show. These results show that even though the participants were able to identify some of the generated melodies as possibly generated, they were overall confused and not able to clearly distinguish between original and generated pieces, as the numbers in bold show.

VII. CONCLUSIONS AND FUTURE WORK

This work extends the automatic music generation paper by Goienetxea and Conklin [20], which presents a music generation method that is based on the use of the semiotic structure extracted from a template piece and a statistical model created from a corpus, in order to obtain new coherent melodies in the style of the corpus. In this extension of the method, besides the melodic information of the melody, the rhythmic content is also generated. To do so, an independent coherence structure of rhythmic segments, also extracted from the template piece, is used along with a new statistical model of the corpus, which tries to capture its rhythmic features. In the rhythmic structure a nested pattern discovery step has also been added, which allows the discovery of patterns with several levels of nesting. The generation method is applied in a corpus of basque bertso melodies, to get new melodies that can be used in bertso singing. An evaluation process has been carried out where some melodies generated with this method have been evaluated by 31 participants from the university. The obtained results show that even though some of the generated melodies were identified as generations, in general the participants were confused and not able to clearly distinguish between original and generated pieces. The scores obtained by the generated pieces also show that the participants overall liked these melodies.

In the presented method the melodic and the rhythmic information are treated independently; independent coherence structures are created to analyze the melodic line and the rhythm of the template piece, and two independent statistical models are also created. As mentioned before, it has been decided that it is the best way to deal with bertso melodies, considering that they can have segments with rhythmic repetitions that are not completely related melodically. However, the method presented herein is intended to be a general music generation method, and the melodic and rhythmic information can also be combined just linking melodic and rhythmic viewpoints. These linked viewpoints would be then used both for the semiotic analysis of the template piece and the building of the statistical model.

Thanks to the simplicity of viewpoints to describe events in various abstraction levels, new levels can be added in the melodic analysis or in the rhythmic one, or even to extract the structures from polyphonic pieces using viewpoints that describe vertical relations.

As future work different paths have been defined. The use of heterogeneous patterns is being considered; patterns that not only describe one type of segment relation, but patterns that can have different abstraction mixed. The need of nested pattern discovery within the melodic coherence structure should be studied.

.

APPENDIX A EVALUATION SCORES

ACKNOWLEDGMENT

This work has been partially supported by the Basque Government Research Teams grant (IT900-16) and the Spanish Ministry of Economy and Competitiveness RTI2018-093337-B-I00 grant.

REFERENCES

[1] D. Cope, Computers and Musical Style. Madison, WI, USA: A-R Editions, Inc., 1991.
[2] S. A. Hedges, "Dice music in the eighteenth century," Music & Letters, vol. 59, no. 2, pp. 180–187, 1978.
[3] M. Edwards, "Algorithmic Composition: Computational Thinking in Music," Communications of the ACM, vol. 54, no. 7, pp. 58–67, Jul. 2011.
[4] J. D. Fernández and F. Vico, "AI Methods in Algorithmic Composition: A Comprehensive Survey," Journal of Artificial Intelligence Research, vol. 48, no. 1, pp. 513–582, 2013.
[5] J. Sundberg and B. Lindblom, "Generative Theories in Language and Music Descriptions," Cognition, vol. 4, no. 1, pp. 99–122, 1976.
[6] N. G. Thomas, P. Pasquier, A. Eigenfeldt, and J. B. Maxwell, "A Methodology for the Comparison of Melodic Generation Models Using MetaMelo," in Proceedings of the 14th International Society for Music Information Retrieval Conference, ISMIR, Curitiba, Brazil, 2013, pp. 561–566.
[7] C. Thornton, "Generation of Folk Song Melodies using Bayes Transforms," Journal of New Music Research, vol. 40, no. 4, pp. 293–312, 2011.
[8] D. Conklin, "Multiple Viewpoint Systems for Music Classification," Journal of New Music Research, vol. 42, no. 1, pp. 19–26, 2013.

SET A:

Piece	1	2	3	4	5	6	7	I know it
$Piece_1$	2	5	3	1	4	0	0	1
$Piece_2$	0	2	2	**4**	**5**	2	0	0
$Piece_3$'	0	1	1	2	**3**	**6**	2	0
$Piece_4$	**12**	3	0	0	0	0	0	**9**
$Piece_5$'	0	0	2	**3**	**5**	**5**	0	0

SET B:

Piece	1	2	3	4	5	6	7	I know it
$Piece_1$'	0	1	0	0	**12**	**3**	0	0
$Piece_2$'	0	2	**4**	2	**6**	2	0	0
$Piece_3$	**3**	**4**	**7**	0	2	0	0	0
$Piece_4$'	0	**3**	**6**	0	**6**	1	0	0
$Piece_5$	3	**8**	**4**	0	1	0	0	1

Score of the generated pieces:

Piece	Mean score	Standard deviation
$Piece_1$'	6.06	1.06
$Piece_2$'	6.82	1.69
$Piece_3$'	6.47	1.55
$Piece_4$'	7.63	1.42
$Piece_5$'	6.33	2.12

TABLE 4: Results obtained in the evaluation of sets A and B. The pieces generated with the presented method are shown in red, and the options that obtained more votes are presented in bold.

[9] ——, "Music Generation from Statistical Models," in Proceedings of the AISB 2003 Symposium on Artificial Intelligence and Creativity in the Arts and Sciences, Aberystwyth, Wales, 2003, pp. 30–35.

[10] S. Dubnov, G. Assayag, O. Lartillot, and G. Bejerano, "Using machine-learning methods for musical style modeling," Computer, vol. 36, no. 10, pp. 73–80, Oct 2003.

[11] M. Allan and C. K. I. Williams, "Harmonising chorales by probabilistic inference," in Advances in Neural Information Processing Systems, 2004, pp. 25–32.

[12] L. M. Zbikowski, "Musical coherence, motive, and categorization," Music Perception, vol. 17, no. 1, pp. 5–42, 1999.

[13] A. Patel, Music, Language, and the Brain. Oxford University Press, USA, 2008.

[14] C. Anagnostopoulou, "Cohesion in Linguistic and Musical Discourse," in Proceedings of the 3rd European Society for the Cognitive Sciences of Music Conference. Uppsala, Sweden: ESCOM, 1997.

[15] J. Leach and J. Fitch, "Nature, Music, and Algorithmic Composition," Computer Music Journal, vol. 19, no. 2, pp. 23–33, Feb 1995.

[16] L. B. Meyer, "Meaning in Music and Information Theory," Journal of Aesthetics and Art Criticism, vol. 15, pp. 412–424, 1957.

[17] A. Schoenberg, P. Carpenter, and S. Neff, The Musical Idea and the Logic, Technique, and Art of Its Presentation. Indiana University Press, 2006.

[18] D. Cope, "An expert system for computer–assisted composition," Computer Music Journal, vol. 11, no. 4, pp. 30–46, 1987.

[19] F. Bimbot, E. Deruty, G. Sargent, and E. Vincent, "Semiotic structure labeling of music pieces: Concepts, methods and annotation conventions," in 13th International Society for Music Information Retrieval Conference (ISMIR), Porto, Portugal, Oct. 2012, pp. 235–240.

[20] I. Goienetxea and D. Conklin, "Melody transformation with semiotic patterns," in CMMR 2017: 13th International Symposium on Computer Music Multidisciplinary Research, Porto, Portugal, 2017, pp. 328–339.

[21] J. Garzia, A. Egaña, and J. Sarasua, The Art of Bertsolaritza: Improvised Basque Verse Singing. Donostia, Bertsozale Elkartea; Andoain, Bertsolari Liburuak, 2001.

[22] M. Chemillier, "Toward a formal study of jazz chord sequences generated by steedman's grammar," Soft Computing, vol. 8, no. 9, pp. 617–622, 2004.

[23] J. A. Biles, Improvizing with Genetic Algorithms: GenJam. London: Springer London, 2007, pp. 137–169.

[24] M. Scirea, J. Togelius, P. Eklund, and S. Risi, MetaCompose: A Compositional Evolutionary Music Composer. Cham: Springer International Publishing, 2016, pp. 202–217.

[25] F. P. Brooks, A. L. Hopkins Jr., P. G. Neumann, and W. V. Wright, "An experiment in musical composition," IRE Transactions on Electronic Computers, vol. EC–5, pp. 175–182, 1956.

[26] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription." in International Conference on Machine Learning, Edinburgh, Scotland, 2012.

[27] J. Briot, G. Hadjeres, and F. Pachet, "Deep learning techniques for music generation - A survey," CoRR, vol. abs/1709.01620, 2017.

[28] R. P. Whorley and D. Conklin, "Music generation from statistical models of harmony," Journal of New Music Research, vol. 45, no. 2, pp. 160–183, 2016.

[29] D. Herremans, K. Sörensen, and D. Conklin, "Sampling the extrema from statistical models of music with variable neighbourhood search." Athens, Greece: ICMC/SMC, 2014.

[30] V. Padilla and D. Conklin, "Statistical generation of two-voice florid counterpoint," in SMC 2016: 13th Sound and Music Computing Conference, Hamburg, 2016, pp. 380–387.

[31] T. Collins, R. Laney, A. Willis, and P. H. Garthwaite, "Developing and evaluating computational models of musical style," Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 2014.

[32] D. Meredith, K. Lemström, and G. A. Wiggins, "Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music," Journal of New Music Research, vol. 31, no. 4, pp. 321–345, 2002.

[33] J. Briot and F. Pachet, "Music generation by deep learning - challenges and directions," CoRR, vol. abs/1712.04371, 2017. [Online]. Available: http://arxiv.org/abs/1712.04371

[34] M. Bretan, G. Weinberg, and L. P. Heck, "A unit selection methodology for music generation using deep neural networks," CoRR, vol. abs/1612.03789, 2016. [Online]. Available: http://arxiv.org/abs/1612.03789

[35] L. Yang, S. Chou, and Y. Yang, "Midinet: A convolutional generative adversarial network for symbolic-domain music generation using 1d and

2d conditions," CoRR, vol. abs/1703.10847, 2017. [Online]. Available: http://arxiv.org/abs/1703.10847

[36] H. Dong, W. Hsiao, L. Yang, and Y. Yang, "Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," in Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, 2018, pp. 34–41. [Online]. Available: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17286

[37] J. Dorronsoro, Bertso Doinutegia. Euskal Herriko Bertsolari Elkartea, 1995.

[38] D. Conklin and I. H. Witten, "Multiple Viewpoint Systems for Music Prediction," Journal of New Music Research, vol. 24, pp. 51–73, 1995.

[39] D. Conklin and M. Bergeron, "Feature Set Patterns in Music," Computer Music Journal, vol. 32, no. 1, pp. 60–70, 2008.

[40] J. Ayres, J. Flannick, J. Gehrke, and T. Yiu, "Sequential PAttern Mining Using a Bitmap Representation," in Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ser. KDD '02. New York, NY, USA: ACM, 2002, pp. 429–435.

[41] D. Conklin, "Discovery of Distinctive Patterns in Music," Intelligent Data Analysis, vol. 14, no. 5, pp. 547–554, Oct. 2010.

[42] D. Conklin and S. Weisser, "Pattern and antipattern discovery in Ethiopian bagana songs," in Computational Music Analysis, D. Meredith, Ed. Springer, 2016, pp. 425–443.

[43] J. van Helden, B. André, and J. Collado-Vides, "Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies," Journal of Molecular Biology, vol. 281, no. 5, pp. 827 – 842, 1998.

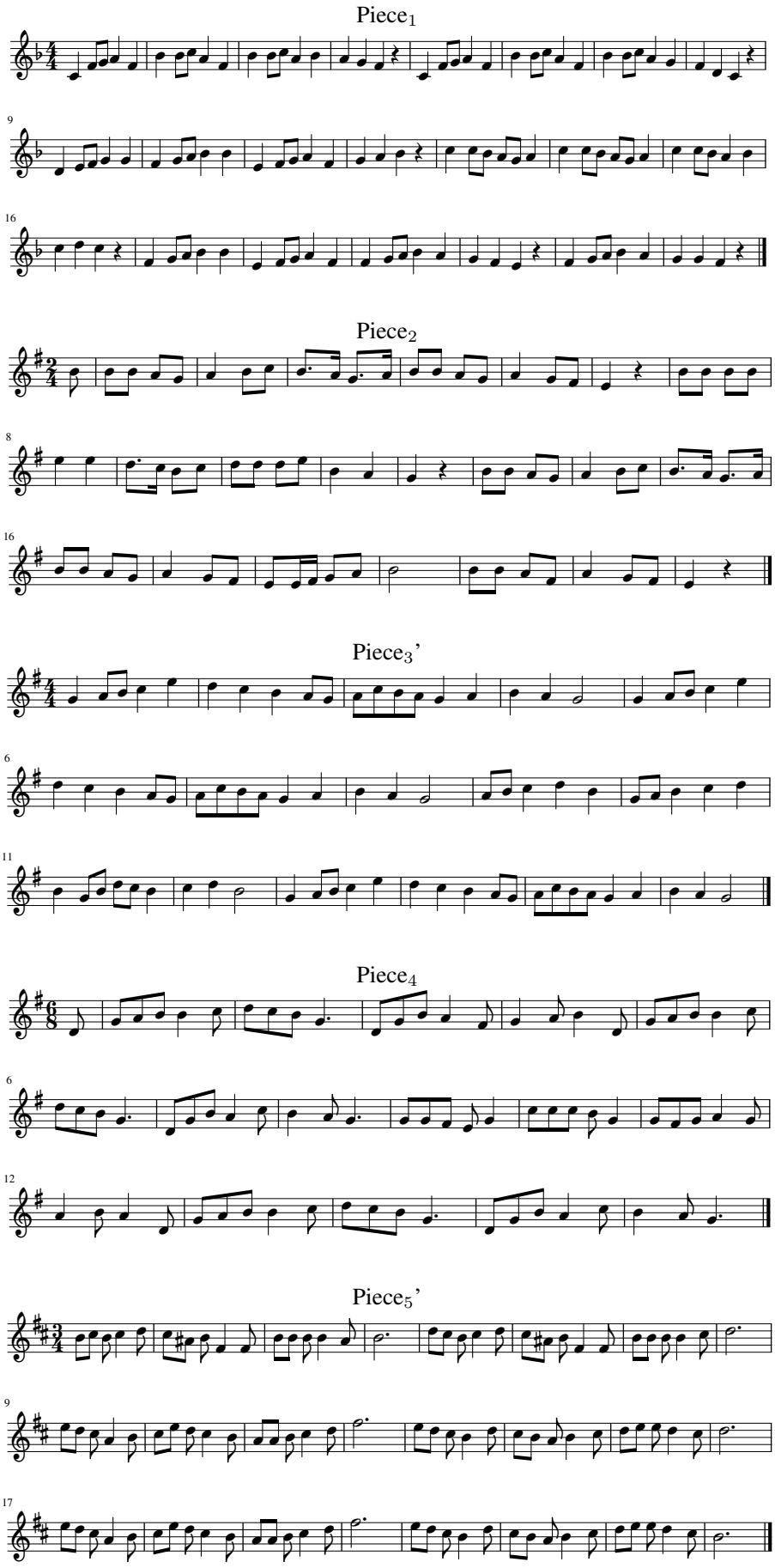


FIGURE 18: Scores of the pieces in set A

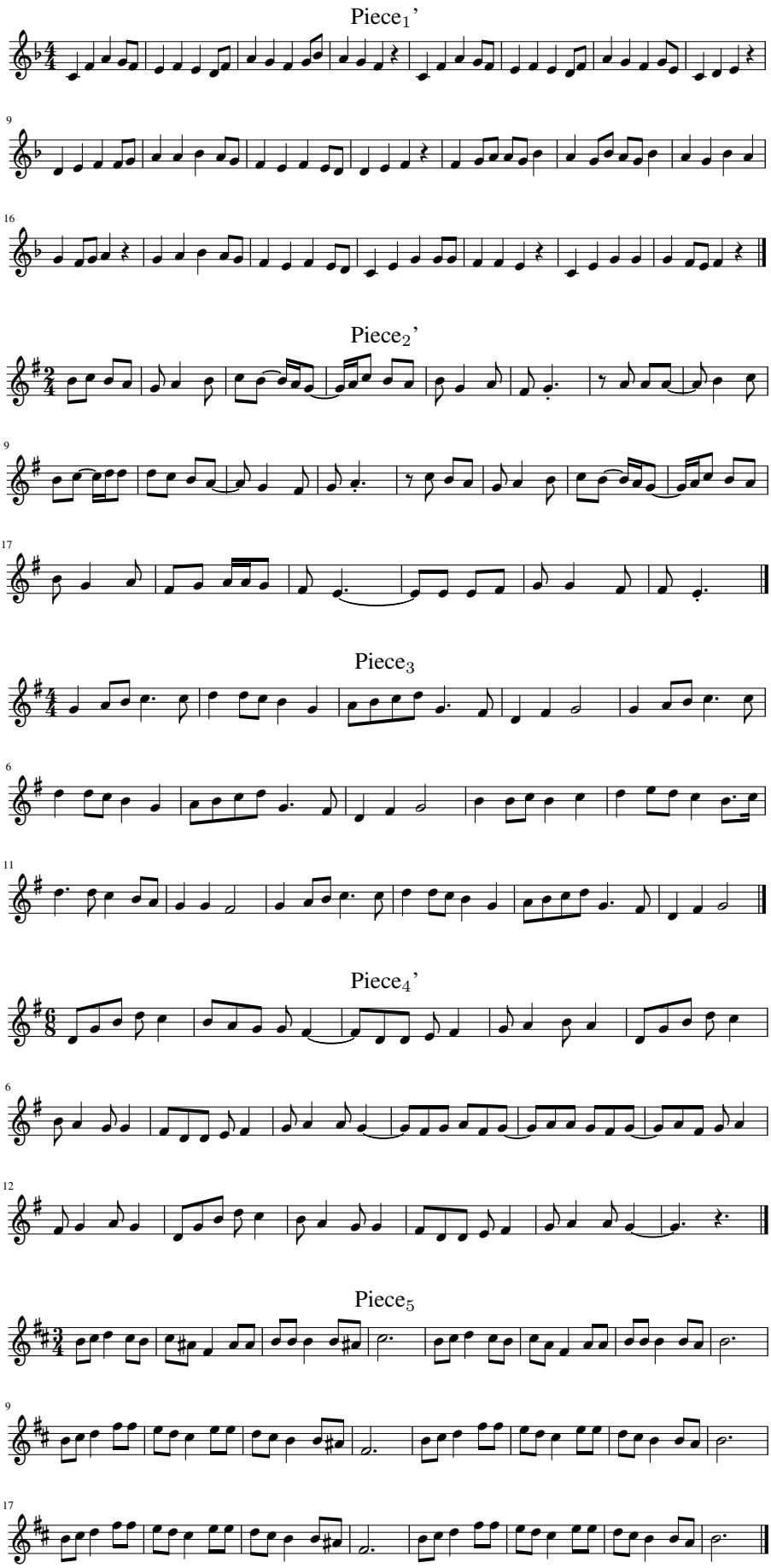


FIGURE 19: Scores of the pieces in set B

Part II

Article Related to Supervised Classification

9

Problems Selection Under Dynamic selection of the best base classifier in One versus One: PSEUDOVO

Authors: Izaro Goienetxea, Iñigo Mendialdua, Igor Rodríguez and Basilio Sierra.

Journal: Submitted to Applied Soft Computing

Submitted in February 2019

Year: 2019

Publisher: Elsevier

Problems Selection Under Dynamic selection of the best base classifier in One versus One: PSEUDOVO

Izaro Goienetxea[a], Iñigo Mendialdua[b], Igor Rodriguez[a], Basilio Sierra[a]

[a]*Department of Computer Science and Artificial Intelligence, University of the Basque Country UPV/EHU, San Sebastian, Spain*
[b]*Department of Computer Languages and Systems, University of the Basque Country UPV/EHU, San Sebastian, Spain*

Abstract

Class binarization techniques are used to decompose multi-class problems into several easier-to-solve binary sub-problems. One of the most popular binarization techniques is One versus One (OVO), which creates a sub-problem for each pair of classes of the original problem. Different versions of OVO have been developed to try to solve some of its problems, like DYNOVO, which dynamically tries to select the best classifier for each sub-problem.

In this paper a new extension that has been made to DYNOVO, named PSEUDOVO, is presented, which also tries to avoid the non competent sub-problems. An empirical study has been carried out over several UCI databases, as well as a new database of musical pieces of well known classical composers. Promising results have been obtained, from which can be concluded that the extension that PSEUDOVO made to DYNOVO improves its performance.

Keywords: Supervised classification, Decomposition strategies, One versus One, Dynamic Classifier Selection

1. Introduction

Supervised classification is a machine learning approach that consists in extracting knowledge from a set of input examples to classify new unlabeled examples in their correct class. It is applied in many different domains, and several strategies have been developed to tackle this classification problem. These strategies create a "general rule" or function based on a set of well-labeled examples, which predict class labels to new instances.

Even if the concept of classification is general for m-class problems, some of the existing strategies work much better in classification problems where they have to distinguish only between two classes. These are called binary classification problems, and a simple example of this type of classification problem would be a true/false problem. However, many real world problems are multi-class problems; the classifier needs to distinguish between more than two classes. In order to deal with multi-class classification, binarization techniques are widely used, which decompose the original multi-class problem into several easier to solve binary classification problems, faced by binary classifiers.

Class binarization process consists of two main steps; decomposition and combination. In the decomposition step the original problem is decomposed into several binary sub-problems. Even if different decomposition techniques can be found in the literature, there are three main strategies:

- One-vs-One (OVO)[1]: The problem is decomposed into a series of two-class problems, one problem for each pair of classes.
- One-vs-All (OVA)[2]: A sub-problem is created for each class, where the classifier learns to distinguish between the class and all the other classes.
- Error Correcting Output Codes (ECOC)[3]: The problem is decomposed into a larger number of binary sub-problems, where each classifier is trained on a two meta-class problem, where each meta-class consists of a combination of some of the original classes.

In the classification step each binary classifier returns a prediction, and different methods have been considered to combine them, such as majority vote or weighted vote.

Although class binarization techniques seem to be simple, they are effective dealing with multi-class problems, and they are applied in several real world classification problems. For example Liu et al. [4] proposed a novel combination of OVO and OVA to classify intentions based on the brain signals obtained with a near-infrared spectroscopy, and Zhou et al. [5] applied OVO for predicting listing status of companies in China. Class binarization techniques have also been applied in tasks like label ranking [6], bearing diagnostics [7], OCR [8] and natural language processing [9, 10].

