# Finding Music Formal Concepts Consistent with Acoustic Similarity

Yoshiaki OKUBO

Faculty of Information Science and Technology, Hokkaido University
N-14 W-9, Sapporo 060-0814, JAPAN

**Abstract.** In this paper, we present a method of finding conceptual clusters of music objects based on Formal Concept Analysis.
A formal concept (FC) is defined as a pair of extent and intent which are sets of objects and terminological attributes commonly associated with the objects, respectively. Thus, an FC can be regarded as a conceptual cluster of similar objects for which its similarity can clearly be stated in terms of the intent. We especially discuss FCs in case of music objects, called music FCs.
Since a music FC is based solely on terminological information, we often find extracted FCs would not always be satisfiable from acoustic point of view. In order to improve their quality, we additionally require our FCs to be consistent with acoustic similarity. We design an efficient algorithm for extracting desirable music FCs. Our experimental results for *The MagnaTagATune Dataset* shows usefulness of the proposed method.

**Keywords:** formal concept analysis, music formal concepts, music objects, terminological similarity, acoustic similarity.

## 1 Introduction

*Clustering* has been well known as a fundamental task in *Data Analysis* and *Data Mining* [1]. In the decade, it is paid much attention as a representative of *Unsupervised Learning* in the field of *Machine Learning* [2].

The task of clustering is to find groupings, called *clusters*, of data objects given as a database, where each group consists of similar objects in some sense. Based on the clusters, we can overlook the database. If we find some of them interesting, we might intensively examine those attractive ones.

In this paper, we are concerned with a problem of clustering for *music objects*. Clustering plays important roles in many real applications of *Music Information Retrieval* (MIR) [3, 4]. A typical application would be music recommendation [5]. Several CF(collaboration filtering)-based methods for music recommendation have been proposed with the help of clustering techniques, e.g., see [6]. A clustering-based method for automatically creating playlists of music objects has been investigated in [7]. Clustering is also fundamental in visualizing our music collection [8].

Since clustering is a representative of unsupervised-tasks, we need to try to interpret obtained clusters by some means. It is, however, not always easy to have

David C. Wyld et al. (Eds): COMIT, SIPO, AISCA, MLIOB, BDHI - 2020
pp. 01-15, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.101601

adequate interpretations or explanations. It would be especially difficult in case of clusters of music objects because those objects are highly perceptual and thus not descriptive. Nevertheless, meaningful clusters would be preferable for many MIR tasks. For example, such clusters provides us a very informative and insightful overview of our music collection.

In this paper, we discuss a method of finding conceptual clusters of music objects. Particularly, we try to detect our clusters based on the notion of *formal concept*.

A formal concept (FC) is defined as a pair of *extent* and *intent* which are sets of objects and terminological attributes commonly associated with the objects, respectively. Thus, such an FC can be regarded as a conceptual cluster of similar objects for which its similarity can clearly be stated in terms of the intent.

Although music objects are usually given in some audio format such as MP3 and WAV, they are often provided linguistic information including playing artists, composers, genres, etc. Moreover, they would often be freely assigned user-tags by active users of popular music online services. Assuming such linguistic information as terminological attributes of music objects, therefore, we can extract *music FCs* from our music database.

It is noted that since a music FC is based only on linguistic information, we often find that extracted FCs would not always be satisfiable from acoustic point of view. In order to improve their quality, we formalize a problem of finding music FCs consistent with acoustic similarity and then design an efficient algorithm for extracting them.

Our experimental results for *The MagnaTagATune Dataset* [9] shows that we can efficiently detect satisfiable music clusters excluding many undesirable ones with acoustical inconsistency.

The remainder of this paper is organized as follows. The next section discusses previous work closely related to our framework. In Section 3, we introduce the fundamental notion of music FCs. We then formalize our problem of finding music formal concepts consistent with acoustic similarity in Section 4. An algorithm for the problem with a simple pruning rule is also presented. We show our experimental results in Section 5, discussing usefulness of our framework. Section 6 concludes the paper with a summary and future work.

## 2 Related Work

In the field of MIR, main approaches to processing music objects can generally be divided into two categories, *content-based* [3] and *context-based* [10] ones. In the former, each music object is represented by their intrinsic acoustic features extracted with the help of adequate signal processing techniques. In the latter, on the other hand, they are processed based on their external semantic features. Those features are often referred to as *metadata* which can be classified into three categories, editorial, cultural and acoustic metadata [11]. Although both content-based and context-based approaches have been separately investigated in traditional studies in MIR, effectiveness of combined approaches has been verified recently.

For the task of artistic style clustering, Wang et al. have argued that using both linguistic and acoustic information is a useful approach [12]. They have proposed a novel language model, called Tag+Content (TC) Model, in which style distribution of each artist can be related to each other by making use of both information, while standard topic language models impractically assume their independence.

Miotto and Orio have proposed a probabilistic retrieval framework in which content-based acoustic similarity and (pre-annotated) tags are combined together [13]. In the framework, a music collection is represented as a similarity graph, where each music is described by a set of tags. Then, the documents relevant for a given query are extracted as some paths in the graph most likely related to the request.

Knees et al. have extended a search engine for music objects in which contextual queries are accepted [14]. In order to improve quality of its text-based