Since in binarization techniques the original classification problem is decomposed into several sub-problems, different strategies have been developed

to treat those sub-problems independently in order to improve the classification accuracies. Some of the developed strategies are static; they try to select the best classifier for each sub-problem in a static way, while others select the best base classifiers dynamically, using Dynamic Classifier Selection methods.

Some works that have applied DCS techniques in OVO classification are DYNOVO [11], which selects the base classifier dynamically for each sub-problem of OVO, or the works proposed by Galar et al. [12] and Bagheri et al. [13], that use DCS to reduce the number of sub-problems.

In this paper we propose a new extension of our previous proposal called DYNOVO. While DYNOVO selects the base classifier in each sub-problem of OVO dynamically, in this new version the sub-problems that are going to take part in the final decision are also selected dynamically. The performance obtained by the presented method is compared to some versions of OVO, DYNOVO and the method presented by Galar et al. [12].

The rest of the paper is organized as follows: Section 2 explains how OVO binarization technique works, Section 3 gives an overview of Dynamic Classifier Selection techniques and their applications in real life. Section 4 reviews some of the works that have been developed on the field of binarization techniques, Section 5 describes PSEUDOVO, the method that is proposed in this paper. Section 6 presents the experiments that have been carried out to test the suitability of the method, and finally, in Section 7 some conclusions are presented.

2. Class Binarization

The class binarization technique chosen to be used in this work is the One-vs-One binarization, also called Pairwise Classification. This scheme divides a C class multi-class problem, $\theta_1 \dots \theta_C$, into $C(C-1)/2$ two-class sub-problems, where in each sub-problem only two classes are considered for classification, while the other classes are ignored.

To represent how the classes are grouped in each sub-problem the code matrix is used. Figure 1 shows the code matrix of a four class problem, where +1 and -1 represent the classes that are taken into account in a sub-problem and 0 represents a class that is not used in it. In the presented example, f_1 is a sub-problem in which the classifier is only trained to distinguish between classes θ_1 and θ_2, while classes θ_3 and θ_4 are ignored.

$$
\begin{array}{c} \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{array}
\begin{array}{c}
\begin{array}{cccccc} f_1 & f_2 & f_3 & f_4 & f_5 & f_6 \end{array} \\
\begin{pmatrix} +1 & +1 & +1 & 0 & 0 & 0 \\ -1 & 0 & 0 & +1 & +1 & 0 \\ 0 & -1 & 0 & -1 & 0 & +1 \\ 0 & 0 & -1 & 0 & -1 & -1 \end{pmatrix}
\end{array}
$$

Figure 1: OVO's code matrix for a four class classification problem.

When a new instance is being classified using this method all the sub-problems give a prediction of its class, and all these outputs are then combined. To do so, several strategies have been developed. The simplest one is the majority vote [1, 14], where each sub-problem returns a vote, and the class with the largest amount of votes is predicted. An extension to this method is the Weighted Voting, where the confidence of the classifiers of each sub-problem is used to assign a weight to each output [15]. On the other hand, the method by Hastie and Tibshirani [16] tries to estimate the best approximation of the posterior probability of the class, starting from the posterior probabilities of the pairwise sub-problems.

3. Dynamic Classifier Selection

Even if many different classifiers exist, not all of them are experts in classifying all the new samples they find; different classifiers usually make errors on different samples. In order to solve this issue methods to select different classifiers for different instances are used.

Dynamic Classifier Selection (DCS) methods try to select the best single classifier for each new instance to be classified, and they have been used in several real life problems. They are used in credit scoring problems [17] for example, where several DCS methods are used. On the other hand, Tsymbal et al. [18] proposed a dynamic ensemble method to deal with the antibiotic resistance problem.

Some of the most popular methods are Overall Local Accuracy (OLA) and Local Class Accuracy (LCA) presented by Woods [19]. The former obtains the accuracy of the classifier in local regions around the new instance. LCA is similar to OLA, and it also evaluates the competence level of each individual classifier in the local region, but the competence of each base classifier is calculated based on its local accuracy with respect to some output class.

Giacinto and Roli [20] also extended Woods's work proposing two new approaches: A Priori and A Posteriori. A Priori strategy considers the probability of correct classification of base classifiers in the local region, but instead of using only the labels assigned to each new sample, it considers the vector containing the posterior probabilities for each class. The second proposed approach, A Posteriori, is a mixture of LCA with A Priori.

DCS methods have been extended to Dynamic Ensemble Selection (DES), which select ensembles of the most suitable classifiers for each new sample. Some works like [21] show that these ensembles can be more accurate than single classifiers for certain classification tasks with techniques such as K-Nearest Oracles (KNORA). It explores the properties of the Oracle concept, which is an abstract model that always selects the classifier that will predict the correct label for the new sample if that classifier exists [22].

Dos Santos et al. [23] proposed a two-step DES method: in the first step, highly accurate candidate ensembles are generated; in the second step, for each test sample, the ensemble with the highest confidence level is selected among those generated ensembles. In a further work Cavalin et al. [24] extended the previous work and adapt it to Dynamic Multistage Organization (DMO) strategy.

A complete review on Dynamic Classifier Selection methods can be found in [25], where a more detailed description of the mentioned methods is presented, including their applications and some comparative studies.

3.1. Overall Local Accuracy (OLA)

In this work the DCS method OLA has been chosen to select the sub-problems of OVO and the base classifiers of each sub-problem. This technique is divided into two steps; validation and classification.

The aim of the validation step is to see which base classifier gives correct or incorrect predictions to each training instance. To do so each training instance is classified by different base classifiers and the results are recorded. These results are then used to compute the local accuracy in the classification step.

In the classification step, when a new instance arrives its local region is defined by obtaining its K nearest neighbors. The local accuracy of each base classifier for these neighbors is computed using the results recorded in the validation step and the classifier with the highest accuracy is selected to classify the new instance.

4. Related Works

Several works have been developed to compare the different binarization techniques. Galar et al. [26] compare different OVO and OVA versions, concluding that the best accuracies are achieved with OVO. García-Pedrajas and Ortiz-Boyer [27] also compared different binarization techniques, concluding that even if OVO and ECOC have a similar performance, OVO is the technique that achieves better performances when weak classifiers are used, due to its simplicity.

It should be mentioned that the OVO binarization technique has also some drawbacks:

1. Number of classifiers: As mentioned in Section 2, OVO creates a sub-problem for each pair of possible classes. Creating so many sub-problems may lead to having many classifiers that do not take into account the actual class of the instance, returning wrong predictions. On the other hand, the created sub-problems are simpler, hence, easier to build.
2. Weak classifiers: It is not easy choosing an optimal classifier for the dataset that is accurate in the classification of all the sub-problems.

Since these issues might have a negative effect in the classification accuracies, some approaches have been developed to try to solve them.

To tackle the issue of the high number of sub-problems that are produced in OVO different works have been developed. Some works proposed to combine OVA and OVO [28, 29]; firstly OVA was applied to obtain the two classes with the highest confidence level, then OVO was applied with these two classes. In this way they reduced the number of sub-problems to $C+1$. Other approaches proposed to dispose the sub-problems in a hierarchical structure such as graphs [30, 31, 32] or binary trees [33].

Other works try to solve the issue of the weak classifiers. Some works try to select the best base classifier in each sub-problem [34, 35] while others try to select the hyper-parameters of SVM classifier for each sub-problem of OVO. Many of these works use evolutionary algorithms, like Lebrun et al. [36] and Liepert [37] that use Genetic Algorithms or Souza et al. [38], who use Particle Swarm Optimization methods.

Several works try to solve the two previously mentioned drawbacks of OVO combining it with Dynamci Classifier Selection methods.

In the literature several works of this kind can be found. The aim of some of these works is to select the sub-problems that participate in the final

decision for each new instance dynamically. Other works intend to select the best base classifier in each sub-problem. Some of the existing works that apply these techniques are presented below.

4.1. Dynamic Sub-problem Selection in OVO

In order to reduce the number of sub-problems of OVO Galar et al. [12] and Bagheri et al. [13] proposed a new method, which intended to avoid the non-competent sub-problems. To do so, they applied Dynamic Classifier Selection methods: when an instance to be classified arrives, its K nearest neighbors are obtained, being the value of K three times the number of classes. Then OVO is applied considering only the classes in the neighborhood, while the remaining classes are ignored. In this way for each new instance the sub-problems that take part in the final decision are selected dynamically.

Galar et al. [39] updated their sub-problem selection to Distance-based Relative Competence Weighting combination (DRCW-OVO). This method weighs the competence of the outputs of the selected classifiers depending on the distance of the new sample to the nearest neighbors of each class in the problem.

4.2. Dynamic Classifier Selection in OVO

Mendialdua et al. [11] introduced a novel approach called DYNOVO that adapts the Dynamic Classifier Selection methods to OVO: when a new instance to be classified arrives, the multi-class problem is decomposed following OVO and for each sub-problem, independently, a Dynamic Classifier Selection method is applied to select the best base classifier.

Among the different Dynamic Classifier Selection methods they chose to use Overall Local Accuracy (OLA) and its extension Distance Weighted-Overall Local Accuracy (DW-OLA). Both methods work similarly: given an instance to be classified they select the classifier with the best behavior in the local region of the instance. To get the local region of the instance the well known K-Nearest Neighbor algorithm is used.

They considered that OLA selected better classifiers when the class distribution of the local region were well balanced. That is why they proposed to use K-NNE instead of K-NN to obtain the local regions. K-NNE [40] is a version of K-NN, where the K nearest neighbors of each class are obtained independently.

Recently Zhang et al. [41] proposed an extension of DYNOVO that combines OVO with an improved DES procedure, dynamically selecting a group of competent heterogeneous classifiers in each sub-problem for each query example.

5. Proposed approach: PSEUDOVO

In this section we present our proposal, which is an extension of our previous work called DYNOVO [11]. As mentioned before, DYNOVO tries to select the best base classifier for each sub-problem of OVO dynamically, using a Dynamic Classifier Selection method called Overall Local Accuracy (OLA). Following Galar et al. [12] or Bagheri et al. [13], we propose to use OLA also to reduce the number of sub-problems of OVO, avoiding the non-competent sub-problems. Hence we have named to the new approach PSEUDOVO, which stands for Problem SElection Under Dynamic One Versus One.

5.1. New method: PSEUDOVO

Our new approach is divided into two steps; for each test sample to be classified, firstly the sub-problems that take part in the final decision are selected and then the base classifier for each of those sub-problems is chosen.

5.1.1. Select the sub-problems

To select the sub-problems that will be taken into account in the classification of each test sample, the approach proposed by Galar et al. [12] is used. When an instance to be classified arrives, its local region is obtained using K-NN (being the value of K three times the number of classes). Then the sub-problems with the classes within the local region take part in the final decision while the sub-problems with a class that is not in the local region are ignored.

Figure 2 illustrates how the sub-problems are selected to classify 2 new samples (▲ and ●) in a 5-class problem. After their local regions are obtained selecting their 15 nearest neighbors, the sub-problems with class 5 are ignored to classify ▲ and the sub-problems with classes 2 and 4 are ignored to classify ● .

5.1.2. Select the base classifiers for each sub-problem

Once the sub-problems that take part in the final decision are chosen, it is time to select the base classifiers for each of those sub-problems. To

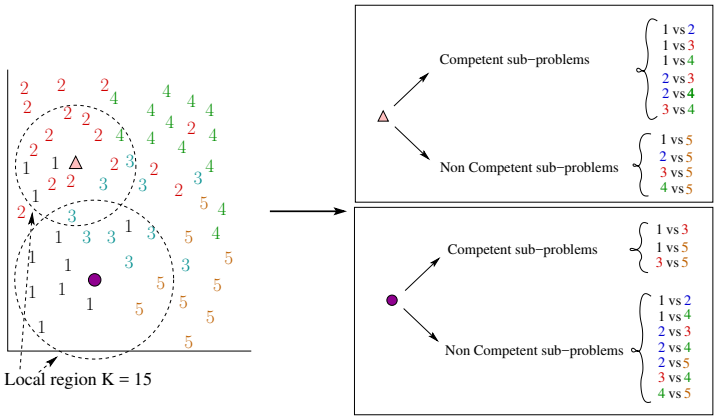


Figure 2: Example of how the competent sub-problems are selected for two new samples

do so, DYNOVO is applied. In each sub-problem, independently, Distance Weighted-Overall Local Accuracy (DW-OLA) is applied with a little change; instead of K-NN, K-NNE is used to get the local region. Then the local accuracy of each base classifier is calculated for the instances of the local region, where they receive weights depending on their distance to the test sample. Finally the base classifier with the highest local accuracy is used to build the classifier of the sub-problem.

Following with the previous example, Figure 3 illustrates how the base classifiers are selected in each sub-problem for candidate • . First the local region is obtained getting the 3 nearest neighbors of each class. Then the local accuracy of the different base classifiers is calculated. Finally the classifier with the highest local accuracy is selected. For example, in the sub-problem 1-vs-5 C_1 classifier is selected since it performs better in the closest instances of the test sample, in given that C_2 classifies correctly more instances of the local region.

6. Experiments

6.1. Datasets

To study the performance of the method presented in this work a set of 25 datasets has been selected from the UCI repository [42]. A summary of the used datasets can be found on Table 1, where the number of instances (#Cases), the number of attributes (#Attributes) and number of classes (#Classes) of each database has been presented.

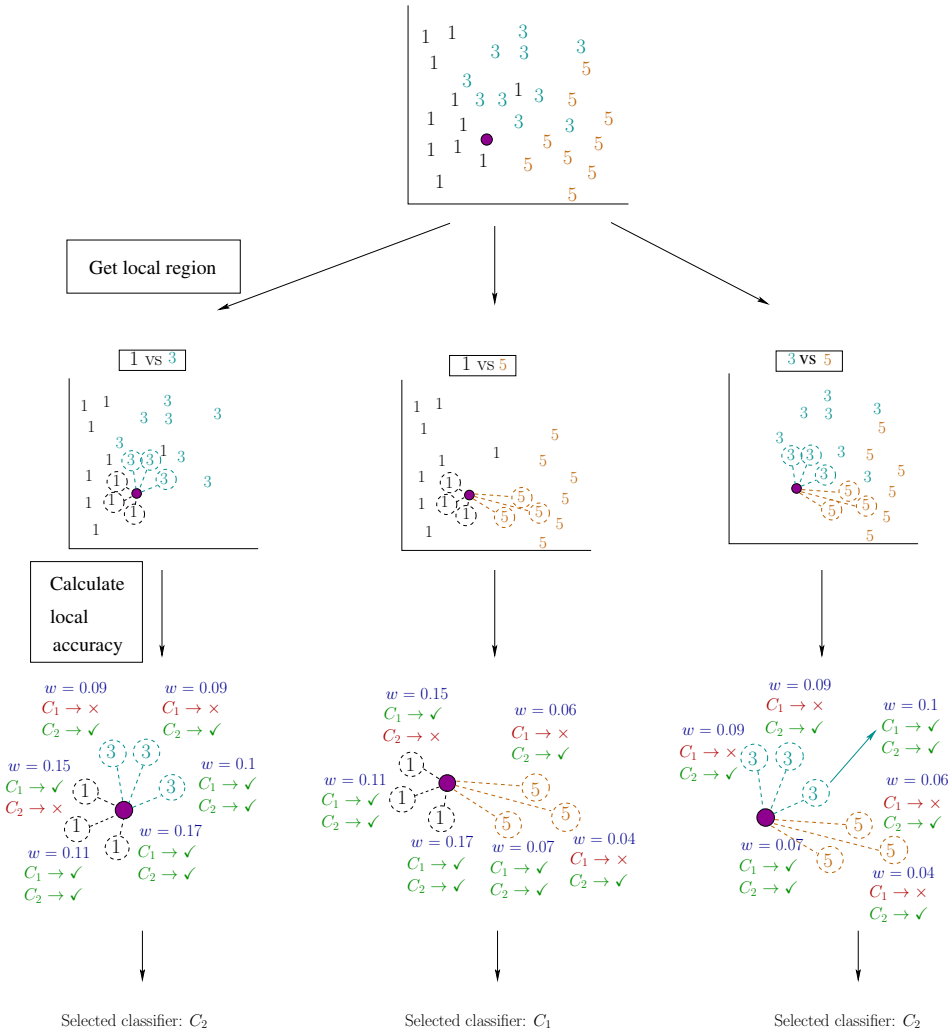


Figure 3: Example of how the base classifiers are selected for each sub-problem

A new database, *composers*, is also used, which has been created for automatic composer recognition and which is described below.

6.1.1. Composers

To test the performance of the proposed method, a new real life example has been introduced; a classical composer classification. To do so, a corpus with compositions of three well known classical composers, Bach,

Database	#Cases	#Attributes	#Clases
anneal	898	38	5
balance-scale	625	4	3
car	1728	6	4
cmc	1473	9	3
composers	1138	12	3
dermatology	366	33	6
ecoli	336	7	8
glass	214	9	7
iris	150	4	3
imgsegment	2310	19	7
lymph	148	18	4
nursery	12960	8	5
optdigits	5620	64	10
page-blocks	5473	10	5
pendigits	10992	16	10
satimg	6435	36	7
solar-flare1	323	12	6
solar-flare2	1066	12	6
soybean	683	35	19
vehicle	846	18	4
vowel	990	13	11
waveform-5000	5000	21	3
wine	178	13	3
winered	1599	10	6
yeast	1484	8	10
zoo	101	17	7

Table 1: Summary of the databases used in this work.

Beethoven and Haydn, has been compiled. The files included in the corpus have been downloaded from the KernScores website [43], which currently contains 7,866,496 notes in 108,703 files. It was developed to organize musical scores by the Center for Computer Assisted Research in the Humanities (CCARH), at Stanford University. A total number of 1138 instances have been collected, with a class distribution presented in Table 2. Even if the KernScores database contains more entries that the ones included in the composers collection, some of the pieces have been discarded because their length was no longer than a single bar or because they had some other problem.

Class	Instances
Bach	694
Beethoven	190
Haydn	254

Table 2: Number of melodies of each composer included in the database *composers*.

Files in standard MIDI format have been used to create the corpus, and a global feature set presented by Herremans et al. [44] for composer characterization has been chosen to represent the pieces. 12 melodic features have been extracted from each piece in the collection, which are presented in Table 3 along with a short description of each one. The jSymbolic feature extractor [45] has been used in the feature extraction step, which is able to extract up to 160 features from MIDI files.

Variable	Feature Description
x_1	Prevalence of most common pitch
x_2	Prevalence of most common pitch class
x_3	Relative prevalence of top pitches
x_4	Relative prevalence of top pitch classes
x_5	Prevalence of most common melodic interval
x_6	Relative prevalence of most common melodic intervals
x_7	Repeated notes
x_8	Chromatic motion
x_9	Stepwise motion
x_{10}	Melodic thirds
x_{11}	Melodic perfect fifths
x_{12}	Melodic octaves

Table 3: Global feature collection used in the composers database.

6.2. Base classifiers

Five different base classifiers from a software package for machine learning called Weka [46] have been used in the classification steps. Classifiers of diverse natures have been selected to give variability and reliability to the experimental phase. Following, the used base classifiers are described:

- J48 (C4.5 clone) [47]: Decision tree algorithm which makes a post-pruning phase, based on error based pruning algorithm.

- SMO (SVM clone) [48]: Kernel method that creates a hyperplane where the categories are divided by a clear gap that is as wide as possible.
- JRip (Ripper clone) [49]: Rule induction classifier which builds a rule-set by repeatedly adding rules to an empty rule-set until all positive examples are covered.
- Naive Bayes (NB) [50]: Statistical learning algorithm based on Bayesian rules. Given that the value of the class is known, it assumes independence between the occurrence of feature values to predict the class.
- Bayesian Network (BNet) [51]: Statistical learning algorithm which consists of a directed acyclic graph where random variables are represented as nodes and direct correlations between variables are represented as arcs.

6.3. Experimental setup

The classification accuracy of each binarization method is obtained by means of a stratified 10-fold cross-validation. Some of the methods also need a validation process that consists of a 5-fold cross-validation for each training fold.

In this work DCS methods are used in two phases of the process; the selection of sub-problems that participate in the final prediction and the dynamic selection of the base classifier for each sub-problem generated in OVO. The methods to select the base classifiers are based on K-NNE to define the local regions of the samples to be classified. Since the results may vary depending on the value of K, the methods have been tried with different values for K; 3, 6, 9, 12 and 15.

6.4. Obtained results

In this section the results obtained with different versions of OVO are presented in order to compare their performances.

The classification accuracies obtained with DYNOVO and PSEUDOVO for each database used in this work are shown in Table 4, where the results obtained with different K values are presented.

Table 5 presents the results obtained with both the DYNOVO and the new method and some of the state-of-the-art methods. A brief description of the compared methods is presented below.

- Best single (OVO-BS) [14]: Every classifier defined in Section 6.2 is used to classify each database using OVO decomposition. The results of the classifier with the highest accuracy are shown for each database.
- Static selection (OVO-ST) [52]: For each sub-problem the base classifier that obtains the best results in a validation process is selected.
- Galar et al. (GALAR) [12]: The K nearest neighbors of the test instance are found and OVO is applied, only considering those classes in the neighborhood. The value of K is established to 3 times the number of classes, and the result obtained with the best classifier is shown in each database.
- DYNOVO: The results obtained with DYNOVO and the K that gets the overall best accuracy are shown.
- PSEUDOVO: The results obtained with PSEUDOVO and the K that gets the overall best accuracy are shown.

DB	DYNOVO					PSEUDOVO				
	K = 3	K = 6	K = 9	K = 12	K = 15	K = 3	K = 6	K = 9	K = 12	K = 15
anneal	99.332	99.220	99.443	99.443	99.443	99.332	**99.332**	99.443	99.443	99.443
balance-scale	89.600	90.880	90.560	90.560	90.720	89.600	90.880	90.560	90.560	90.720
car	96.817	96.181	96.470	96.470	96.470	96.817	96.181	96.470	96.470	96.470
cmc	61.371	62.322	62.322	**63.069**	62.865	**61.439**	**62.390**	**62.390**	62.933	**62.933**
composers	86.819	88.137	88.576	89.104	88.576	**87.698**	**89.016**	**89.279**	**89.543**	**89.104**
dermatology	96.994	96.994	96.721	97.268	97.541	96.994	96.994	96.721	97.268	97.541
ecoli	87.798	88.095	89.583	89.583	90.179	87.798	88.095	89.583	89.583	90.179
glass	75.701	76.636	75.234	75.701	75.234	75.701	**77.103**	**75.701**	**76.168**	**75.701**
iris	96.000	95.333	95.333	95.333	95.333	96.000	95.333	95.333	95.333	95.333
imgsegment	**97.706**	**97.576**	**97.706**	**97.532**	**97.576**	97.619	97.446	97.576	97.489	97.532
lymph	87.162	87.838	86.486	87.162	86.486	**87.838**	**88.513**	**87.162**	**87.838**	**87.162**
nursery	**98.634**	**98.495**	**98.387**	**98.696**	**98.719**	98.619	98.449	98.341	98.650	98.696
optdigits	98.114	98.114	98.078	98.025	97.900	**98.363**	**98.416**	**98.381**	**98.345**	**98.221**
page-blocks	97.625	97.625	97.533	97.552	97.460	**97.643**	**97.789**	**97.716**	**97.734**	**97.734**
pendigits	98.381	98.344	98.372	98.472	98.463	**98.926**	**98.926**	**98.981**	**98.999**	**98.999**
satimg	89.899	89.806	89.930	90.070	90.023	**90.225**	**90.412**	**90.505**	**90.521**	**90.536**
solar-flare1	73.994	73.994	73.994	73.684	73.375	73.994	73.994	73.994	73.684	73.375
solar-flare2	76.735	77.486	77.486	76.923	**76.923**	76.735	77.486	77.486	76.923	76.829
soybean	**93.997**	**94.290**	**94.583**	**94.729**	**94.436**	93.851	94.143	94.436	94.583	94.290
vehicle	**81.087**	82.151	82.624	83.215	83.806	80.969	82.151	**82.742**	83.215	**83.924**
vowel	90.909	91.010	90.505	90.303	90.606	**91.010**	**91.414**	**90.707**	**90.707**	**90.707**
waveform-5000	84.460	84.280	84.500	85.000	85.200	**84.560**	**84.420**	**84.620**	**85.160**	**85.300**
wine	96.067	96.629	96.629	96.629	96.629	**96.629**	96.629	96.629	96.629	96.629
winered	65.854	67.917	68.730	69.543	69.669	**65.979**	**68.043**	**68.793**	69.543	**69.856**
yeast	64.151	65.970	**66.712**	67.318	67.588	**64.218**	**66.038**	66.644	**67.453**	**67.722**
zoo	97.030	97.030	97.030	97.030	97.030	97.030	97.030	97.030	97.030	97.030
Mean	87.778	88.167	88.213	88.401	88.394	**87.907**	**88.332**	**88.355**	**88.531**	**88.537**

Table 4: Classification accuracies of each database obtained with DYNOVO and PSEUDOVO for different values of K. The best results for each database are highlighted in bold.

As it can be seen in Table 4, PSEUDOVO outperforms DYNOVO in several datasets. Although, as it happens with classifiers when dealing with different datasets, the previous approach shows better performance in some of

the used datasets such as *composers, optdigits and satimg*, the new paradigm is better in mean than DYNOVO for all the considered K values. Existing differences are not remarkable, nevertheless, the new approach is better in mean, which allows us to infer that it is better at generalizing.

DB	OVO-BS	OVO-ST	GALAR	DYNOVO	PSEUDOVO
anneal	98.552	98.998	98.552	**99.443**	**99.443**
balance-scale	90.400	89.120	90.400	90.560	**90.720**
car	93.866	93.692	93.866	**96.470**	**96.470**
cmc	54.582	53.089	54.447	**63.069**	62.933
composers	85.589	85.677	86.292	**89.104**	**89.104**
dermatology	97.541	96.995	**98.361**	97.268	97.541
ecoli	86.607	85.417	86.905	89.583	**90.179**
glass	73.832	70.561	73.832	**75.701**	**75.701**
iris	**96.667**	**96.667**	**96.667**	95.333	95.333
imgsegment	97.186	97.186	97.013	**97.532**	**97.532**
lymph	**87.162**	86.486	**87.162**	**87.162**	**87.162**
nursery	97.238	97.824	97.130	**98.696**	**98.696**
optdigits	98.292	98.256	**98.523**	98.025	98.221
page-blocks	97.223	97.003	97.168	97.552	**97.734**
pendigits	98.026	98.426	98.299	98.472	**98.999**
satimg	88.283	88.454	88.361	90.070	**90.536**
solar-flare1	70.279	69.969	70.588	**73.684**	73.375
solar-flare2	75.516	75.141	75.516	**76.923**	76.829
soybean	94.583	93.997	93.704	**94.729**	94.290
vehicle	75.414	75.887	75.887	83.215	**83.924**
vowel	82.828	84.141	83.636	90.303	**90.707**
waveform-5000	86.700	86.680	**86.720**	85.000	85.300
wine	**98.876**	96.067	**98.876**	96.629	96.629
winered	61.789	58.974	60.788	69.543	**69.856**
yeast	59.434	58.895	59.771	67.318	**67.722**
zoo	96.040	95.050	96.040	**97.030**	**97.030**
Mean	86.250	85.717	86.327	88.401	**88.537**
Rank	3.46	4.15	3.29	2.19	**1.90**

Table 5: Classification accuracies obtained with different methods for each database. The best results are highlighted in bold.

As shown in Table 5, the new approach (PSEUDOVO) obtains the best result among the used paradigms in 17 out of 26 datasets, being as well the classifier which shows the best mean, and the best ranking value. Regarding

the differences, it is worth noticing that they are in general not very high, although some of them are remarkable. For instance, in the *wine* dataset, both PSEUDOVO and DYNOVO perform 2% worse than OVO and Galar; on the contrary, in *cmc*, *vehicle*, *winered* and *yeast* the new proposed approach obtains an accuracy 8% better than the previous approaches. In mean, PSEUDOVO outperforms all of the remaining paradigms, hence obtaining the best rank (1.90), followed by DYNOVO (2.19) and further from the other approaches. If the best result among all the paradigms is compared with each one individually, i.e., the maximum value is compared with the one obtained by a single approach, it is again the PSEUDOVO which obtains the best results, obtaining a mean difference of -0.26%, while the other approaches have -0.41% (DYNOVO) -2.48% (Galar) -2.56% (OVO-BS) and -3.09% (OVO-ST); in this sense it can be established that PSEUDOVO has proven to be the best in the performed experimental phase.

6.5. Statistical analysis

As we have several methods to compare, according to García et al. [53], we have used the Iman–Davenport test to detect statistical differences among the different algorithms. Then if the difference exists, we apply the Shaffer post hoc test in order to find out which algorithms are distinctive among them. We show the most relevant p-values obtained in the pairwise comparisons in tables, where "+" symbol implies that the first algorithm is statistically better than the confronting one, whereas "=" means that there are not significant differences between them.

The results of the statistical analysis reject the null hypothesis that all the methods are equivalent, because the p-value $(1 * 10^{-8})$ returned by the Iman–Davenport test is lower than our α-value (0.01). So we execute the Shaffer post-hoc test and the most relevant achieved p-values could be seen in Table 6. The results show that PSEUDOVO is the most robust strategy since it outperforms significantly OVO-ST, OVO-BS and GALAR. Also we may stress the good behaviour of DYNOVO since it outperforms OVO-ST.

6.6. Computational Load

In Table 7 we compare the mean number of sub-problems that each strategy needs. As it can be seen PSEUDOVO and GALAR are the strategies that need less sub-problems. Moreover, in databases such as *ecoli*, *imgsegment*, *optdigits*, *pendigits*, *satimg* and *soybean* PSEUDOVO and GALAR need 10 times less sub-problems than the remaining methods and it can be seen that

Hypothesis	p-value
PSEUDOVO vs OVO-ST	**+(2.9E-6)**
DYNOVO vs OVO-ST	**+(4.6E-5)**
PSEUDOVO vs OVO-BS	**+(0.0022)**
PSEUDOVO vs GALAR	**+(0.0096)**
DYNOVO vs OVO-BS	=(0.0228)
DYNOVO vs GALAR	=(0.0497)
GALAR vs OVO-ST	=(0.1938)
OVO-BS vs OVO-ST	=(0.3432)

Table 6: Shaffer test results

PSEUDOVO and GALAR use on average 3 or less sub-problems in 18 of the 26 databases. These facts are important since the less sub-problems are used, the more understandable is the final decision.

Finally, considering all the experiments, we can confirm the robustness of PSEUDOVO. On one hand, it outperforms the accuracy and reduces the computational complexity of the methods OVO-BS, OVO-ST and DYNOVO. On the other hand, it improves significantly the classification accuracy of GALAR which has the same complexity level.

6.7. Discussion

In order to observe the performance of the technique that we use to select the sub-problems (explained in 5.1.1) we have carried out a new experiment. In this experiment, we obtain the percentage of the instances that have an instance with the same class in its local region. That is, the percentage of the instances where its real class takes part in the final decision. The percentages are shown in Table 8.

Considering the results of Table 8 it can be observed that in most of the databases the technique considers the appropriate sub-problems for almost 100% of the instances. However, there are several datasets (*cmc, ecoli, glass, winered* and *yeast*) where in at least 1% of the cases the real class is not in the local region, hence, these cases are classified wrong. We consider most of these databases as "difficult" since viewing Table 5 their accuracies are low, only *ecoli* outperforms 76% of accuracy. However, we can see that PSEUDOVO outperforms DYNOVO on these datasets, hence, we conclude that this loss is assumable.

DB	OVO-BS	OVO-ST	GALAR	DYNOVO	PSEUDOVO
anneal	10	10	1.057	10	1.057
balance-scale	3	3	1.992	3	1.992
car	6	6	2.031	6	2.031
cmc	3	3	2.504	3	2.504
composers	3	3	1.322	3	1.322
dermatology	15	15	1.694	15	1.694
ecoli	28	28	2.619	28	2.619
glass	21	21	3.893	21	3.893
iris	3	3	1.000	3	1.000
imgsegment	21	21	1.351	21	1.351
lymph	6	6	1.182	6	1.182
nursery	10	10	2.609	10	2.609
optdigits	45	45	1.349	45	1.349
page-blocks	10	10	1.105	10	1.105
pendigits	45	45	1.097	45	1.097
satimg	21	21	1.329	21	1.329
solar-flare1	15	15	7.554	15	7.554
solar-flare2	15	15	6.172	15	6.172
soybean	171	171	16.507	171	16.507
vehicle	6	6	2.389	6	2.389
vowel	55	55	24.102	55	24.102
waveform-5000	3	3	1.040	3	1.040
wine	3	3	1.011	3	1.011
winered	15	15	3.348	15	3.348
yeast	45	45	7.925	45	7.925
zoo	21	21	5.881	21	5.881
mean	23.038	23.038	4.002	23.038	4.002

Table 7: Mean number of sub-problems used by different strategies

7. Conclusions

In this this paper a new version of DYNOVO called PSEUDOVO is presented, which in addition to the dynamic classifier selection that DYNOVO does for each sub-problem of the binarization step, also performs a dynamic sub-problem selection proposed by Galar et al. [12]. The presented method has two main steps; in the first one the sub-problems that presumably are

Database	#Cases	Local region with real class	Local region without real class	Percentage
anneal	898	897	1	99.89
balance-scale	625	621	4	99.36
car	1728	1728	0	100
cmc	1473	1427	46	96.88
composers	1138	1124	14	98.77
dermatology	366	366	0	100
ecoli	336	328	8	97.62
glass	214	211	3	98.6
iris	150	150	0	100
lymph	148	148	0	100
nursery	12960	12960	0	100
optdigits	5620	5619	1	99.98
page-blocks	5473	5434	39	99.29
pendigits	10992	10984	8	99.93
Satimg	6434	6411	23	99.64
segment	2310	2304	6	99.74
solar-flare1	323	321	2	99.38
solar-flare2	1066	1058	8	99.25
soybean	683	683	0	100
vehicle	846	840	6	99.29
vowel	990	990	0	100
waveform-5000	5000	4994	6	99.88
wine	178	178	0	100
winered	1599	1552	47	97.06
yeast	1484	1448	36	97.57
zoo	101	101	0	100
	$\sum = 63135$	$\sum = 62877$	$\sum = 258$	99.606

Table 8: Percentage of the cases where the real class is in the local region, given that the the number of cases in the local region is three times the number of classes.

competent in the classification of a new instance are selected, and in the second step the most competent classifier for each sub-problem is intended to be chosen.

The performance of the proposed method has been compared to some

OVO methods, DYNOVO and the sub-problem selection technique proposed by Galar et al. [12]. Several databases from the UCI repository have been used to test the accuracies of the tested methods, as well as a new database of features of musical pieces of some well known classical composers.

PSEUDOVO has shown to be a good optimization of DYNOVO taking into account the competitive results that have been obtained.

As future work real applications in which PSEUDOVO could be applied are to be searched; on the other hand, there are some other lines to investigate, such as the inclusion of new classifier algorithms which could adapt better to each considered two-class problem. Finally, the selection of the problems which would take part in the final decision can be tackled as a classification problem itself.

Acknowledgements

This work has been partially supported by the Basque Government Research Teams grant (IT900-16) and the Spanish Ministry of Economy and Competitiveness. TIN2015-64395-R (MINECO/FEDER).

References

[1] J. Fürnkranz, Round robin classification, J. Mach. Learn. Res. 2 (2002) 721–747.

[2] R. Anand, K. Mehrotra, C. K. Mohan, S. Ranka, Efficient classification for multiclass problems using modular neural networks, Trans. Neur. Netw. 6 (1995) 117–124.

[3] T. G. Dietterich, G. Bakiri, Solving multiclass learning problems via error-correcting output codes, J. Artif. Int. Res. 2 (1995) 263–286.

[4] H. Liu, W. Zheng, G. Sun, Y. Shi, Y. Leng, P. Lin, R. Wang, Y. Yang, J. feng Gao, H. Wang, K. Iramina, S. Ge, Action understanding based on a combination of one-versus-rest and one-versus-one multi-classification methods, in: 2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), pp. 1–5.

[5] L. Zhou, Q. Wang, H. Fujita, One versus one multi-class classification fusion using optimizing decision directed acyclic graph for predicting listing status of companies, Information Fusion 36 (2017) 80 – 89.

[6] E. Hüllermeier, J. Fürnkranz, W. Cheng, K. Brinker, Label ranking by learning pairwise preferences, Artificial Intelligence 172 (2008) 1897 – 1916.

[7] S. S. Y. Ng, P. W. Tse, K. L. Tsui, A one-versus-all class binarization strategy for bearing diagnostics of concurrent defects, Sensors (Basel, Switzerland) 14 (2014) 1295–1321.

[8] H. Deng, G. Stathopoulos, C. Y. Suen, Error-correcting output coding for the convolutional neural network for optical character recognition, in: 10th International Conference on Document Analysis and Recognition, ICDAR 2009, Barcelona, Spain, 26-29 July 2009, pp. 581–585.

[9] R. Ghani, Using error-correcting codes for text classification, in: Proc. 17th International Conf. on Machine Learning, Morgan Kaufmann, San Francisco, CA, 2000, pp. 303–310.

[10] T.-Y. Wang, H.-M. Chiang, One-against-one fuzzy support vector machine classifier: An approach to text categorization, Expert Systems with Applications 36 (2009) 10030 – 10034.

[11] I. Mendialdua, J. M. Martínez-Otzeta, I. Rodriguez-Rodriguez, T. Ruiz-Vazquez, B. Sierra, Dynamic selection of the best base classifier in one versus one, Knowledge-Based Systems 85 (2015) 298 – 306.

[12] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, Dynamic classifier selection for one-vs-one strategy: Avoiding non-competent classifiers, Pattern Recogn. 46 (2013) 3412–3424.

[13] M. A. Bagheri, Q. Gao, S. Escalera, Efficient pairwise classification using local cross off strategy, in: L. Kosseim, D. Inkpen (Eds.), Advances in Artificial Intelligence, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 25–36.

[14] J. H. Friedman, Another approach to polychotomous classification, Technical Report, Department of Statistics, Stanford University, 1996.

[15] E. Hüllermeier, S. Vanderlooy, Combining predictions in pairwise classification: An optimal adaptive voting strategy and its relation to weighted voting, Pattern Recognition 43 (2010) 128 – 142.

[16] T. Hastie, R. Tibshirani, Classification by pairwise coupling, in: Proceedings of the 1997 Conference on Advances in Neural Information Processing Systems 10, NIPS '97, MIT Press, Cambridge, MA, USA, 1998, pp. 507–513.

[17] H. Xiao, Z. Xiao, Y. Wang, Ensemble classification based on supervised clustering for credit scoring, Appl. Soft Comput. 43 (2016) 73–86.

[18] A. Tsymbal, M. Pechenizkiy, P. Cunningham, S. Puuronen, Dynamic integration of classifiers for handling concept drift, Inf. Fusion 9 (2008) 56–68.

[19] K. Woods, W. P. Kegelmeyer, Jr., K. Bowyer, Combination of multiple classifiers using local accuracy estimates, IEEE Trans. Pattern Anal. Mach. Intell. 19 (1997) 405–410.

[20] G. Giacinto, F. Roli, Methods for dynamic classifier selection, in: Proceedings 10th International Conference on Image Analysis and Processing, pp. 659–664.

[21] A. H. Ko, R. Sabourin, J. Alceu Souza Britto, From dynamic classifier selection to dynamic ensemble selection, Pattern Recognition 41 (2008) 1718 – 1731.

[22] L. I. Kuncheva, A theoretical study on six classifier fusion strategies, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (2002) 281–286.

[23] E. M. D. Santos, R. Sabourin, P. Maupin, A dynamic overproduce-and-choose strategy for the selection of classifier ensembles, Pattern Recognition 41 (2008) 2993 – 3009.

[24] P. R. Cavalin, R. Sabourin, C. Y. Suen, Dynamic selection approaches for multiple classifier systems, Neural Computing and Applications 22 (2013) 673–688.

[25] R. M. Cruz, R. Sabourin, G. D. Cavalcanti, Dynamic classifier selection: Recent advances and perspectives, Information Fusion 41 (2018) 195 – 216.

[26] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes, Pattern Recognition 44 (2011) 1761 – 1776.

[27] N. García-Pedrajas, D. Ortiz-Boyer, An empirical study of binary classifier fusion methods for multiclass classification, Information Fusion 12 (2011) 111 – 130.

[28] N. Garcia-Pedrajas, D. Ortiz-Boyer, Improving multiclass pattern recognition by the combination of two strategies, Pattern Analysis and Machine Intelligence, IEEE Transactions on 28 (2006) 1001–1006.

[29] J. Ko, H. Byun, Binary classifier fusion based on the basic decomposition methods, in: Proceedings of the 4th international conference on Multiple classifier systems, Springer, 2003, pp. 146–155.

[30] J. C. Platt, N. Cristianini, J. Shawe-Taylor, Large margin dags for multiclass classification, in: Advances in neural information processing systems, pp. 547–553.

[31] B. Kijsirikul, N. Ussivakul, Multiclass support vector machines using adaptive directed acyclic graph, in: Proceedings of the 2002 International Joint Conference on Neural Networks, volume 1, IEEE, pp. 980–985.

[32] I. Mendialdua, G. Echegaray, I. Rodriguez, E. Lazkano, B. Sierra, Undirected cyclic graph based multiclass pair-wise classifier: Classifier number reduction maintaining accuracy, Neurocomputing 171 (2016) 1576 – 1590.

[33] B. Fei, J. Liu, Binary tree of svm: a new fast multiclass training and classification algorithm, IEEE Transactions on Neural Networks 17 (2006) 696–704.

[34] G. Szepannek, B. Bischl, C. Weihs, On the combination of locally optimal pairwise classifiers, Engineering Applications of Artificial Intelligence 22 (2009) 79–85.

[35] A. Arruti, I. Mendialdua, B. Sierra, E. Lazkano, E. Jauregi, New one versus allone method: Nov@, Expert Systems with Applications 41 (2014) 6251 – 6260.

[36] G. Lebrun, O. Lezoray, C. Charrier, H. Cardot, An ea multi-model selection for svm multiclass schemes, in: Proceedings of the 9th International Work Conference on Artificial Neural Networks, IWANN'07, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 260–267.

[37] M. Liepert, Topological fields chunking for german with svm's: Optimizing svm-parameters with ga's, in: Proceedings of the International Conference on Recent Advances in Natural Language Processing.

[38] B. F. D. Souza, A. C. p. l. f. De Carvalho, R. Calvo, R. P. Ishii, Multiclass svm model selection using particle swarm optimization, in: 2006 Sixth International Conference on Hybrid Intelligent Systems (HIS'06), pp. 31–31.

[39] M. Galar, A. FernÃ¡ndez, E. Barrenechea, F. Herrera, Drcw-ovo: Distance-based relative competence weighting combination for one-vs-one strategy in multi-class problems, Pattern Recognition 48 (2015) 28 – 42.

[40] B. Sierra, E. Lazkano, I. Irigoien, E. Jauregi, I. Mendialdua, K nearest neighbor equality: Giving equal chance to all existing classes, Information Sciences 181 (2011) 5158 – 5168.

[41] Z.-L. Zhang, X.-G. Luo, S. García, J.-F. Tang, F. Herrera, Exploring the effectiveness of dynamic ensemble selection in the one-versus-one scheme, Knowledge-Based Systems 125 (2017) 53 – 63.

[42] D. Dheeru, E. Karra Taniskidou, UCI machine learning repository, 2017.

[43] C. S. Sapp, Online database of scores in the humdrum file format, in: ISMIR 2005, 6th International Conference on Music Information Retrieval, London, UK, 11-15 September 2005, Proceedings, pp. 664–665.

[44] D. Herremans, K. Sörensen, D. Martens, Classification and generation of composer-specific music using global feature models and variable neighborhood search, Computer Music Journal 39 (2015) 71–91.

[45] C. Mckay, I. Fujinaga, jsymbolic: A feature extractor for midi files, in: In Proceedings of the International Computer Music Conference, pp. 302–305.

[46] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, The weka data mining software: An update, SIGKDD Explorations Newsletter 11 (2009) 10–18.

[47] J. R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[48] J. C. Platt, Advances in kernel methods, MIT Press, Cambridge, MA, USA, 1999, pp. 185–208.

[49] W. W. Cohen, Fast effective rule induction, in: Proceedings of the Twelfth International Conference on International Conference on Machine Learning, ICML'95, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995, pp. 115–123.

[50] G. H. John, P. Langley, Estimating continuous distributions in bayesian classifiers, in: Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, UAI'95, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995, pp. 338–345.

[51] N. Friedman, D. Geiger, M. Goldszmidt, Bayesian network classifiers, Mach. Learn. 29 (1997) 131–163.

[52] G. Szepannek, B. Bischl, C. Weihs, On the combination of locally optimal pairwise classifiers, Engineering Applications of Artificial Intelligence 22 (2009) 79 – 85.

[53] S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power, Information Sciences 180 (2010) 2044–2064.

Part III

Articles Related to Music Classification

10

Towards the use of similarity distances to music genre classification: A comparative study

Authors: Izaro Goienetxea, Iñigo Mendialdua, Igor Rodríguez and Basilio Sierra.

Journal: PLOS ONE

Year: 2018

Publisher: PLOS

DOI: https://doi.org/10.1371/journal.pone.0191417

PLOS ONE

RESEARCH ARTICLE

Towards the use of similarity distances to music genre classification: A comparative study

Izaro Goienetxea[1]*, José María Martínez-Otzeta[1], Basilio Sierra[1], Iñigo Mendialdua[2]

1 Department of Computer Science and Artificial Intelligence, University of the Basque Country UPV/EHU, San Sebastián, Spain, **2** Department of Computer Languages and Systems, University of the Basque Country UPV/EHU, San Sebastián, Spain

* izaro.goienetxea@ehu.eus

OPEN ACCESS

Citation: Goienetxea I, Martínez-Otzeta JM, Sierra B, Mendialdua I (2018) Towards the use of similarity distances to music genre classification: A comparative study. PLoS ONE 13(2): e0191417. https://doi.org/10.1371/journal.pone.0191417

Editor: Enrique Hernandez-Lemus, Instituto Nacional de Medicina Genomica, MEXICO

Received: April 19, 2017

Accepted: January 4, 2018

Published: February 14, 2018

Data Availability Statement: All data used in this study are the property of Bertsozale Elkartea (http://www.bertsozale.eus/en?set_language=en) and are available online via their database webpage: http://bdb.bertsozale.eus/en/web/doinutegia/bilaketa.

Funding: This work was supported by IT900-16 Research Team from the Basque Government.

Competing interests: The authors have declared that no competing interests exist.

Abstract

Music genre classification is a challenging research concept, for which open questions remain regarding classification approach, music piece representation, distances between/within genres, and so on. In this paper an investigation on the classification of generated music pieces is performed, based on the idea that grouping close related known pieces in different sets –or clusters– and then generating in an automatic way a new song which is somehow "inspired" in each set, the new song would be more likely to be classified as belonging to the set which inspired it, based on the same distance used to separate the clusters. Different music pieces representations and distances among pieces are used; obtained results are promising, and indicate the appropriateness of the used approach even in a such a subjective area as music genre classification is.

Introduction

Automatic music classification is a topic that is getting more and more attention with the development of the multimedia technologies and the growth of available information. It is used in music genre classification, tune family identification or to classify tunes in geographical regions for example, and approaches that use both symbolic information and audio information have been developed [1, 2].

Music genre classification is an important task since genre is a descriptor that is usually used to organize large collections of music, specially in the Internet, where it is often used in search queries. Many different approaches have been developed to identify music genre in audio or symbolic representation, like Support Vector Machines [3, 4], similarity measures of symbolic representation [5], neural networks [6, 7] or deep learning methods [8].

Automatic music generation has interested people for centuries and many different algorithms have been developed since the first steps in automatic music composition, like knowledge based systems, evolutionary and other population-based methods, fractals or statistical models [9].

The developed methods for music generation can be classified in several categories, like stochastic methods, knowledge-based systems and artificial intelligence systems. Stochastic methods involve random variables and are the simplest to generate. Some early examples can be the *Musikalisches Würfelspiel* or musical dice games, like the one published in 1792 that was attributed to Mozart [10].

Knowledge-based systems use series of sets of rules or grammars to guide the composition of melodies, expanding high-level symbols into sequences of symbols [9]. These grammars can be learned from a corpus of a melodies or they can be invented.

Statistical models have been used in computational modelling of several musical style since they are able to capture some musical features that make it possible to generate new musical sequences that reflect an explicit musical style, and they can be learned from a corpus of melodies [11].

In order to use statistical models for coherent music generation the intra-opus problem needs to be considered: the generated piece must contain material that repeats through the piece. Almost all forms of music involve repetition [12], either of pitch sequences or at some more abstract levels, and that repetition gives a sense of meaning to music [13]. Musical cohesion is analyzed in [14], where music is compared to linguistic discourse, and it is concluded that music is composed by semantically related segments, which support the coherence of the piece. Describing the coherence of a piece is currently a scientific challenge, and different approaches have been developed, like the description of acoustic structure, functional structure or semiotic structure. Semiotic structure is the description of segments in a piece using a set of symbols, where each symbol represents a class of similar segments [15].

Music generation methods using a segmental structure extracted from an existing piece have been developed, to generate music in the "style" of the original piece, but with different melodic content, like the method developed by Collins et al [16]. This method discovers the repeated and transposed segment on a polyphonic piece and uses it to guide the generation of a new melody, which has different notes but the same coherence as the original piece.

This paper presents a folk melody classification method, which is based on similarity distances of symbolic representation of music, and which is combined with an automatic generation method. An unsupervised classification of a folk melody corpus is made and the discovered sets are used to generate new melodies, which are then classified into the discovered clusters.

The chosen corpus is a collection of *bertso* melodies. *Bertsolaritza* or *bertsolarism* is the art of singing improvised songs in Basque (bertsos), respecting various melodic and rhyming patterns. It is defined as a sung, rhymed and metered discourse by the book *The Art of Bertsolaritza: Improvised Basque Verse Singing* [17]. There is evidence of bertso singing and written bertso poem samples since the 15th century, and it is a very popular art nowadays in the Basque Country.

Bertsos are sung in many different occasions, like informal lunches with friends, homage ceremonies or competitions and any topic can occur in a bertso. Many bertsolari competitions take place every year in the Basque Country, and every four years the national championship final is held, with around 15000 people in attendance.

The main technical aspects of the bertso are the rhyme, meter and melody, which can be classified into traditional folk melodies (the great majority), modern melodies that coincide with one of the traditional metres and melodies that are specifically composed. Experts say the chosen melody for singing a bertso and the manner in which it is sung can be the key for the communicative success of the bertsolari, since the chosen melody must be able to combine with the created lyrics to transmit what the bertsolari wants to express with the bertso.

This paper is structured as follows; Section 'related work' overviews the work that has been done in music classification, Section 'proposed approach' describes the approach we propose, Sections 'experimental setup' and 'experimental results' present the experimental setup designed to test the method and the results obtained, and finally Section 'conclusions and future works' presents the conclusions that have been extracted from this work.

Related work

Several approaches have been used in the literature to deal with music classification for different tasks, like tune family identification or automatic music genre classification. The idea behind many of them is to obtain a representation of the analyzed music and afterwards build a model which would be able to classify the characteristics of the music treated on the approach, namely genre, structure, artist, composer, and so forth.

Automatic music genre classification is a task that has attracted the interest of the music community for more than two decades, and several similarity methods and machine learning techniques have been studied in the literature to deal with it. Kotsifakos et al. [5] deal with music genre classification for symbolic music, and specifically MIDI, by combining the recently proposed novel similarity measure for sequences, SMBGT, with the k-Nearest Neighbor (k-NN) classifier. For all MIDI songs they first extract all of their channels and then transform each channel into a sequence of 2D points, providing information for pitch and duration of their music notes.

Mendel and Ellis [4] present an approach based on support vector machines to classify songs based on global features.

Chai and Vercoe [18] worked on the classification of folk music pieces coming from different countries using monophonic melodies by means of hidden Markov models. In this paper the authors state that "This shows that melodies of folk music do carry some statistical features to distinguish them".

Bergstra, J et al. [19] present an algorithm based on ADABOOST that predicts musical genre and artist from an audio waveform.

Xu et al. [20] propose effective algorithms to automatically classify and summarize music content. Support vector machines are applied to classify music into pure music and vocal music by learning from training data. Based on calculated features, a clustering algorithm is applied to structure the music content.

Fu et al. [21] deal with music information retrieval (MIR), which addresses the problem of querying and retrieving certain types of music from large music data set.

Pinquier et al. [22] deal with a novel approach to speech/music segmentation. Three original features, entropy modulation, stationary segment duration and number of segments are extracted. They are merged with the classical 4Hz modulation energy.

Zhang et al. [8] propose the use of computational deep learning modules for extracting invariant and discriminative audio representations which can then be used to classify music in different genres.

Sturn [23] argue that an evaluation of system behaviour at the level of the music is required to usefully address the fundamental problems of music genre recognition (MGR), and indeed other tasks of music information retrieval, such as autotagging.

A challenging open question in music classification is which music representation (i.e., audio features) and which machine learning algorithm is appropriate for a specific music classification task. The goal is to find a set of linear mappings from several feature spaces to the semantic space spanned by the class indicator vectors [24]. Valverde-Rebaza et al. [25] present

a novel feature vector obtained directly from a description of the musical structure described in MIDI files for music representation.

Recently Febres and Jaffe [26] proposed a new music representation and classification system based on extracting the *Minimal Entropy Description* of polyphonic music files. The Minimal Entropy Description is the sequence of characters forming symbols for which the corresponding entropy is minimal, and this representation is used to compare computer files associated to a score, considering already available parameters such as their symbolic diversity and entropy.

In the work of Lee et al. [27] the bag of words (BoW) representation of modulation spectral analysis of spectral as well as cepstral features are constructed for music genre classification. This is an approach used as well in text classification [28] which can be improved by means of a Singular Value Decomposition approach [29].

Recent success with deep neural network architectures on large-scale datasets has inspired numerous studies in the machine learning community for various pattern recognition and classification tasks such as automatic speech recognition, natural language processing, audio classification and computer vision [30–32]. Music genre classification has been done as well; Rajann et al. [33] show that neural networks are comparable with classic learning models when the data is represented in a rich feature space. Chun and Hong [34] used a BP neural network (BPNN) music classification method.

In this paper, Basque Folk music is used to perform the experiments; Bassiou et al. dealed with Greek folk music genre classification [35]. Hillewaere et al. worked on automatic classification of dances using the *Dance-9* corpus [36].

Proposed approach

In this paper a three step method is presented to analyze a melody collection and create K clusters of similar melodies, use each of the clusters to generate 10 new pieces and classify each of the new generated pieces in one of the clusters. A schema of the process is shown on Fig 1.

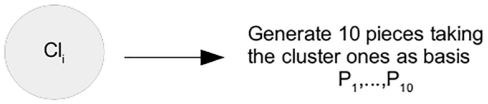
Corpus

In this work a collection of 100 bertso melodies of the corpus *Bertso doinutegia* is used. Bertso doinutegia is a collection of 2382 bertso melodies, created by Joanito Dorronsoro and published for the first time on 1995. It is updated every year by Xenpelar Dokumentazio Zentroa with new melodies that are used in bertso competitions and exhibitions. Entries in the collection are MIDI files which have a melody name, the name or type of the strophe, type of the melody (genre), creator, bertsolari who has used it, name and location of the person who has collected the melody, and year of the collection. Melodies have been manually classified in 17 genres according to their melodic features and the lyrics that are usually related to them.

To perform the classification task presented in this work, the melodies in the collection are represented using a viewpoint representation, presented in [37]. A viewpoint τ is a function that maps an event sequence $e_1, \ldots, e_\ell$ to a more abstract derived sequence $\tau(e_1), \ldots, \tau(e_\ell)$, comprising elements in the codomain of the function τ. Two viewpoints have been selected to represent the pieces in the corpus; pitch class interval (`intpc`) which computes the shortest distance in pitch class space between two unordered pitch classes (mod 12 interval), and five point contour (`5pc`) which represent the contour between two consecutive notes. A five point representation is used for contour, where `ld` and `lu` records whether a note is approached by a leap of three or more semitones (down or up), `sd` and `su` represent a step (smaller than three semitones) approximation and `eq` represents a unison. Fig 2 shows the viewpoint representation of the first two bars of the melody *Abiatu da bere bidean*, where the pitch class

1. Apply a clustering method on the corpus to identify K sets of similar songs.

2. For each cluster (set) i:



3. Classify each of the new pieces (10 * K) as belonging to one of the K clusters based on the chosen distance.

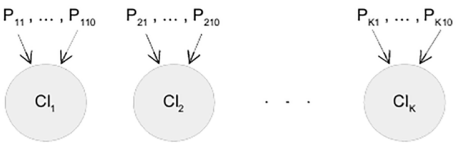


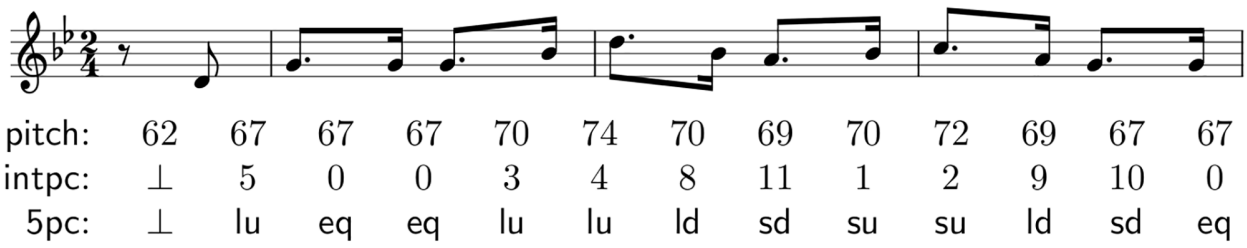
Fig 1. Method. Schema of the method presented in this work.

https://doi.org/10.1371/journal.pone.0191417.g001

interval and five point contour representations of the notes in the segment can be seen, along with their pitch numbers.

Matrices

In order to discover similarities between the different pieces in the corpus they are represented using matrices that capture their melodic information. Using the `intpc` and `5pc` viewpoints two matrix types are defined; interval matrices and contour matrices. Interval matrices are 12×12 matrices which count the number of transitions between all the interval pairs that occur in each melody. In order to build them the mod 12 interval between each contiguous note pair is computed. Then, the number of occurrences of each possible interval transition is computed. On the other hand, contour matrices are 5×5 matrices which count the number of transitions between all the contour pairs of each piece. To build the contour matrices the contour



pitch:	62	67	67	67	70	74	70	69	70	72	69	67	67
intpc:	⊥	5	0	0	3	4	8	11	1	2	9	10	0
5pc:	⊥	lu	eq	eq	lu	lu	ld	sd	su	su	ld	sd	eq

Fig 2. Viewpoint representation. Viewpoint representation of the first two bars of the melody *Abiatu da bere bidean*.

https://doi.org/10.1371/journal.pone.0191417.g002

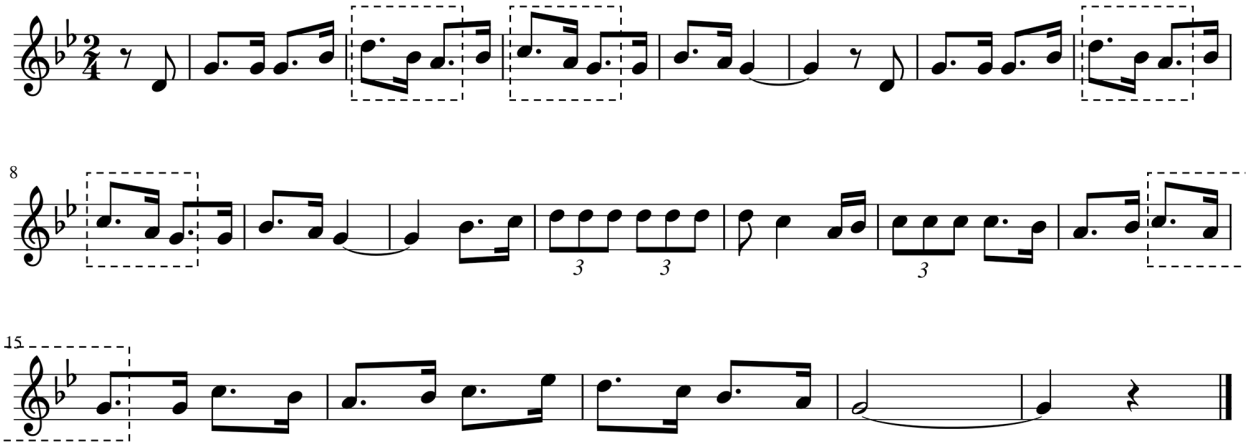


Fig 3. Example score. Score of the melody *Urruti nere menditik* included in the corpus. Contour sequences [ld,sd] are highlighted.

https://doi.org/10.1371/journal.pone.0191417.g003

transition between each pair of notes is computed and represented using the five point representation presented on Section 'corpus'. Then, the number of occurrences of each possible contour transition is computed. A contour matrix and an interval matrix are computed for each piece in the corpus. An example of a contour matrix and an interval matrix extracted from the piece in Fig 3 are shown in Figs 4 and 5.

To compute a position in the contour matrix, for example the [ld,sd], the number of times in the piece where a contour leap down (an interval larger than two semitones down) is followed by a contour step down (a step of one or two semitones down) is counted, which in this piece is 5. On Fig 3 these sequences have been highlighted to illustrate better where these sequences can be found on the example score shown.

Unsupervised classification

With the matrices obtained in the previous step, a method to group together similar songs has been developed through an unsupervised learning process.

$$
\begin{array}{c} contour \\ < -2 \\ -1 \\ 0 \\ 1 \\ > 2 \end{array}
\begin{array}{c}
\begin{array}{ccccc} < -2 & -1 & 0 & 1 & > 2 \end{array} \\
\begin{pmatrix}
0 & 5 & 0 & 1 & 1 \\
2 & 8 & 3 & 4 & 1 \\
0 & 2 & 9 & 0 & 5 \\
3 & 0 & 2 & 6 & 1 \\
2 & 4 & 2 & 1 & 2
\end{pmatrix}
\end{array}
$$

Fig 4. Contour matrix. Example of a contour matrix extracted from the piece *Urruti nere menditik*. Contours ld and lu represent a leap down or up of three or more semitones, contours sd and su represent a step down or up of one or two semitones and contour eq represents unison.

https://doi.org/10.1371/journal.pone.0191417.g004

interval	0	1	2	3	4	5	6	7	8	9	10	11
0	9	0	0	4	0	1	0	0	0	0	2	0
1	0	0	5	0	0	0	0	0	0	0	0	0
2	2	0	1	1	0	0	0	0	0	3	0	0
3	0	0	1	0	2	0	0	0	0	0	0	3
4	0	0	0	0	0	0	0	0	2	0	0	0
5	2	0	0	0	0	0	0	0	0	0	1	0
6	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	1	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	2
9	0	1	0	0	0	0	0	0	0	0	3	0
10	3	0	0	1	0	0	0	1	0	1	1	3
11	0	4	0	0	0	0	0	0	0	0	4	0

Fig 5. Interval matrix. Example of an interval matrix from the piece *Urruti nere menditik*.

https://doi.org/10.1371/journal.pone.0191417.g005

In order to discover relationships among the songs, an agglomerative hierarchical clustering algorithm has been used (Sequential Agglomerative Hierarchical Non-overlapping algorithm (SAHN)) [38]. This algorithm starts with a partition where each case is associated to a different cluster, therefore there are so many clusters as different cases. At each subsequent step the algorithm merges two clusters following certain optimization criteria, until all the data belongs to the same cluster. The output of the algorithm is a hierarchy along with the merging steps. Then, if a partition with N clusters is wanted, it is necessary to traverse the hierarchy until the right cutting point is found. The criteria to merge two clusters in the building phase is the complete linkage method, where the distance between two clusters is the maximum distance between their individual components.

In Fig 6 is shown an example of a dendrogram showing the clusters created after applying the SAHN method to the set of numbers {1,2,6,10,11,30,31,33,36,38,45,46,50}. As it can be seen from the figure, sets of numbers that are very close to each other according to the complete linkage method are grouped together lower in the hierarchy, while the sets that are father apart are grouped in the top. If we are interested in the partition with a given number of clusters, it is necessary to check the level of the dendrogram where such partition is created. For example, the red vertical line of Fig 6 shows the level of the dendrogram where a partition of four clusters is created. These clusters are {10,11}, {1,2,6}, {45,46,50} and {30,31,33,36,38}.

In the research described in this paper matrices representations are used, and therefore suitable distances between matrices are needed. Several distances have been tested. These distances are the following ones:

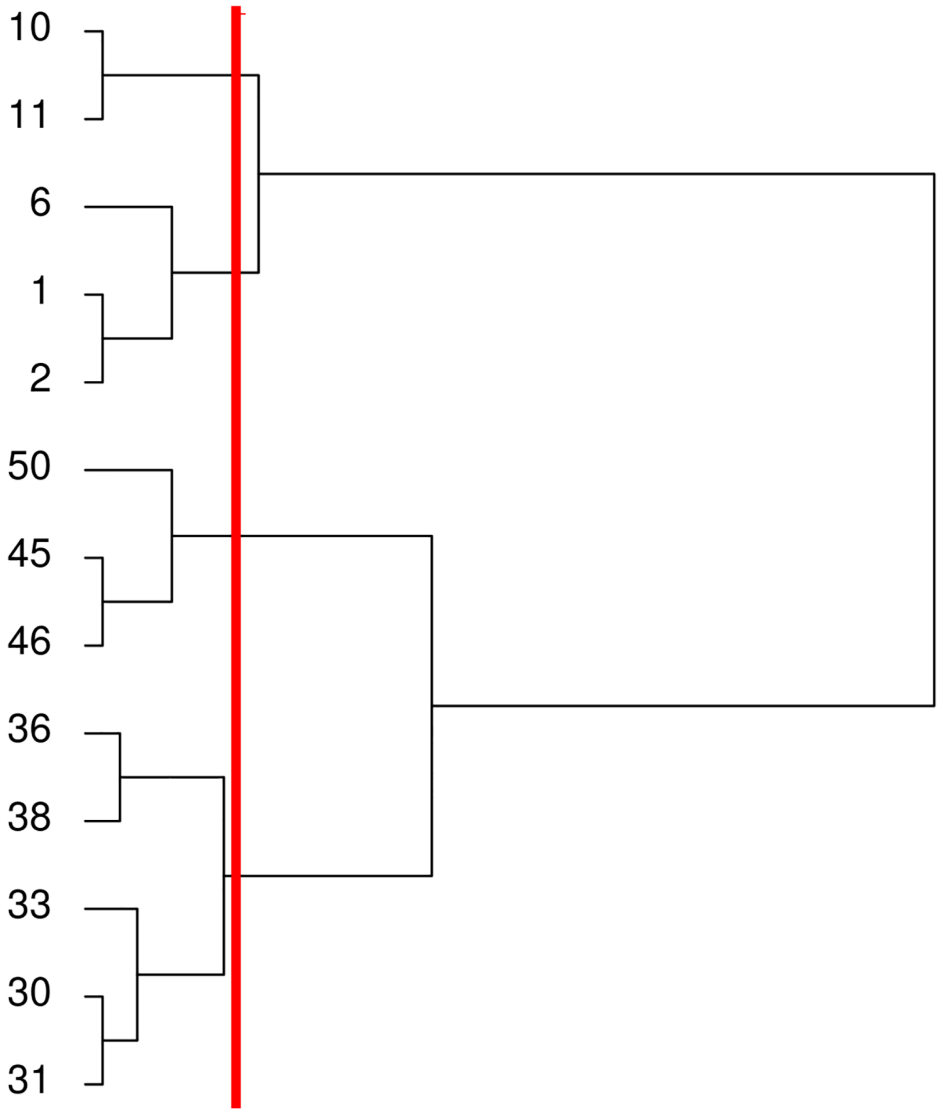


Fig 6. Example dendrogram. Example of a dendrogram created by the SAHN method.

https://doi.org/10.1371/journal.pone.0191417.g006

- The distances induced by the following norms: 1-norm, ∞-norm, Frobenius norm, maximum modulus norm.
- Kullback-Leibler and Jeffrey divergences.
- Earth mover's, Manhattan and Intersect distances.

In the following paragraphs we will explain them briefly:

1-norm. The 1-norm is computed as the maximum of the sums of the absolute values of the elements of each column. For an M-by-N matrix A, its value is

$$\max_{1\leq j\leq N}\sum_{i=1}^{M}|a_{ij}|.$$

∞-norm. The ∞-norm is computed as the maximum of the sums of the absolute values of the elements of each row. For an M-by-N matrix A, its value is

$$\max_{1\leq i\leq M}\sum_{j=1}^{N}|a_{ij}|.$$

Frobenius norm. The Frobenius norm (F-norm) of a matrix, sometimes also called the Euclidean norm, is computed as the square root of the sum of the absolute squares of its elements. For an M-by-N matrix A, its value is

$$\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}|a_{ij}|^2}.$$

Maximum modulus norm. The maximum modulus norm (M-norm) of a matrix is computed as the maximum of the absolute values of its elements. For an M-by-N matrix A, its value is

$$\max_{1\leq i,j\leq M,N}|a_{ij}|.$$

Kullback-Leibler divergence. The Kullback-Leibler divergence (KL) can be interpreted as the number of additional bits needed to encode instances coming from a distribution $p(x)$ if coded according with another distribution $q(x)$. For two M-by-N matrices A and B interpreted as distributions over a two-dimensional grid, its value is

$$\sum_{1\leq i,j\leq M,N}a_{ij}\log\frac{a_{ij}}{b_{ij}}.$$

Jeffrey divergence. The Jeffrey divergence is a measure that tries to address one of the problems of the Kullback-Leibler divergence, the lack of symmetry. It is defined as

$$D_{KL}(A,B)+D_{KL}(B,A).$$

Earth mover's distance. The earth mover's distance (EMD) is a distance between two probability distributions. The name comes from its intuitive interpretation: if the probability distributions are modelled as amounts of material over a surface, the EMD distance is the cost of moving the amounts from one disposition to another. For two M-by-N matrices A and B interpreted as distributions over a two-dimensional grid, its value is

$$\frac{\sum_{1\leq i,j\leq M,N}\sum_{1\leq k,l\leq M,N}f_{ijkl}d_{ijkl}}{\sum_{1\leq i,j\leq M,N}\sum_{1\leq k,l\leq M,N}f_{ijkl}}.$$

where f_{ijkl} is the flow between a_{ij} and b_{kl} that minimizes the total cost, with d_{ijkl} the distance between the elements a_{ij} and b_{kl}.

Manhattan distance. The Manhattan distance between two M-by-N matrices A and B is defined as

$$\sum_{i=1}^{M}\sum_{j=1}^{N}|a_{ij}-b_{ij}|.$$

Intersect distance. The Intersection distance between two M-by-N matrices A and B is defined as

$$\sum_{i=1}^{M}\sum_{j=1}^{N}\min(a_{ij},b_{ij}).$$

These distances or norms are all used in our work; interested readers could refer to [39] to have a better view and further knowledge about distances and their use in Machine Learning.

After applying the SAHN algorithm with the previous matrices distances to the pieces in the corpus, several clusters partitions are created. Those clusters partitions are used to generate new melodies that are intended to be similar to the original pieces.

Music generation

To generate new melodies a music generation method based on statistical models and a coherence structure is used. The coherence structure of a piece describes which segments are related on a piece, where the relations between segments can be exact repetitions or transpositions. Transposed segments are segments that have the same interval sequence, but different notes. A coherence structure is extracted from a template piece and is then used to guide the generation process in order to get new coherent melodies. As a result of the process pieces that have the same coherence structure of the template, but different melodic content, are created.

Coherence structure. In order to extract the coherence structure of a melody a manual or automatic segmentation is performed to identify the segments that are related through the piece. Many related segments may exist within a piece, but the most meaningful ones are retained, manually creating a structure of segments that do not overlap. The extracted structure is then used as a guide on the generation of new musical information, which segments in the new melody must be repeated or transposed.

Fig 7 shows a segmentation for one of the pieces used as templates in the generation, where several segments have been highlighted. Segments A, B, D and E are repetition segments, they occur twice unaltered within the piece, and segment C is a transposition segment.

In the generation process the defined coherence structure will be used as a constraint, to assure that the generated melodies respect the coherence of the template piece.

Statistical models. A statistical model is built from each of the clusters identified in the previous step of the presented method. Once it is built, it is used to measure the probabilities of the generated melodies, using the single viewpoint model described in [40] and presented in the equation below. Letting $v_i = \tau(e_i|v_i, e_{i-1})$ be the feature τ of event e_i in the context of its preceding event e_{i-1}, the probability of the piece is computed as:

$$P(e)=\prod_{i=1}^{\ell}P(v_i)\times P(e_i|v_i,e_{i-1}).$$
$$P(e_i|v_i,e_{i-1})=|\{x:\tau(x|e_{i-1})=v_i\}|^{-1}. \quad (1)$$

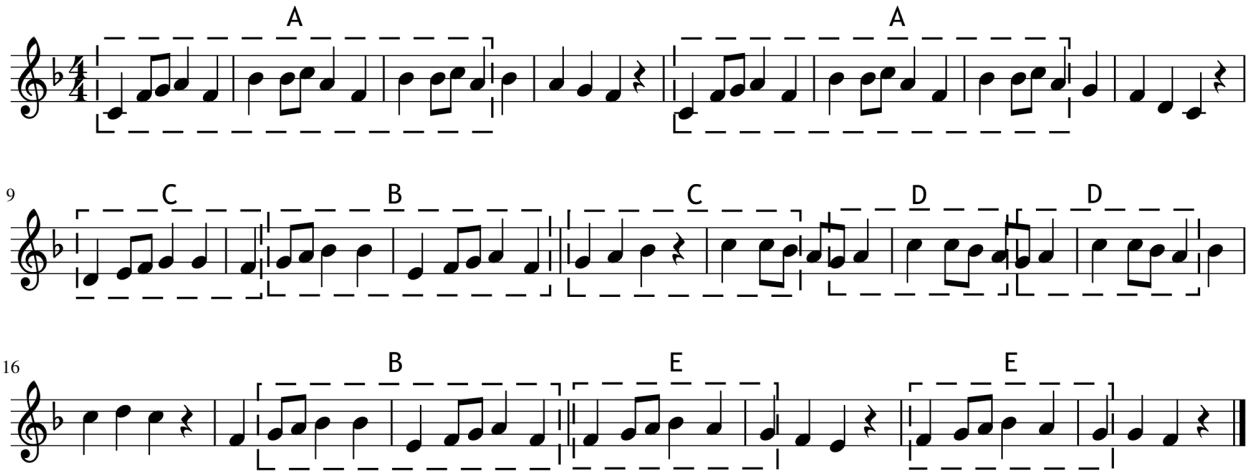


Fig 7. Segmentation example. Example of a segmentation performed on the template piece *Abiatu da bere bidean* used in this work. All the different segments are labelled from A to E, where A, B, D and E are repetition segments and C is a transposed segment.

https://doi.org/10.1371/journal.pone.0191417.g007

On trained and validated models, sequences having high probability are assumed to retain more aspects of the music style under consideration than sequences with low probability, therefore, they are considered better melodies.

Sampling. The sampling process consists on generating new melodic information that respects the coherence structure extracted from the template piece with a high probability according to the statistical model created from the corpus. For sampling a *stochastic hill climbing* optimization method is used, which is iterated 10^4 times. This method takes a new piece as a starting point, which respect the coherence structure extracted from the template piece and which has random notes sampled into the different segments of the structure. To create it a left to right sampling is used, which samples random notes into each position of the piece, including the positions that are not part of any segment of the coherence structure. Every time a whole segment is sampled all the other occurrences of the segment are also sampled. In Fig 8 an example of a piece generated as a starting point for this method is shown. The highlighted segments show that the coherence of the template piece is respected, but the notes within the

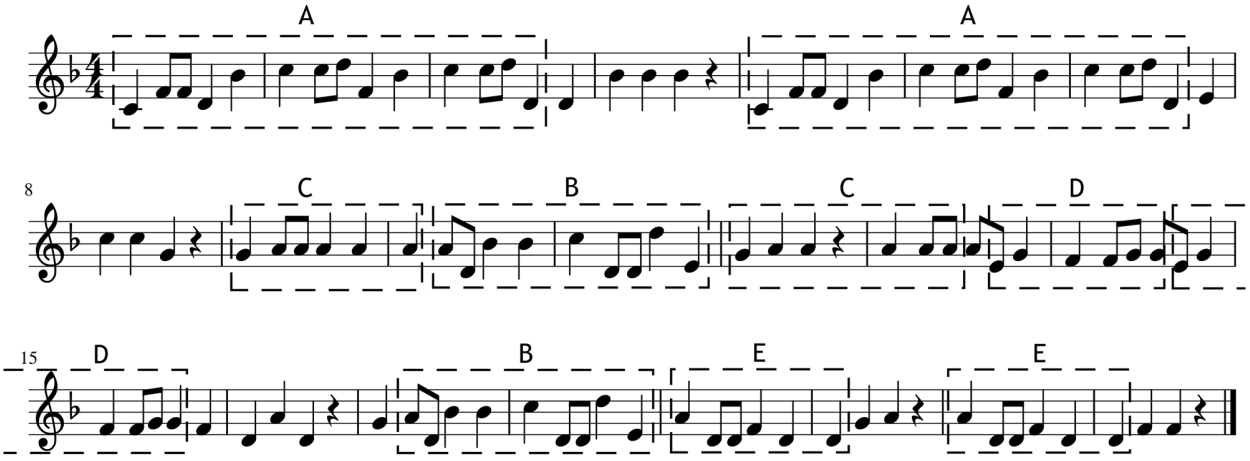


Fig 8. Sampling starting point. Example of a starting point for the stochastic hill climbing method.

https://doi.org/10.1371/journal.pone.0191417.g008

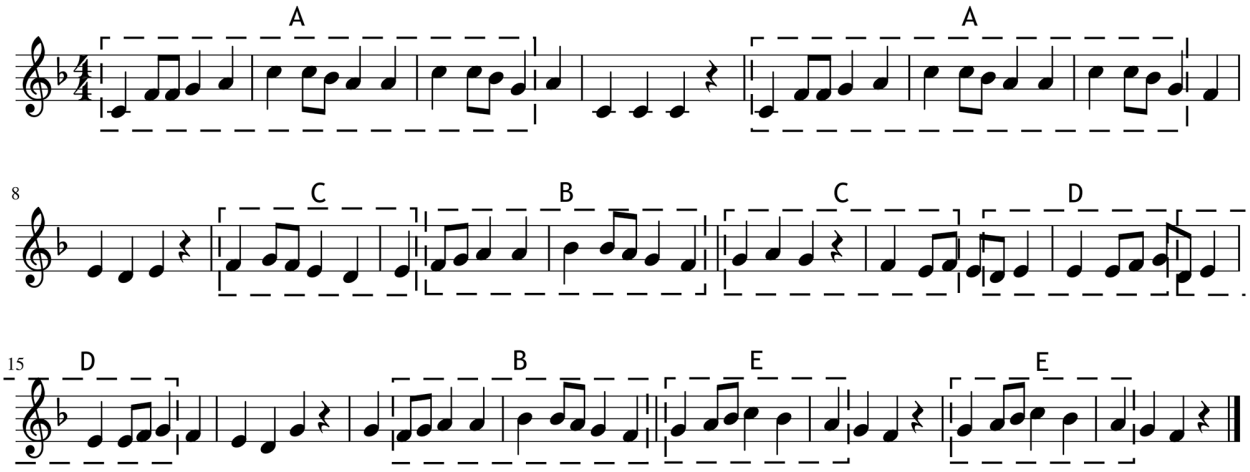


Fig 9. Generation example. Example of a melody generated using the coherence structure of the melody *Abiatu da bere bidean*, shown in Fig 7.

https://doi.org/10.1371/journal.pone.0191417.g009

segments are randomly selected. It can be seen that the melody is not smooth, it has many big leaps between the notes, which is not very common in the melodies used in the corpus, making its probability low.

In order to improve the generated piece the method modifies it iteratively, where in each iteration a random location in the piece is chosen and a random note from the vocabulary of the template piece is substituted into that position, producing a new piece with an updated probability, computed using the Eq 1. If the new probability is higher than the last saved one the change is conserved. To conserve the coherence structure of the original template every time a position that is covered by a segment is changed all the other occurrences of that segment are also changed. Fig 9 shows an example generation guided by the coherence structure of the template piece shown in Fig 7. It can be seen that even if the melodies are different they share the repetition structure, which should endow the generations with coherence.

Experimental setup

A set of 100 random pieces of the corpus described in Section 'corpus' used to extract a representation of pitch class interval and five point contour viewpoints of each piece, from which the contour and interval matrices of each melody are computed. These matrices are then used to perform an unsupervised classification and group similar songs into clusters. These clusters are then used to build statistical models that are used in the automatic music generation process.

A first experiment with the melody named *Abiatu da bere bidean*, which is part of the corpus, but is not part of the 100 piece set, is used to extract the coherence structure that guides the generation, along with the statistical models computed from the clusters identified in the classification process. 10 different generations have been made for each cluster, and they have been represented as contour and interval matrices to be classified in the next step. Three extra experiments have been performed with three more melodies randomly chosen from the corpus.

Experimental results

As commented in the previous section, two types of matrices have been obtained for each melody, and both have been used to test the proposed approach.

Table 1. Contour: Obtained accuracies by distance type and cluster number.

Cluster Num	2	3	4	5	6	Mean
1-norm	0.500	0.417	**0.500**	0.350	**0.583**	0.470
∞-norm	0.500	0.417	0.250	0.250	0.208	0.325
M-norm	0.750	**0.750**	0.438	**0.550**	0.417	**0.581**
F-norm	0.625	0.417	0.375	0.100	0.250	0.353
EMD	**0.875**	0.667	**0.500**	0.450	0.333	0.565
Jeffrey	0.500	0.333	0.250	0.250	0.167	0.300
Manhattan	0.500	0.417	0.250	0.250	0.167	0.317
Intersect	0.375	0.333	0.250	0.200	0.125	0.257
KL	0.375	0.333	0.313	0.450	0.417	0.378

https://doi.org/10.1371/journal.pone.0191417.t001

Contour

Obtained classification accuracies are shown in Table 1. As it can be appreciated, obtained results are very different regarding the used distance and the number of cluster selected. It can be inferred, indeed, that there is a distance, EMD, which out-stands clearly from the other when a low number of clusters is used. As a matter of fact, the best results are obtained using this EMD distance for cluster numbers 2 and 4; concerning to other number of clusters, normalized distances appear to be the best choice, being M-norm which obtains the best mean among all. It is worth remarking the result obtained by 1-norm distance when six clusters are used: it obtains by far the best result among all the distances used (0.583).

Interval

The same experiment has been repeated, using Interval type matrices, and the obtained accuracy results have been presented in Table 2. In this case, EMD distance out-stands as the best one in the performed experiments; best results are obtained using this distance for 3 to 6 clusters, and the best mean is obtained with this distance as well. Remarkable result of Manhattan distance for two clusters (0.875), which makes it candidate for low cluster situations; it obtains the second best mean among all distances.

Extra experiments

In order to provide a better overview of the proposed approach, a set of extra experiments have been set up; 3 pieces have been randomly selected for the corpus. These new three

Table 2. Interval: Obtained accuracies by distance type and cluster number.

Cluster Num	2	3	4	5	6	Mean
1-norm	0.500	0.333	0.375	**0.400**	0.292	0.380
∞-norm	0.750	0.333	0.313	0.250	0.167	0.363
M-norm	0.625	0.333	0.250	0.200	0.250	0.332
F-norm	0.500	0.167	0.250	0.200	0.333	0.290
EMD	0.500	**0.667**	**0.625**	**0.400**	**0.542**	0.547
Jeffrey	0.750	0.500	0.313	0.150	0.083	0.359
Manhattan	**0.875**	0.333	0.250	**0.400**	0.375	0.447
Intersect	0.500	**0.667**	0.188	0.150	0.125	0.326
KL	0.625	0.333	0.250	0.200	0.083	0.298

https://doi.org/10.1371/journal.pone.0191417.t002

Table 3. Contour: Obtained accuracies by distance type and cluster number (melody ID 1360).

Cluster Num	2	3	4	5	6	Mean
1-norm	0.500	0.333	0.250	**0.380**	0.283	0.349
∞-norm	0.500	0.333	0.450	0.360	**0.333**	0.395
M-norm	**0.750**	**0.500**	**0.475**	0.320	0.183	**0.446**
F-norm	0.500	0.333	0.300	0.140	0.183	0.291
EMD	0.500	0.367	0.300	0.240	0.267	0.335
Jeffrey	0.400	0.333	0.250	0.160	0.017	0.232
Manhattan	0.500	0.333	0.300	0.280	0.183	0.319
Intersect	0.250	0.300	0.250	0.200	0.183	0.237
KL	0.550	0.400	0.275	0.320	0.217	0.352

https://doi.org/10.1371/journal.pone.0191417.t003

melodies are *Aita semeak tabernan daude I* (which from now on will be identified with the melody ID 1360), *Gure herriko bikariuak* (melody ID 1476) and *Zazpi ahizparen gai den oihala I* (melody ID 1599). The approach presented in this paper has been applied taking as template piece each melody of the new experiment set.

Tables 3 and 4 show the obtained results for the first piece (melody ID 1360) for contour and interval representation respectively. As it can be seen, the same result is obtained for the 2 clusters scenario, but the results differ between both representations in the remaining cluster numbers considered. Interval representation is slightly better, although the best distance mean is obtained by M-norm in the Contour case. Different distances obtain the best result for different cluster numbers, which indicates that the appropriate one should be carefully selected for each considered case.

Regarding the second piece (melody ID 1476), obtained results are shown in Tables 5 (contour) and 6 (interval). In this case, interval representation is the best one, being the best mean accuracy obtained using the EMD distance. When the number of clusters is 2 or 3, the M-norm distance is the one which obtains better results.

For the third selected musical piece (melody ID 1599) the obtained results are shown in Tables 7 and 8 for contour and interval representation respectively. Once again, interval is the best representation, and the results differ depending on the number of clusters used. The best mean is obtained by M-norm distance for contour representation.

It is worth mentioning that the results obtained in the extra experiments do not differ with the ones shown in Tables 1 and 2 which indicates that the proposed approach gives an accurate way to classify different songs once the model has been trained using an appropriate subset of representative melodies.

Table 4. Interval: Obtained accuracies by distance type and cluster number (melody ID 1360).

Cluster Num	2	3	4	5	6	Mean
1-norm	0.500	0.333	0.350	**0.360**	0.317	0.372
∞-norm	0.650	0.333	0.250	0.300	0.183	0.343
M-norm	0.500	0.333	0.275	0.200	0.267	0.315
F-norm	0.500	0.333	0.250	0.200	0.267	0.310
EMD	0.500	0.333	**0.475**	0.200	0.267	0.355
Jeffrey	0.650	0.300	0.325	0.320	0.133	0.346
Manhattan	0.700	0.367	0.425	0.260	0.317	**0.414**
Intersect	0.500	**0.667**	0.250	0.240	**0.350**	0.401
KL	**0.750**	0.233	0.075	0.200	0.067	0.265

https://doi.org/10.1371/journal.pone.0191417.t004

Table 5. Contour: Obtained accuracies by distance type and cluster number (melody ID 1476).

Cluster Num	2	3	4	5	6	Mean
1-norm	0.500	0.333	0.300	0.400	0.333	0.373
∞-norm	0.500	0.333	0.325	0.300	0.233	0.338
M-norm	0.500	0.600	**0.450**	0.380	**0.433**	**0.473**
F-norm	0.550	0.333	0.275	0.260	0.183	0.320
EMD	**0.650**	**0.533**	0.425	0.420	0.200	0.446
Jeffrey	0.500	0.400	0.275	0.100	0.233	0.302
Manhattan	0.500	0.333	0.275	0.220	0.300	0.326
Intersect	0.200	0.033	0.250	0.140	0.150	0.155
KL	0.550	0.400	0.300	**0.440**	0.133	0.365

https://doi.org/10.1371/journal.pone.0191417.t005

Table 6. Interval: Obtained accuracies by distance type and cluster number (melody ID 1476).

Cluster Num	2	3	4	5	6	Mean
1-norm	0.650	0.400	0.500	0.280	0.300	0.426
∞-norm	0.600	0.433	0.325	0.160	0.183	0.340
M-norm	**0.800**	**0.733**	0.450	0.160	0.317	0.492
F-norm	0.500	0.333	0.300	0.260	0.217	0.322
EMD	0.750	0.433	**0.550**	**0.440**	**0.350**	**0.505**
Jeffrey	0.650	0.067	0.075	0.160	0.050	0.200
Manhattan	0.500	0.467	0.375	0.280	0.233	0.371
Intersect	0.500	0.400	0.350	0.140	0.133	0.305
KL	0.450	0.400	0.200	0.060	0.233	0.269

https://doi.org/10.1371/journal.pone.0191417.t006

Table 7. Contour: Obtained accuracies by distance type and cluster number (melody ID 1599).

Cluster Num	2	3	4	5	6	Mean
1-norm	0.500	0.333	**0.500**	**0.500**	**0.500**	0.467
∞-norm	0.500	0.333	0.250	0.200	0.200	0.297
M-norm	0.700	**0.800**	**0.500**	0.440	0.417	**0.571**
F-norm	0.600	0.500	0.400	0.200	0.333	0.407
EMD	**0.750**	0.667	**0.500**	0.400	0.267	0.517
Jeffrey	0.350	0.500	0.450	0.360	0.033	0.339
Manhattan	0.500	0.567	0.350	0.240	0.400	0.411
Intersect	0.650	0.367	0.275	0.160	0.133	0.317
KL	0.200	0.367	0.350	0.220	0.233	0.274

https://doi.org/10.1371/journal.pone.0191417.t007

Table 8. Interval: Obtained accuracies by distance type and cluster number (melody ID 1599).

Cluster Num	2	3	4	5	6	Mean
1-norm	0.700	0.467	0.675	**0.500**	0.400	**0.548**
∞-norm	**0.900**	0.333	0.250	0.340	0.267	0.418
M-norm	0.750	**0.600**	0.400	0.320	0.300	0.474
F-norm	0.500	0.333	0.250	0.200	0.250	0.307
EMD	0.550	0.433	**0.725**	0.400	0.450	0.512
Jeffrey	0.500	0.267	0.125	0.140	0.167	0.240
Manhattan	0.500	0.500	0.350	0.360	**0.483**	0.439
Intersect	0.500	0.533	0.175	0.160	0.133	0.300
KL	0.600	0.500	0.275	0.120	0.067	0.312

https://doi.org/10.1371/journal.pone.0191417.t008

Conclusions and future works

In this paper an investigation of the classification of automatically generated melodies is performed; the main idea that grouping close related known pieces in different sets –or clusters–, and afterwards generating new melodies in an automatic way, which are somehow "inspired" in each set. The new melodies are supposed to be classified to this set, using the same distance used to identify the clusters.

Although obtained results could be seen as not so good for other kind of data –we do not expect a medical research giving us a 66% of suffering a disease, or a industrial task telling us that certain piece is among tolerance-threshold on a 56% probability– it has to be remarked the artistic environment the performed experiment have been carried out, in an area which is no deterministic, and in genres that could be confused among each other.

Nevertheless, obtained results indicate the appropriateness of the whole process: results over 0.5 can be considered encouraging, especially when the cluster number is 4 or more. Some extra experiments have been performed using three different songs as template, and using the previously obtained clustering as classification model. Obtained results are similar to the previous ones, which indicates the soundness of the proposed approach.

As future work a deeper analysis is envisaged, and a combination of both representations (contour and interval) in order to obtain a better idea of the genre divisions obtained by the clustering process. Another open line remain in the use of different distances to classify the new generated melodies and to divide the existing songs in different clusters. On the music generation topic the rhythm generation and the use of harmonic information to generate melodies are lines that should also be studied in the future.

Author Contributions

Investigation: Izaro Goienetxea, José María Martínez-Otzeta, Basilio Sierra.

Methodology: Izaro Goienetxea, José María Martínez-Otzeta, Basilio Sierra.

Software: Izaro Goienetxea, José María Martínez-Otzeta, Basilio Sierra.

Writing – original draft: Izaro Goienetxea, José María Martínez-Otzeta, Basilio Sierra.

Writing – review & editing: Izaro Goienetxea, José María Martínez-Otzeta, Basilio Sierra, Iñigo Mendialdua.

References

1. Fu Z, Lu G, Ting KM, Zhang D. A Survey of Audio-Based Music Classification and Annotation. Multimedia, IEEE Transactions on. 2011; 13(2):303–319. https://doi.org/10.1109/TMM.2010.2098858
2. Hillewaere R. Computational models for folk music classification. Vrije Universiteit Brussel; 2013.
3. Guo D, Li SZ. Content-based Audio Classification and Retrieval by Support Vector Machines. IEEE Trans on Neural Networks. 2003; 14(1):209–215. https://doi.org/10.1109/TNN.2002.806626 PMID: 18238003
4. Mandel MI, Ellis D. Song-Level Features and Support Vector Machines for Music Classification. In: ISMIR. vol. 2005; 2005. p. 594–599.
5. Kotsifakos A, Kotsifakos EE, Papapetrou P, Athitsos V. Genre classification of symbolic music with SMBGT. In: Proceedings of the 6th International Conference on PErvasive Technologies Related to Assistive Environments. ACM; 2013. p. 44.
6. Dieleman S, Brakel P, Schrauwen B. Audio-based music classification with a pretrained convolutional network. In: Proceedings of the 12th international society for music information retrieval conference. University of Miami; 2011. p. 669–674.
7. Chiliguano P, Fazekas G. Hybrid music recommender using content-based and social information. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016,

Shanghai, China, March 20–25, 2016; 2016. p. 2618–2622. Available from: https://doi.org/10.1109/ICASSP.2016.7472151.

8. Zhang C, Evangelopoulos G, Voinea S, Rosasco L, Poggio T. A deep representation for invariance and music classification. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE; 2014. p. 6984–6988.
9. Fernández JD, Vico F. AI Methods in Algorithmic Composition: A Comprehensive Survey. Journal of Artificial Intelligence Research. 2013; 48(1):513–582.
10. Hedges SA. Dice Music in the Eighteenth Century. Music & Letters. 1978; 59(2):180–187. https://doi.org/10.1093/ml/59.2.180
11. Conklin D. Music Generation from Statistical Models. In: Proceedings of the AISB 2003 Symposium on Artificial Intelligence and Creativity in the Arts and Sciences. Aberystwyth, Wales; 2003. p. 30–35.
12. Leach J, Fitch J. Nature, Music, and Algorithmic Composition. Computer Music Journal. 1995; 19 (2):23–33. https://doi.org/10.2307/3680598
13. Meyer LB. Meaning in Music and Information Theory. Journal of Aesthetics and Art Criticism. 1957; 15:412–424. https://doi.org/10.2307/427154
14. Anagnostopoulou C. Cohesion in Linguistic and Musical Discourse. In: Proceedings of the 3rd European Society for the Cognitive Sciences of Music Conference. Uppsala, Sweden; 1997.
15. Bimbot F, Deruty E, Sargent G, Vincent E. Semiotic structure labeling of music pieces: Concepts, methods and annotation conventions. In: 13th International Society for Music Information Retrieval Conference (ISMIR). Porto, Portugal; 2012. p. 235–240. Available from: https://hal.inria.fr/hal-00758648.
16. Collins T, Laney R, Willis A, Garthwaite PH. Developing and evaluating computational models of musical style. Artificial Intelligence for Engineering Design, Analysis and Manufacturing. 2014.
17. Garzia J, Egaña A, Sarasua J. The Art of Bertsolaritza: Improvised Basque Verse Singing. Donostia, Bertsozale Elkartea; Andoain, Bertsolari Liburuak; 2001.
18. Chai W, Vercoe B. Folk music classification using hidden Markov models. In: Proceedings of International Conference on Artificial Intelligence. vol. 6. Citeseer; 2001.
19. Bergstra J, Casagrande N, Erhan D, Eck D, Kégl B. Aggregate features and AdaBoost for music classification. Machine learning. 2006; 65(2-3):473–484. https://doi.org/10.1007/s10994-006-9019-7
20. Xu C, Maddage NC, Shao X. Automatic music classification and summarization. IEEE transactions on speech and audio processing. 2005; 13(3):441–450. https://doi.org/10.1109/TSA.2004.840939
21. Fu Z, Lu G, Ting KM, Zhang D. A survey of audio-based music classification and annotation. IEEE Transactions on Multimedia. 2011; 13(2):303–319. https://doi.org/10.1109/TMM.2010.2098858
22. Pinquier J, Rouas JL, E-OBRECHT RA. Robust speech/music classification in audio documents. Entropy. 2002; 1(2):3.
23. Sturm BL. Classification accuracy is not enough. Journal of Intelligent Information Systems. 2013; 41 (3):371–406. https://doi.org/10.1007/s10844-013-0250-y
24. Panagakis Y, Kotropoulos C, Arce GR. Music genre classification via sparse representations of auditory temporal modulations. In: 2009 17th European Signal Processing Conference; 2009. p. 1–5.
25. Valverde-Rebaza J, Soriano A, Berton L, de Oliveira MCF, de Andrade Lopes A. Music genre classification using traditional and relational approaches. In: Intelligent Systems (BRACIS), 2014 Brazilian Conference on. IEEE; 2014. p. 259–264.
26. Febres G, Jaffe K. Music viewed by its entropy content: A novel window for comparative analysis. PLOS ONE. 2017; 12(10):1–30. https://doi.org/10.1371/journal.pone.0185757
27. Lee CH, Lin HS, Chen LH. Music classification using the bag of words model of modulation spectral features. In: 2015 15th International Symposium on Communications and Information Technologies (ISCIT). IEEE; 2015. p. 121–124.
28. Zelaia A, Alegria I, Arregi O, Sierra B. A multiclass/multilabel document categorization system: Combining multiple classifiers in a reduced dimension. Applied Soft Computing. 2011; 11(8):4981–4990. https://doi.org/10.1016/j.asoc.2011.06.002
29. Zelaia A, Arregi O, Sierra B. Combining Singular Value Decomposition and a multi-classifier: A new approach to support coreference resolution. Engineering Applications of Artificial Intelligence. 2015; 46:279–286. https://doi.org/10.1016/j.engappai.2015.09.007
30. Mallat S. Understanding deep convolutional networks. Phil Trans R Soc A. 2016; 374(2065):20150203. https://doi.org/10.1098/rsta.2015.0203 PMID: 26953183
31. Karatzoglou A, Hidasi B, Tikk D, Sar-Shalom O, Roitman H, Shapira B. RecSys' 16 Workshop on Deep Learning for Recommender Systems (DLRS). In: Proceedings of the 10th ACM Conference on Recommender Systems. ACM; 2016. p. 415–416.

32. Dorfer M, Arzt A, Widmer G. Towards score following in sheet music images. In: Proceedings of the International Society for Music Information Retrieval Conference (ISMIR); 2016.
33. Rajanna AR, Aryafar K, Shokoufandeh A, Ptucha R. Deep Neural Networks: A Case Study for Music Genre Classification. In: 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA). IEEE; 2015. p. 655–660.
34. Chun L, Song H, Yang J. Research on music classification based on MFCC and BP neural network. In: 2nd International Conference on Information, Electronics and Computer. Atlantis Press; 2014.
35. Bassiou N, Kotropoulos C, Papazoglou-Chalikias A. Greek folk music classification into two genres using lyrics and audio via canonical correlation analysis. In: 2015 9th International Symposium on Image and Signal Processing and Analysis (ISPA). IEEE; 2015. p. 238–243.
36. Hillewaere R, Manderick B, Conklin D. Alignment Methods for Folk Tune Classification. In: Spiliopoulou M, Schmidt-Thieme L, Janning R, editors. Data Analysis, Machine Learning and Knowledge Discovery. Springer International Publishing; 2014. p. 369–377. Available from: http://dx.doi.org/10.1007/978-3-319-01595-8_40.
37. Conklin D, Witten IH. Multiple Viewpoint Systems for Music Prediction. Journal of New Music Research. 1995; 24:51–73. https://doi.org/10.1080/09298219508570672
38. Jain AK, Dubes RC. Algorithms for Clustering Data. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.; 1988.
39. Sierra B, Lazkano E, Jauregi E, Irigoien I. Histogram Distance-based Bayesian Network Structure Learning: A Supervised Classification Specific Approach. Decis Support Syst. 2009; 48(1):180–190. https://doi.org/10.1016/j.dss.2009.07.010
40. Conklin D. Multiple Viewpoint Systems for Music Classification. Journal of New Music Research. 2013; 42(1):19–26. https://doi.org/10.1080/09298215.2013.776611

11 Melody classification with pattern covering

Authors: Izaro Goienetxea and Darrell Conklin.

Booktitle: Proceedings of 9th International Workshop on Music and Machine Learning (MML 2016).

Year: 2016

Melody classification with pattern covering

Izaro Goienetxea[1] Kerstin Neubarth[2] Darrell Conklin[1,3]

[1] Department of Computer Science and Artificial Intelligence
University of the Basque Country UPV/EHU, San Sebastián, Spain
[2] Canterbury Christ Church University, United Kingdom
[3] IKERBASQUE: Basque Foundation for Science, Bilbao, Spain

Abstract. In this work a method for tune family classification is proposed, based on pattern sets and nearest neighbor classification. The method computes a covering of two pieces with shared statistically interesting patterns revealed by pattern discovery, which is used to measure the similarity between pieces. On a corpus of 360 Dutch folk melodies the method achieves a maximum classification accuracy of 95.6%.

1 Introduction

Classification of folk tunes into tune families has long been a research topic in folk music studies and music data mining. Tune families are groups of related melodies which presumably share common ancestors [2]. Generally documentary evidence of historic origin of tunes is unavailable, and tune families may be proposed based on melodic similarity between tunes [13].

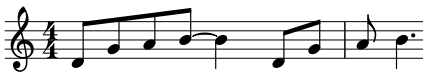
Computational studies of tune families have applied different techniques like content-based retrieval [10] or predictive classification [4]. For tune family classification nearest neighbor methods have achieved good classification results, compared to other classification tasks including classification into folk music genres or geographical regions [6]. Nearest neighbor methods compare an unclassified example against already classified examples, and assign the unclassified example a class based on one (1-NN) or more (k-NN) most similar examples. Applications for automatic tune family classification measure similarity between tunes by geometric distance on global-feature [11] or wavelet representations [12], edit distance on single-viewpoint string representations [6] or combined event features [11], and compression distance on point-set [8] or multiple-viewpoint representations [7].

This paper presents a classification method which represents tunes by sequences of features and measures similarity between pairs of tunes based on the probability of interesting patterns covering both tunes.

2 Data and Methods

As a data corpus this study uses a subset of the Meertens Folk Tune Collection: The Annotated Corpus[1] is a collection of 360 Dutch folk song melodies classified

[1] http://www.liederenbank.nl/



pitch	62	67	69	71	62	67	69	71
dur	840	840	840	2520	840	840	840	2520
int	$\perp$	5	2	2	-9	5	2	2
ioi	$\perp$	840	840	840	2520	840	840	840
intref	P5	P1	M2	M3	P5	P1	M2	M3
c3(pitch)	$\perp$	$+$	$+$	$+$	$-$	$+$	$+$	$+$
c5(pitch, 3)	$\perp$	$++$	$+$	$+$	$--$	$++$	$+$	$+$
c3(dur)	$\perp$	$=$	$=$	$+$	$-$	$=$	$=$	$+$
c3i(level)	$\perp$	$+$	$-$	$+$	$-$	$+$	$-$	$+$
int $\otimes$ intref	$\perp$	5, P1	2, M2	2, M3	-9, P5	5, P1	2, M2	2, M3
intref $\otimes$ c3(pitch)	$\perp$	P1, $+$	M2, $+$	M3, $+$	P5, $-$	P1, $+$	M2, $+$	M3, $+$

Fig. 1: Small fragment from a Dutch tune. Each line shows a different viewpoint and its transformation of the event sequence. Top: basic viewpoints, middle: derived viewpoints, bottom: linked viewpoints.

into 26 tune families. Tune families contain between 8 and 27 melodies, each of them with a length of around 150 notes.

To describe the pieces of the corpus a multiple viewpoint representation is used [5]. A viewpoint τ is a function that maps an event sequence $e_1, \ldots, e_\ell$ to a more abstract derived sequence $\tau(e_1), \ldots, \tau(e_\ell)$, comprising elements in the codomain of the function τ. Figure 1 presents a short tune fragment with different viewpoints used in this work, including interval from the previous note int or key note intref, 3-point contour c3 of pitch, duration or inverse metric level, 5-point contour of pitch c5(pitch,3), which records whether the note was approached by a leap (three seminotes or larger), a step (less than three semitones) or unison, and inter-onset-interval ioi. Another viewpoint used in this work is phrpos which records whether the note is first, last or inside a phrase [9].

A *pattern* is a sequence of event features described using viewpoints, and a piece instantiates a pattern if the pattern occurs one or more times in the piece. The number of pieces instantiating a pattern gives the *piece count* of the pattern. Melodic patterns are discovered with a sequential pattern discovery algorithm [1], which is run once on the whole corpus and extracts all repeated patterns from the viewpoint representations of the corpus. Among these patterns the *interesting* ones are identified to be used in the classification: a pattern is considered the more interesting the more surprising is its occurrence in both the query and the current target tune. More formally, using a binomial distribution the probability of a pattern P of length c occurring one or more times in a tune of length ℓ is $\mathbb{B}_{\geq}(1; \ell - c + 1; p)$, where p is the background probability of pattern P calculated from a zero-order model of the training corpus. Then the expected

piece count of the pattern in a query tune of length ℓ_q and a target tune of length ℓ_t is

$$\lambda = \mathbb{B}_{\geq}(1; \ell_q - c + 1; p) + \mathbb{B}_{\geq}(1; \ell_t - c + 1; p)$$

and the interest $\mathbb{I}$ of pattern P is the deviation between the expected piece count of the pattern (λ) and its actual piece count (always 2 in this case), computed using the negative logarithm of the Poisson approximation of the binomial distribution

$$\mathbb{I}(P) = \lambda + \ln(2) - 2\ln(\lambda)$$

with a higher value of $\mathbb{I}(P)$ indicating a more surprising pattern. The similarity between the query and target tune is the summed interest of patterns shared by the two tunes.

To identify the set of most surprising patterns shared by both query and target tunes, a *covering* method is applied. Candidate patterns are sorted by their interest $\mathbb{I}$, which is computed excluding the statistics of the query tune in the zero-order model, and the sorted pattern list is processed iteratively to choose the best pattern that in each iteration fits into some of the positions of the pieces that have not been yet covered by any pattern, not allowing overlapping between contiguous patterns. The covering thus produces a *pattern set*.

Two situations can be distinguished. (1) Single viewpoints: patterns are discovered using one viewpoint representation at a time, and all patterns in the covering pattern set have the same single viewpoint, where the viewpoint can be primitive or linked (e.g. int or int $\otimes$ ioi). (2) Multiple viewpoints: patterns are discovered for each of the chosen viewpoints, and the pattern set covering the tunes can include patterns of different viewpoints (e.g. both intref and c3(pitch) patterns). In both cases, the covering results in one pattern set for each pair of tunes, from which the similarity score is computed. The query piece is assigned the class of the most similar labelled tune.

3 Results and Discussion

The classification results are presented in Table 1, which shows the three best results obtained using single viewpoint (top), the best results obtained with multiple viewpoints (middle) and the results obtained with Fully Saturated Viewpoints (bottom). Fully Saturated and Linked Viewpoints contain all possible dyadic linked viewpoints formed from the following viewpoints: intref, c3(dur), c3(pitch), c5(pitch, 3), c3i(level), int, ioi, phrpos. Fully Saturated Viewpoints contain the single viewpoints in addition to all the linked viewpoints.

Generally, highly accurate classification is achieved with both single and multiple viewpoints, confirming the relevance of event features and patterns or motifs in tune family classification (see also [10, 11]). The top results for linked and multiple viewpoints suggest that for this particular task and dataset metric and phrase information are important in addition to pitch or interval information: 333 out of 347 (96.0%) tunes are correctly classified using a linked viewpoint

Viewpoints	Classification	Accuracy
intref	336/360	93.3%
intref ⊗ c3i(level) ⊗ phrpos	333/347*	96.0%
intref ⊗ c3i(level)	320/347*	92.2%
intref ⊗ c3i(level) ⊗ phrpos & intref ⊗ phrpos	344/360	95.6%
intref & int ⊗ ioi	344/360	95.6%
intref ⊗ c3i(level) ⊗ phrpos & int ⊗ ioi	340/360	94.4%
intref ⊗ c3i(level) & int ⊗ ioi	334/360	92.8%
Fully Saturated and Linked Viewpoints (28)	332/360	92.2%
Fully Saturated Viewpoints (36)	315/360	87.5%

Table 1: Classification accuracy with different viewpoints. (*)Classification on 347 pieces done where the viewpoint c3i(level) is used, since it is undefined for 13 pieces of the corpus.

intref ⊗ c3i(level) ⊗ phrpos, and another 11 tunes correctly classified if this viewpoint is combined with intref ⊗ phrpos to also cover tunes with undefined metric levels. Similarly van Kranenburg et al. [11] reported their highest classification accuracy for a combined edit distance on pitchband, metric weight and phrase position. Alternatively, combining intref and int ⊗ ioi also correctly classifies 344 out of 360 (95.6%) tunes.

All of the viewpoint selections listed in Table 1 achieve higher classification accuracies than earlier studies on the same corpus which used pitch-time representations and nearest neighbor classification (83.9% and 85.6%) [8, 12]. The best results are above the 94.4% accuracy for interval-based edit distance [6] and multiple viewpoint representation with corpus compression distance [7], but slightly lower than the 96.7% accuracy with multiple-viewpoint probabilistic classification [4]. The classification accuracy of 98.9% reported by van Kranenburg et al. [11], using multiple-feature alignment and nearest neighbor classification, has not yet been achieved by any other method.

The results obtained in this work (average accuracies in leave-one-out) indicate that the pattern discovery, ranking and covering presented in this work is effective for tune family classification. The method differs from previous pattern-based approaches in several ways. Compared to a representation by compressed viewpoints [7], our method represents tunes by explicitly described pattern sets, which could be inspected to gain further insight into the similarity between tunes; in addition the method supports tunes being represented by heterogeneous sets of viewpoint patterns. Explicit patterns are employed in two earlier studies [13, 3]. The first of these manually defines interesting motif classes; in contrast, our classification method integrates automatic pattern discovery and ranking. The second study applies distinctive pattern discovery to generate a decision list of patterns ranked by their confidence on the complete training corpus; a test tune is classified based on a single most confident matching pattern. The method pre-

sented in this paper, on the other hand, is a lazy learning method which does not require a prior training phase on labelled examples.

Acknowledgements This research is supported by the project Lrn2Cre8 which is funded by the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 610859.

References

1. J. Ayres, J. Flannick, J. Gehrke, and T. Yiu. Sequential PAttern Mining Using a Bitmap Representation. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '02)*, pages 429–435, New York, NY, USA, 2002. ACM.
2. S. P. Bayard. Prolegomena to a study of the principal melodic families of British-American folk song. *The Journal of American Folklore*, 63(247):1–44, 1950.
3. D. Conklin. Melody classification using patterns. In *Proceedings of the 2nd International Workshop on Machine Learning and Music at ECML/PKDD 2009 (MML 2009)*, pages 37–41, Bled, Slovenia, 2009.
4. D. Conklin. Fusion functions for multiple viewpoints. In *5th International Workshop on Music and Machine Learning at ECML/PKDD 2013 (MML 2013)*, Prague, Czech Republic, 2013.
5. D. Conklin. Multiple Viewpoint Systems for Music Classification. *Journal of New Music Research*, 42(1):19–26, 2013.
6. R. Hillewaere, B. Manderick, and D. Conklin. Alignment methods for folk tune classification. In M. Spiliopoulou, L. Schmidt-Thieme, and R. Janning, editors, *Data Analysis, Machine Learning and Knowledge Discovery*, pages 369–377. Springer International Publishing, 2014.
7. C. Louboutin and D. Meredith. Using general-purpose compression algorithms for music analysis. *Journal of New Music Research*, 45(1):1–16, 2016.
8. D. Meredith. Using point-set compression to classify folk songs. In *Proceedings of the 4th International Workshop on Folk Music Analysis (FMA 2014)*, pages 29–35, Istanbul, Turkey, 2014.
9. P. van Kranenburg and D. Conklin. A pattern mining approach to study a collection of dutch folk-songs. In *Proceedings of the 6th International Workshop on Folk Music Analysis (FMA 2016)*, pages 71–73, Dublin, Ireland, 2016.
10. P. van Kranenburg, A. Volk, and F. Wiering. On identifying folk song melodies employing recurring motifs. In *Proceedings of the 12th International Conference on Music Perception and Cognition and the 8th Triennial Conference of the European Society for the Cognitive Sciences of Music*, pages 1057–1062, Thessaloniki, 2012.
11. P. van Kranenburg, A. Volk, and F. Wiering. A comparison between global and local features for computational classification of folk song melodies. *Journal of New Music Research*, 42(1):1–18, 2013.
12. G. Velarde, T. Weyde, and D. Meredith. Wavelet-filtering of symbolic music representations for folk tune segmentation and classification. In *Proceedings of the 3rd International Workshop on Folk Music Analysis (FMA 2013)*, pages 56–62, Amsterdam, Netherlands, 2013.
13. A. Volk and P. van Kranenburg. Melodic similarity among folk songs: An annotation study on similarity-based categorization in music. *Musicae Scientiae*, 16(3):317–339, 2012.

12

On the Use of Matrix Based Representation to Deal with Automatic Composer Recognition

Authors: Izaro Goienetxea, Iñigo Mendialdua and Basilio Sierra.

Booktitle: Proceedings of AI 2018: Advances in Artificial Intelligence - 31st Australasian Joint Conference,

Year: 2018

Publisher: Springer

On the Use of Matrix Based Representation to Deal with Automatic Composer Recognition

Izaro Goienetxea[1(✉)], Iñigo Mendialdua[2], and Basilio Sierra[1]

[1] Department of Computer Science and Artificial Intelligence, University of the Basque Country UPV/EHU, San Sebastian, Spain
{izaro.goienetxea,inigo.mendialdua,b.sierra}@ehu.eus
[2] Department of Computer Languages and Systems, University of the Basque Country UPV/EHU, San Sebastian, Spain

Abstract. In this article the use of a matrix based representation of pieces is tested for the classification of musical pieces of some well known classical composers. The pieces in two corpora have been represented in two ways: matrices of interval pair probabilities and a set of 12 global features which had previously been used in a similar task. The classification accuracies of both representations have been computed using several supervised classification algorithms. A class binarization technique has also been applied to study how the accuracies change with this kind of methods. Promising results have been obtained which show that both the matrix representation and the class binarization techniques are suitable to be used in the automatic composer recognition problem.

Keywords: Matrices · Pairwise classification · Composer recognition

1 Introduction

Automatic music classification is a task within the field of Music Information Retrieval (MIR) which is getting more attention with the growth of the available information, thanks to the digital media. When dealing with automatic composer recognition it is important to choose the features that will be used to represent the pieces. Global feature sets, n-grams and string methods have been used to represent folk song collections [11], as well as event models [10]. Other works represent the pieces with multiple viewpoints [1] or discover patterns within collections of pieces, to find melodic families [7,12] within them. Herremans et al. [9] use a 12 global feature set to classify pieces of three classical composers, and Dor and Reich [3] manually extract some pitch based features and use a classifier tool named CHECKUP to discover more features that they then use to classify pieces of nine composers.

After the pieces are represented using one of the methods above, a classifier is built, which is first trained with a set of pieces with a known composer, and will

T. Mitrovic et al. (Eds.): AI 2018, LNAI 11320, pp. 531–536, 2018.
https://doi.org/10.1007/978-3-030-03991-2_48

predict a class for new pieces with no composer information. Since usually the classifier has to distinguish among several classes (composers), class binarization techniques can also be applied, to decompose the original multi-class (more than two classes) classification problem into multiple binary sub-problems [5].

In this work the automatic composer recognition problem is studied, using two corpora of symbolic representation of pieces of well known composers. A corpus of pieces of three composers, Bach, Beethoven and Haydn, similar to the one used in [9], is created, and a matrix based representation presented in the classification method of [6] is used to characterize the pieces. A matrix representation is tested to compare the classification accuracies obtained with it to the results obtained with a global feature set presented in [9], which achieves a promising accuracy. A binarization method is also applied, to see the effect that it has on the classification accuracies. Finally, the corpus is extended with the pieces of two more composers, Mozart and Vivaldi, and their classification accuracies are also computed, in order to observe the effect that the increase of classes has on the accuracies when applying class-binarization techniques.

2 Corpora

The two corpora used in this work have been downloaded from the KernScores website [14], which was developed by the Center for Computer Assisted Research in the Humanities (CCARH), at Stanford University, to organize musical scores.

The first corpus, which from this point will be referenced as corpus_3 includes pieces of the composers Bach, Beethoven and Haydn, similar to the corpus used in [9]. It has a total number of 1138 pieces, and the distribution of the composers and pieces can be seen in the top part of Table 1.

The second corpus, referenced as corpus_5, is an extension of the first one, but it also includes pieces of Mozart and Vivaldi. It contains 1586 pieces, and its composer/piece number distribution can be seen in Table 1. All the pieces used in the corpora are polyphonic MIDI files.

Table 1. Number of pieces of each composer used in this work. The central part shows the composers and piece numbers of corpus composers_3. The two composers of the right extend the first corpus to corpus_5.

Composer	Bach	Beethoven	Haydn	Mozart	Vivaldi
Instances	694	190	254	313	135

3 Methods

The method presented in this work has two main steps; representation and classification. A matrix-based melody representation presented in [6] is tried and the its classification accuracies are compared to the results of a global feature set used in [9], which obtained acceptable accuracy in a similar classification task.

3.1 Representation

The matrix representation of the pieces intends to capture some of their features that ideally would be able to characterize them well enough to be used in the classification process. To create the matrices of the pieces of the corpora, first they are represented using viewpoints. A viewpoint τ is a function that maps an event sequence $e_1, \ldots, e_l$ to a more abstract sequence $\tau(e_1), \ldots, \tau(e_l)$ [2]. In this work an interval viewpoint has been chosen to represent the voices in the pieces; intpc, which computes the pitch class interval (modulo 12). In Fig. 1 the first two bars of the first voice of a Bach chorale included in the corpora can be seen, along with their viewpoint representation.

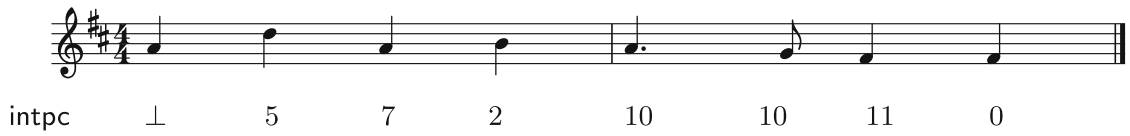


Fig. 1. First two bars the first voice of a Bach chorale and its viewpoint representation.

Once the viewpoint representation of the scores is made, the $\text{matrix}_{\text{intpc}}$ matrices are built, which are 12×12 matrices that describe the probabilities of the transitions between all the pitch class interval pairs that occur in each piece.

From every piece of the corpora a $\text{matrix}_{\text{intpc}}$ has been built and linked to its composer in a *arff* file that is then used by Weka in the classification process. In order to compare the classification results obtained with the matrix representation to the results of the global feature set presented in [9] the jSymbolic feature extractor is used [13]. The used feature collection will be referenced as global_{12}.

3.2 Class Binarization

Class binarization is composed of two main steps; decomposition and combination. In the decomposition step the original problem is divided into several binary sub-problems, for what two main techniques have been developed; One versus All (OVA) and One versus One (OVO). In the classification step, each binary classifier returns a prediction, which need to be combined. When a new instance is being classified using this method all the sub-problems give a prediction of its class, and all these outputs need to be combined. To do so, there are several strategies, but in this work the majority vote strategy [4] is used, where each sub-problem returns a vote, and the class with the largest amount of votes is predicted.

4 Experiments and Results

4.1 Experimental Setup

To test the suitableness of the matrix based representation two experiments have been performed, for which the two corpora (corpus_3 and corpus_5) presented in

Sect. 2 have been used. The pieces in the corpora have been represented with global_{12} and $\text{matrix}_{\text{intpc}}$ representations. We have applied a stratified 10 fold cross-validation in each classification process, and used different base classifiers to study their accuracies, both in multi-class and binarized classifications.

Seven base classifiers from the machine learning software Weka [8] have been used in the classification steps: J48, SMO, JRip, Naive Bayes (NB), Bayesian Network (BNet), Random Forest (RF) and Multilayer Perceptron (MP).

4.2 Results

corpus_3. The classification accuracies of corpus_3 with all the base classifiers, with and without OVO, are presented in Table 2.

Table 2. Accuracy results of the classifications with each single classifier and OVO technique for corpus_3.

	J48	J48-OVO	SMO*	JRIP	JRIP-OVO	NB	NB-OVO
global_{12}	84.007	83.568	86.028	82.074	83.655	66.784	66.872
$\text{matrix}_{\text{intpc}}$	81.459	82.1617	**89.982**	81.986	82.162	80.668	80.580

	BNet	BNet-OVO	RF	RF-OVO	MP	MP-OVO	Mean	Mean-OVO
global_{12}	78.647	78.735	87.171	87.786	87.346	87.434	***81.722***	***82.011***
$\text{matrix}_{\text{intpc}}$	82.513	81.986	86.907	88.401	**89.982**	89.631	***84.559***	***84.986***

The best results are obtained with the $\text{matrix}_{\text{intpc}}$ representation and SMO or Multilayer Perceptron classifier. Even if this representation obtains the best classification accuracy, that does not happen for J48, JRip and Random Forest. The mean accuracies show that overall, better results are obtained with the $\text{matrix}_{\text{intpc}}$ representation and OVO technique.

The choice of the classifier that is used has a great impact on the obtained accuracies. Depending on the classifier that is used, the accuracies can vary from 66.8% to 87.3% in the case of the global_{12} representation.

corpus_5. The classification accuracies of corpus_5 with all the base classifiers, with and without OVO, are presented in the Table 3.

The best accuracy is again obtained with the $\text{matrix}_{\text{intpc}}$ representation and a Multilayer Perceptron classifier, which achieves an accuracy of 80.7%. The results obtained with the $\text{matrix}_{\text{intpc}}$ representation are better than the ones obtained with the global_{12} representation for every classifier but J48. It can be seen that the difference for some classifiers, such as SMO, is significant.

4.3 Statistical Results

We have applied Wilcoxon signed-rank test in order to detect statistical differences between global_{12} and $\text{matrix}_{\text{intpc}}$ representations. The result of the statistical analysis rejects the null hypothesis that both methods are equivalent, since

Table 3. Accuracy results of the classifications with each single classifier and OVO technique for the five composers corpus_5.

	J48	J48-OVO	SMO*	JRIP	JRIP-OVO	NB	NB-OVO
global_{12}	71.402	75.126	73.864	70.328	72.980	56.692	56.692
$\text{matrix}_{\mathsf{intpc}}$	70.266	72.917	80.556	72.033	72.854	71.970	71.843

	BNet	BNet-OVO	RF	RF-OVO	MP	MP-OVO	Mean	Mean-OVO
global_{12}	65.530	67.361	77.904	79.104	74.432	76.641	***70.022***	***71.681***
$\text{matrix}_{\mathsf{intpc}}$	72.096	73.106	79.419	80.556	**80.682**	80.177	***75.289***	***76.001***

the p-value (0.0014) returned by the Wilcoxon test is lower than our α-value (0.01).

We have also carried out another statistical analysis in order to detect statistical differences between OVO and single classifier, in this case we have also applied Wilcoxon signed-rank test. Again, the obtained results rejects the null hypothesis since the p-value (0.0039) returned is lower than our α-value (0.01).

5 Conclusions

In this work the use of a matrix based representation is tested to be used for the automatic recognition of some well known composers. The classification accuracies of the interval matrices with several different base classifiers have been compared to the accuracies obtained with a global feature set which had already been used in a similar classification task with acceptable results. The application of binarization techniques in classification is also proposed, and their effect is studied.

The best accuracies have been achieved with the $\text{matrix}_{\mathsf{intpc}}$ representation in both corpora, and even if the accuracy obtained with this representation does not improve the global_{12} representation for every classifier, its mean accuracy is better in both corpora. The statistical analysis has also shown that there are significant differences between the results obtained with $\text{matrix}_{\mathsf{intpc}}$ representation and the ones obtained with global_{12} representation.

The application of OVO binarization technique has proven beneficial to the classification accuracies in general, even if in this work the best accuracies were obtained with single classifiers. Its effects are more noticeable when it is used on corpus_5 with five possible target classes, where the results with OVO are better for almost all the classifiers. The results of the statistical test also show that there are statistical differences between global_{12} and $\text{matrix}_{\mathsf{intpc}}$ representations and between single classifiers and OVO classifications.

Considering that the use of interval based matrices have obtained promising accuracies, more complex viewpoints should be considered to build the matrices, to study how the classification can be improved with more complex information.

Acknowledgements. This work has been partially supported by the Basque Government Research Teams grant (IT900-16) and the Spanish Ministry of Economy and Competitiveness. TIN2015-64395-R (MINECO/FEDER).

References

1. Conklin, D.: Multiple viewpoint systems for music classification. J. New Music Res. **42**(1), 19–26 (2013)
2. Conklin, D., Witten, I.H.: Multiple viewpoint systems for music prediction. J. New Music Res. **24**, 51–73 (1995)
3. Dor, O., Reich, Y.: An evaluation of musical score characteristics for automatic classification of composers. Comput. Music J. **35**(3), 86–97 (2011)
4. Fürnkranz, J.: Round robin classification. J. Mach. Learn. Res. **2**, 721–747 (2002)
5. Galar, M., Fernández, A., Barrenechea, E., Bustince, H., Herrera, F.: An overview of ensemble methods for binary classifiers in multi-class problems: experimental study on one-vs-one and one-vs-all schemes. Pattern Recognit. **44**(8), 1761–1776 (2011)
6. Goienetxea, I., Martínez-Otzeta, J.M., Sierra, B., Mendialdua, I.: Towards the use of similarity distances to music genre classification: a comparative study. PLOS ONE **13**(2), 1–18 (2018)
7. Goienetxea, I., Neubarth, K., Conklin, D.: Melody classification with pattern covering. In: 9th International Workshop on Music and Machine Learning (MML 2016), Riva del Garda, Italy (2016)
8. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. SIGKDD Explor. Newsl. **11**(1), 10–18 (2009)
9. Herremans, D., Sörensen, K., Martens, D.: Classification and generation of composer-specific music using global feature models and variable neighborhood search. Comput. Music J. **39**(3), 71–91 (2015)
10. Hillewaere, R., Manderick, B., Conklin, D.: Global feature versus event models for folk song classification. In: Proceedings of the 10th International Society for Music Information Retrieval Conference, Kobe, Japan, pp. 729–733 (2009)
11. Hillewaere, R., Manderick, B., Conklin, D.: String methods for folk tune genre classification. In: Proceedings of the 13th International Society for Music Information Retrieval Conference, Porto, Portugal (2012)
12. van Kranenburg, P., Conklin, D.: A pattern mining approach to study a collection of Dutch folk-songs. In: Proceedings of the 5th International Workshop on Folk Music Analysis (FMA 2016), Dublin, pp. 71–73 (2016)
13. Mckay, C., Fujinaga, I.: jsymbolic: a feature extractor for midi files. In: Proceedings of the International Computer Music Conference, pp. 302–305 (2006)
14. Sapp, C.S.: Online database of scores in the humdrum file format. In: ISMIR 2005, Proceedings of 6th International Conference on Music Information Retrieval, 11–15 September 2005, London, UK, pp. 664–665 (2005)

Bibliography

[Ana+95] R. Anand, K. Mehrotra, C. K. Mohan, and S. Ranka. “Efficient Classification for Multiclass Problems Using Modular Neural Networks”. In: *Trans. Neur. Netw.* 6.1 (Jan. 1995), pp. 117–124 (cit. on p. 38).

[Ana97] C Anagnostopoulou. “Cohesion in Linguistic and Musical Discourse”. In: *Proceedings of the 3rd European Society for the Cognitive Sciences of Music Conference*. ESCOM. Uppsala, Sweden, 1997 (cit. on p. 10).

[Arr+14] A. Arruti, I. Mendialdua, B. Sierra, E. Lazkano, and E. Jauregi. “New One Versus AllOne method: NOV@”. In: *Expert Systems with Applications* 41.14 (2014), pp. 6251 –6260 (cit. on p. 40).

[AW04] M. Allan and C. K. I. Williams. “Harmonising chorales by probabilistic inference”. In: *Advances in Neural Information Processing Systems*. 2004, pp. 25–32 (cit. on p. 9).

[Ayr+02] Jay Ayres, Jason Flannick, Johannes Gehrke, and Tomi Yiu. “Sequential PAttern Mining Using a Bitmap Representation”. In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’02. Edmonton, Alberta, Canada: ACM, 2002, pp. 429–435 (cit. on pp. 22, 59).

[Bag+12] Mohammad Ali Bagheri, Qigang Gao, and Sergio Escalera. “Efficient Pairwise Classification Using Local Cross Off Strategy”. In: *Advances in Artificial Intelligence*. Ed. by Leila Kosseim and Diana Inkpen. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 25–36 (cit. on pp. 40, 41).

[Bas+15] Nikoletta Bassiou, Constantine Kotropoulos, and Anastasios Papazoglou-Chalikias. “Greek folk music classification into two genres using lyrics and audio via canonical correlation analysis”. In: *2015 9th International Symposium on Image and Signal Processing and Analysis (ISPA)*. IEEE. 2015, pp. 238–243 (cit. on p. 51).

[Bay50] Samuel P. Bayard. “Prolegomena to a Study of the Principal Melodic Families of British-American Folk Song”. In: *The Journal of American Folklore* 63.247 (1950), pp. 1–44 (cit. on p. 49).

[Ber+06] James Bergstra, Norman Casagrande, Dumitru Erhan, Douglas Eck, and Balázs Kégl. “Aggregate features and AdaBoost for music classification”. In: *Machine learning* 65.2-3 (2006), pp. 473–484 (cit. on p. 52).

[Bil07] John A. Biles. “Improvizing with Genetic Algorithms: GenJam”. In: *Evolutionary Computer Music*. Ed. by Eduardo Reck Miranda and John Al Biles. London: Springer London, 2007, pp. 137–169 (cit. on p. 11).

[Bim+12] Frédéric Bimbot, Emmanuel Deruty, Gabriel Sargent, and Emmanuel Vincent. "Semiotic structure labeling of music pieces: Concepts, methods and annotation conventions". In: *13th International Society for Music Information Retrieval Conference (ISMIR)*. Porto, Portugal, Oct. 2012, pp. 235–240 (cit. on p. 10).

[BL+12] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent. "Modeling Temporal Dependencies in High-Dimensional Sequences: Application to Polyphonic Music Generation and Transcription." In: *International Conference on Machine Learning*. Edinburgh, Scotland, 2012 (cit. on p. 12).

[BP17] Jean-Pierre Briot and François Pachet. "Music Generation by Deep Learning - Challenges and Directions". In: *CoRR* abs/1712.04371 (2017). arXiv: 1712.04371 (cit. on pp. 12, 13).

[Bre+16] Mason Bretan, Gil Weinberg, and Larry P. Heck. "A Unit Selection Methodology for Music Generation Using Deep Neural Networks". In: *CoRR* abs/1612.03789 (2016). arXiv: 1612.03789 (cit. on p. 13).

[Bri+17] Jean-Pierre Briot, Gaëtan Hadjeres, and François Pachet. "Deep Learning Techniques for Music Generation - A Survey". In: *CoRR* abs/1709.01620 (2017). arXiv: 1709.01620 (cit. on p. 12).

[Bro+56] F. P. Brooks, A. L. Hopkins Jr., P. G. Neumann, and W. V. Wright. "An experiment in musical composition". In: *IRE Transactions on Electronic Computers* EC–5 (1956), pp. 175–182 (cit. on p. 12).

[CA06] Darrell Conklin and Christina Anagnostopoulou. "Segmental Pattern Discovery in Music". In: *Informs Journal on Computing* 18 (3 2006), pp. 285–293 (cit. on p. 15).

[CB08] Darrell Conklin and Mathieu Bergeron. "Feature Set Patterns in Music". In: *Computer Music Journal* 32.1 (2008), pp. 60–70 (cit. on p. 22).

[Che04] M. Chemillier. "Toward a formal study of jazz chord sequences generated by Steedman's grammar". In: *Soft Computing* 8.9 (2004), pp. 617–622 (cit. on p. 11).

[Col+14] Tom Collins, Robin Laney, Alistair Willis, and Paul H. Garthwaite. "Developing and evaluating computational models of musical style". In: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* (2014) (cit. on pp. 12, 16, 28).

[Con13a] Darrell Conklin. "Fusion functions for multiple viewpoints". In: *5th International Workshop on Music and Machine Learning at ECML/PKDD 2013 (MML 2013)*. Prague, Czech Republic, 2013 (cit. on pp. 52, 62).

[Con13b] Darrell Conklin. "Multiple Viewpoint Systems for Music Classification". In: *Journal of New Music Research* 42.1 (2013), pp. 19–26 (cit. on p. 58).

[Con16] D. Conklin. "Chord sequence generation with semiotic patterns". In: *Journal of Mathematics and Music* 10.2 (2016), pp. 92–106 (cit. on p. 20).

[Cop04] D. Cope. *Virtual Music: Computer Synthesis of Musical Style*. The MIT Press. MIT Press, 2004 (cit. on p. 10).

[CV01] Wei Chai and Barry Vercoe. “Folk music classification using hidden Markov models”. In: *Proceedings of International Conference on Artificial Intelligence*. Vol. 6. 6.4. Citeseer. 2001 (cit. on p. 52).

[CW16] Darrell Conklin and Stéphanie Weisser. “Pattern and antipattern discovery in Ethiopian Bagana songs”. In: *Computational Music Analysis*. Ed. by D. Meredith. Springer, 2016, pp. 425–443 (cit. on p. 18).

[CW95] Darrell Conklin and Ian H. Witten. “Multiple Viewpoint Systems for Music Prediction”. In: *Journal of New Music Research* 24 (1995), pp. 51–73 (cit. on pp. 4, 17).

[DB95] Thomas G. Dietterich and Ghulum Bakiri. “Solving Multiclass Learning Problems via Error-correcting Output Codes”. In: *J. Artif. Int. Res.* 2.1 (Jan. 1995), pp. 263–286 (cit. on p. 38).

[Dem06] Janez Demšar. “Statistical Comparisons of Classifiers over Multiple Data Sets”. In: *J. Mach. Learn. Res.* 7 (Dec. 2006), pp. 1–30 (cit. on p. 36).

[Die98] Thomas G. Dietterich. “Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms”. In: *Neural Comput.* 10.7 (Oct. 1998), pp. 1895–1923 (cit. on p. 36).

[DKT17] Dua Dheeru and Efi Karra Taniskidou. *UCI Machine Learning Repository*. 2017 (cit. on p. 43).

[Don+18] Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang. “MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment”. In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. 2018, pp. 34–41 (cit. on p. 13).

[Dor+16] Matthias Dorfer, Andreas Arzt, and Gerhard Widmer. “Towards score following in sheet music images”. In: *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*. 2016 (cit. on p. 52).

[Dor95] J. Dorronsoro. *Bertso Doinutegia*. Euskal Herriko Bertsolari Elkartea, 1995 (cit. on p. 14).

[DR11] Ofer Dor and Yoram Reich. “An Evaluation of Musical Score Characteristics for Automatic Classification of Composers.” In: *Computer Music Journal* 35.3 (2011), pp. 86–97 (cit. on p. 53).

[Dub+03] Shlomo Dubnov, G. Assayag, O. Lartillot, and G. Bejerano. “Using machine-learning methods for musical style modeling”. In: *Computer* 36.10 (2003), pp. 73–80 (cit. on p. 9).

[Edw11] Michael Edwards. “Algorithmic Composition: Computational Thinking in Music”. In: *Communications of the ACM* 54.7 (July 2011), pp. 58–67 (cit. on p. 9).

[FÖ2a] Johannes Fürnkranz. “Round Robin Classification”. In: *J. Mach. Learn. Res.* 2 (Mar. 2002), pp. 721–747 (cit. on pp. 38, 39).

[FÖ2b] Johannes Fürnkranz. “Round Robin Classification”. In: *J. Mach. Learn. Res.* 2 (2002), pp. 721–747 (cit. on p. 65).

[FL06] B. Fei and Jinbai Liu. “Binary tree of SVM: a new fast multiclass training and classification algorithm”. In: *IEEE Transactions on Neural Networks* 17.3 (2006), pp. 696–704 (cit. on p. 40).

[Fri96] Jerome H. Friedman. *Another approach to polychotomous classification*. Tech. rep. Department of Statistics, Stanford University, 1996 (cit. on pp. 39, 45).

[FV13] Jose David Fernández and Francisco Vico. “AI Methods in Algorithmic Composition: A Comprehensive Survey”. In: *Journal of Artificial Intelligence Research* 48.1 (2013), pp. 513–582 (cit. on p. 9).

[Gal+13] Mikel Galar, Alberto Fernández, Edurne Barrenechea, Humberto Bustince, and Francisco Herrera. “Dynamic Classifier Selection for One-vs-One Strategy: Avoiding Non-competent Classifiers”. In: *Pattern Recogn.* 46.12 (2013), pp. 3412–3424 (cit. on pp. 40–42, 45–47).

[Gal+15] Mikel Galar, Alberto FernÃindez, Edurne Barrenechea, and Francisco Herrera. “DRCW-OVO: Distance-based relative competence weighting combination for One-vs-One strategy in multi-class problems”. In: *Pattern Recognition* 48.1 (2015), pp. 28 –42 (cit. on p. 40).

[Gar+10] Salvador García, Alberto Fernández, Julián Luengo, and Francisco Herrera. “Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power”. In: *Information Sciences* 180.10 (2010). Special Issue on Intelligent Distributed Information Systems, pp. 2044 –2064 (cit. on p. 36).

[GC15] Izaro Goienetxea and Darrell Conklin. “Transformation of a bertso melody with coherence”. In: *Proceedings of the 5th International Workshop on Folk Music Analysis (FMA 2015)*. Paris, France, 2015, pp. 56–58 (cit. on pp. 5, 16).

[GC18] Izaro Goienetxea and Darrell Conklin. “Melody Transformation with Semiotic Patterns”. In: *Music Technology with Swing*. Ed. by Mitsuko Aramaki, Matthew E. P. Davies, Richard Kronland-Martinet, and Sølvi Ystad. Cham: Springer International Publishing, 2018, pp. 477–488 (cit. on pp. 6, 22).

[Goi+16] Izaro Goienetxea, Kerstin Neubarth, and Darrell Conklin. “Melody classification with pattern covering”. In: *9th International Workshop on Music and Machine Learning (MML 2016)*. Riva del Garda, Italy, 2016 (cit. on pp. 7, 62).

[Goi+18a] Izaro Goienetxea, Iñigo Mendialdua, and Basilio Sierra. “On the Use of Matrix Based Representation to Deal with Automatic Composer Recognition”. In: *AI 2018: Advances in Artificial Intelligence - 31st Australasian Joint Conference, Wellington, New Zealand, December 11-14, 2018, Proceedings*. 2018, pp. 531–536 (cit. on pp. 7, 66).

[Goi+18b] Izaro Goienetxea, José María Martínez-Otzeta, Basilio Sierra, and Iñigo Mendialdua. “Towards the use of similarity distances to music genre classification: A comparative study”. In: *PLOS ONE* 13.2 (2018), pp. 1–18 (cit. on pp. 6, 57, 64).

[GPOB06] Nicolas Garcia-Pedrajas and Domingo Ortiz-Boyer. “Improving multiclass pattern recognition by the combination of two strategies”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 28.6 (2006), pp. 1001–1006 (cit. on p. 40).

[GR99] G. Giacinto and F. Roli. “Methods for dynamic classifier selection”. In: *Proceedings 10th International Conference on Image Analysis and Processing*. 1999, pp. 659–664 (cit. on p. 37).

[Gro16] Ryan Groves. “Automatic Melodic Reduction Using a Supervised Probabilistic Context-Free Grammar”. In: *Proceedings of the 17th International Society for Music Information Retrieval Conference, ISMIR*. 2016, pp. 775–781 (cit. on p. 15).

[Hal+09] Mark Hall, Eibe Frank, Geoffrey Holmes, et al. “The WEKA Data Mining Software: An Update”. In: *SIGKDD Explorations Newsletter* 11.1 (2009), pp. 10–18 (cit. on pp. 44, 65).

[Hed78] Stephen A. Hedges. “Dice Music in the Eighteenth Century”. In: *Music & Letters* 59.2 (1978), pp. 180–187 (cit. on p. 9).

[Hel+98] J. van Helden, B. André, and J. Collado-Vides. “Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies”. In: *Journal of Molecular Biology* 281.5 (1998), pp. 827 –842 (cit. on p. 18).

[Her+14] Dorien Herremans, Kenneth Sörensen, and Darrell Conklin. “Sampling the extrema from statistical models of music with variable neighbourhood search”. In: Athens, Greece: ICMC/SMC, 2014 (cit. on pp. 12, 26).

[Her+15] Dorien Herremans, Kenneth Sörensen, and David Martens. “Classification and Generation of Composer-Specific Music Using Global Feature Models and Variable Neighborhood Search”. In: *Computer Music Journal* 39.3 (2015), pp. 71–91 (cit. on pp. 53, 63, 64).

[Hil+14] Ruben Hillewaere, Bernard Manderick, and Darrell Conklin. “Alignment Methods for Folk Tune Classification”. In: *Data Analysis, Machine Learning and Knowledge Discovery*. Ed. by Myra Spiliopoulou, Lars Schmidt-Thieme, and Ruth Janning. Springer International Publishing, 2014, pp. 369–377 (cit. on pp. 51, 52, 62).

[Hol08] T. Holmes. *Electronic and Experimental Music: Technology, Music, and Culture*. Taylor & Francis, 2008 (cit. on p. 9).

[HT98] Trevor Hastie and Robert Tibshirani. “Classification by Pairwise Coupling”. In: *Proceedings of the 1997 Conference on Advances in Neural Information Processing Systems 10*. NIPS ’97. Denver, Colorado, USA: MIT Press, 1998, pp. 507–513 (cit. on p. 39).

[HV10] Eyke Hüllermeier and Stijn Vanderlooy. “Combining predictions in pairwise classification: An optimal adaptive voting strategy and its relation to weighted voting”. In: *Pattern Recognition* 43.1 (2010), pp. 128 –142 (cit. on p. 39).

[ID80] Ronald L. Iman and James M. Davenport. “Approximations of the critical region of the fbietkan statistic”. In: *Communications in Statistics - Theory and Methods* 9.6 (1980), pp. 571–595 (cit. on p. 36).

[JD88] Anil K. Jain and Richard C. Dubes. *Algorithms for Clustering Data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1988 (cit. on p. 56).

[Kar+16] Alexandros Karatzoglou, Balázs Hidasi, Domonkos Tikk, et al. “RecSys’ 16 Workshop on Deep Learning for Recommender Systems (DLRS)”. In: *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM. 2016, pp. 415–416 (cit. on p. 52).

[KB03] Jaepil Ko and Hyeran Byun. “Binary classifier fusion based on the basic decomposition methods”. In: *Proceedings of the 4th international conference on Multiple classifier systems*. Springer, 2003, pp. 146–155 (cit. on p. 40).

[KC16] P. van Kranenburg and D. Conklin. “A Pattern Mining Approach to Study a Collection of Dutch Folk-Songs”. In: *Proceedings of the 6th International Workshop on Folk Music Analysis (FMA 2016)*. Dublin, Ireland, 2016, pp. 71–73 (cit. on p. 58).

[Koh95] Ron Kohavi. “A Study of Cross-validation and Bootstrap for Accuracy Estimation and Model Selection”. In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*. IJCAI’95. Montreal, Quebec, Canada: Morgan Kaufmann Publishers Inc., 1995, pp. 1137–1143 (cit. on p. 36).

[Kot+13] Alexios Kotsifakos, Evangelos E Kotsifakos, Panagiotis Papapetrou, and Vassilis Athitsos. “Genre classification of symbolic music with SMBGT”. In: *Proceedings of the 6th International Conference on PErvasive Technologies Related to Assistive Environments*. ACM. 2013, p. 44 (cit. on p. 51).

[Kra+12] P. van Kranenburg, A. Volk, and F. Wiering. “On Identifying Folk Song Melodies Employing Recurring Motifs”. In: *Proceedings of the 12th International Conference on Music Perception and Cognition and the 8th Triennial Conference of the European Society for the Cognitive Sciences of Music*. Thessaloniki, 2012, pp. 1057–1062 (cit. on p. 52).

[Kra+13] P. van Kranenburg, A. Volk, and F. Wiering. “A comparison between global and local features for computational classification of folk song melodies”. In: *Journal of New Music Research* 42.1 (2013), pp. 1–18 (cit. on pp. 52, 61, 62).

[KU02] Boonserm Kijsirikul and Nitiwut Ussivakul. “Multiclass support vector machines using adaptive directed acyclic graph”. In: *Proceedings of the 2002 International Joint Conference on Neural Networks*. Vol. 1. IEEE. 2002, pp. 980–985 (cit. on p. 40).

[Leb+07] G. Lebrun, O. Lezoray, C. Charrier, and H. Cardot. “An EA Multi-model Selection for SVM Multiclass Schemes”. In: *Proceedings of the 9th International Work Conference on Artificial Neural Networks*. IWANN’07. San Sebastián, Spain: Springer-Verlag, 2007, pp. 260–267 (cit. on p. 40).

[Lee+15] Chang-Hsing Lee, Hwai-San Lin, and Ling-Hwei Chen. “Music classification using the bag of words model of modulation spectral features”. In: *2015 15th International Symposium on Communications and Information Technologies (ISCIT)*. IEEE. 2015, pp. 121–124 (cit. on p. 52).

[LF95] Jeremy Leach and John Fitch. “Nature, Music, and Algorithmic Composition”. In: *Computer Music Journal* 19.2 (1995), pp. 23–33 (cit. on p. 10).

[Lie03] Martina Liepert. “Topological Fields Chunking for German with SVM’s: Optimizing SVM-parameters with GA’s”. In: *Proceedings of the International Conference on Recent Advances in Natural Language Processing*. 2003 (cit. on p. 40).

[Liu+17] Hui Liu, Wengming Zheng, Gaopeng Sun, et al. “Action understanding based on a combination of one-versus-rest and one-versus-one multi-classification methods”. In: *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*. 2017, pp. 1–5 (cit. on p. 39).

[LM16] Corentin Louboutin and David Meredith. “Using general-purpose compression algorithms for music analysis”. In: *Journal of New Music Research* 45.1 (2016), pp. 1–16 (cit. on pp. 52, 62).

[Mal16] Stéphane Mallat. “Understanding deep convolutional networks”. In: *Phil. Trans. R. Soc. A* 374.2065 (2016), p. 20150203 (cit. on p. 52).

[ME05] Michael I Mandel and Dan Ellis. “Song-Level Features and Support Vector Machines for Music Classification.” In: *ISMIR*. Vol. 2005. 2005, pp. 594–599 (cit. on p. 52).

[Men+15] Iñigo Mendialdua, José María Martínez-Otzeta, Igor Rodriguez-Rodriguez, Txelo Ruiz-Vazquez, and Basilio Sierra. “Dynamic selection of the best base classifier in One versus One”. In: *Knowledge-Based Systems* 85 (2015), pp. 298 –306 (cit. on pp. 41, 70).

[Men+16] I. Mendialdua, G. Echegaray, I. Rodriguez, E. Lazkano, and B. Sierra. “Undirected cyclic graph based multiclass pair-wise classifier: Classifier number reduction maintaining accuracy”. In: *Neurocomputing* 171 (2016), pp. 1576 –1590 (cit. on p. 40).

[Mer+02] David Meredith, Kjell Lemström, and Geraint A. Wiggins. “Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music”. In: *Journal of New Music Research* 31.4 (2002), pp. 321–345 (cit. on p. 12).

[Mer14] David Meredith. “Using point-set compression to classify folk songs”. In: *Proceedings of the 4th International Workshop on Folk Music Analysis (FMA 2014)*. Istanbul, Turkey, 2014, pp. 29–35 (cit. on pp. 52, 62).

[Mey57] L. B. Meyer. “Meaning in Music and Information Theory”. In: *Journal of Aesthetics and Art Criticism* 15 (1957), pp. 412–424 (cit. on p. 10).

[Pat08] A.D. Patel. *Music, Language, and the Brain*. Oxford University Press, USA, 2008 (cit. on p. 10).

[PC00] François Pachet and Daniel Cazaly. “A Taxonomy of Musical Genres”. In: *Content-Based Multimedia Information Access Conference (RIAO) proceedings*. 2000 (cit. on p. 49).

[PC18] Victor Padilla and Darrell Conklin. “Generation of Two-Voice Imitative Counterpoint from Statistical Models”. In: *International Journal of Interactive Multimedia and Artificial Intelligence* 5.3 (2018), pp. 22–32 (cit. on p. 12).

[Pla+00a] John C. Platt, Nello Cristianini, and John Shawe-Taylor. “Large Margin DAGs for Multiclass Classification”. In: *Advances in Neural Information Processing Systems 12*. Ed. by S. A. Solla, T. K. Leen, and K. Müller. MIT Press, 2000, pp. 547–553 (cit. on p. 40).

[Pla+00b] John C Platt, Nello Cristianini, and John Shawe-Taylor. “Large margin DAGs for multiclass classification”. In: *Advances in neural information processing systems*. 2000, pp. 547–553 (cit. on p. 40).

[Raj+15] Arjun Raj Rajanna, Kamelia Aryafar, Ali Shokoufandeh, and Raymond Ptucha. “Deep Neural Networks: A Case Study for Music Genre Classification”. In: *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. IEEE. 2015, pp. 655–660 (cit. on p. 52).

[Sap05] Craig Stuart Sapp. “Online Database of Scores in the Humdrum File Format”. In: *ISMIR 2005, 6th International Conference on Music Information Retrieval, London, UK, 11-15 September 2005, Proceedings*. 2005, pp. 664–665 (cit. on p. 63).

[Sch+06] A. Schoenberg, P. Carpenter, and S. Neff. *The Musical Idea and the Logic, Technique, and Art of Its Presentation*. Indiana University Press, 2006 (cit. on p. 10).

[Sci+16] Marco Scirea, Julian Togelius, Peter Eklund, and Sebastian Risi. “MetaCompose: A Compositional Evolutionary Music Composer”. In: *Evolutionary and Biologically Inspired Music, Sound, Art and Design: 5th International Conference, EvoMUSART 2016, Porto, Portugal, March 30 – April 1, 2016, Proceedings*. Ed. by Colin Johnson, Vic Ciesielski, João Correia, and Penousal Machado. Cham: Springer International Publishing, 2016, pp. 202–217 (cit. on p. 12).

[Sie+11] B. Sierra, E. Lazkano, I. Irigoien, E. Jauregi, and I. Mendialdua. “K Nearest Neighbor Equality: Giving equal chance to all existing classes”. In: *Information Sciences* 181.23 (2011), pp. 5158 –5168 (cit. on p. 41).

[Sou+06] B. F. De Souza, A. C. p. l. f. De Carvalho, R. Calvo, and R. P. Ishii. “Multiclass SVM Model Selection Using Particle Swarm Optimization”. In: *2006 Sixth International Conference on Hybrid Intelligent Systems (HIS’06)*. 2006, pp. 31–31 (cit. on p. 40).

[Sze+09a] G Szepannek, B Bischl, and C Weihs. “On the combination of locally optimal pairwise classifiers”. In: *Engineering Applications of Artificial Intelligence* 22.1 (2009), pp. 79–85 (cit. on p. 40).

[Sze+09b] G. Szepannek, B. Bischl, and C. Weihs. “On the combination of locally optimal pairwise classifiers”. In: *Engineering Applications of Artificial Intelligence* 22.1 (2009), pp. 79 –85 (cit. on p. 45).

[Tsy+08] Alexey Tsymbal, Mykola Pechenizkiy, Pádraig Cunningham, and Seppo Puuronen. “Dynamic Integration of Classifiers for Handling Concept Drift”. In: *Inf. Fusion* 9.1 (Jan. 2008), pp. 56–68 (cit. on p. 37).

[Vel+13] Gissel Velarde, Tillman Weyde, and David Meredith. “Wavelet-filtering of symbolic music representations for folk tune segmentation and classification”. In: *Proceedings of the 3rd International Workshop on Folk Music Analysis (FMA 2013)*. Amsterdam, Netherlands, 2013, pp. 56–62 (cit. on pp. 52, 62).

[VK12] A. Volk and P. van Kranenburg. “Melodic similarity among folk songs: An annotation study on similarity-based categorization in music”. In: *Musicae Scientiae* 16.3 (2012), pp. 317–339 (cit. on p. 50).

[VR+14] Jorge Valverde-Rebaza, Aurea Soriano, Lilian Berton, Maria Cristina Ferreira de Oliveira, and Alneu de Andrade Lopes. “Music genre classification using traditional and relational approaches”. In: *Intelligent Systems (BRACIS), 2014 Brazilian Conference on*. IEEE. 2014, pp. 259–264 (cit. on p. 51).

[WC16] Raymond P. Whorley and Darrell Conklin. “Music Generation from Statistical Models of Harmony”. In: *Journal of New Music Research* 45.2 (2016), pp. 160–183 (cit. on pp. 12, 26).

[Wil92] Frank Wilcoxon. “Individual Comparisons by Ranking Methods”. In: *Breakthroughs in Statistics: Methodology and Distribution*. Ed. by Samuel Kotz and Norman L. Johnson. New York, NY: Springer New York, 1992, pp. 196–202 (cit. on p. 36).

[Woo+97] Kevin Woods, W. Philip Kegelmeyer Jr., and Kevin Bowyer. “Combination of Multiple Classifiers Using Local Accuracy Estimates”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 19.4 (Apr. 1997), pp. 405–410 (cit. on p. 37).

[Xen92] Iannis Xenakis. *Formalized music : thought and mathematics in composition / Iannis Xenakis*. English. Rev. ed. Pendragon Press Stuyvesant, NY, 1992, xiv, 387 p. : (cit. on p. 9).

[Xia+16] Hongshan Xiao, Zhi Xiao, and Yu Wang. “Ensemble Classification Based on Supervised Clustering for Credit Scoring”. In: *Appl. Soft Comput.* 43.C (June 2016), pp. 73–86 (cit. on p. 37).

[Yan+17] Li-Chia Yang, Szu-Yu Chou, and Yi-Hsuan Yang. “MidiNet: A Convolutional Generative Adversarial Network for Symbolic-domain Music Generation using 1D and 2D Conditions”. In: *CoRR* abs/1703.10847 (2017). arXiv: 1703.10847 (cit. on p. 13).

[Zbi99] Lawrence M. Zbikowski. “Musical Coherence, Motive, and Categorization”. In: *Music Perception* 17.1 (1999), pp. 5–42 (cit. on p. 10).

[Zel+11a] A. Zelaia, I. Alegria, O. Arregi, and B. Sierra. “A multiclass/multilabel document categorization system: Combining multiple classifiers in a reduced dimension”. In: *Applied Soft Computing* 11.8 (2011), pp. 4981 –4990 (cit. on p. 68).

[Zel+11b] Ana Zelaia, Iñaki Alegria, Olatz Arregi, and Basilio Sierra. “A multiclass/multilabel document categorization system: Combining multiple classifiers in a reduced dimension”. In: *Applied Soft Computing* 11.8 (2011), pp. 4981–4990 (cit. on p. 52).

[Zel+15] Ana Zelaia, Olatz Arregi, and Basilio Sierra. “Combining Singular Value Decomposition and a multi-classifier: A new approach to support coreference resolution”. In: *Engineering Applications of Artificial Intelligence* 46 (2015), pp. 279–286 (cit. on p. 52).

[Zha+14] Chiyuan Zhang, Georgios Evangelopoulos, Stephen Voinea, Lorenzo Rosasco, and Tomaso Poggio. “A deep representation for invariance and music classification”. In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2014, pp. 6984–6988 (cit. on p. 52).

[Zha+17] Zhong-Liang Zhang, Xing-Gang Luo, Salvador García, Jia-Fu Tang, and Francisco Herrera. “Exploring the effectiveness of dynamic ensemble selection in the one-versus-one scheme”. In: *Knowledge-Based Systems* 125 (2017), pp. 53 –63 (cit. on p. 41).

[Zho+17] Ligang Zhou, Qingyang Wang, and Hamido Fujita. “One versus one multi-class classification fusion using optimizing decision directed acyclic graph for predicting listing status of companies”. In: *Information Fusion* 36 (2017), pp. 80 –89 (cit. on p. 39).

[Jac87] Fred Lerdahl and Ray Jackendoff. “A Generative Theory of Tonal Music”. In: *Journal of Aesthetics and Art Criticism* 46.1 (1987), pp. 94–98 (cit. on p. 15).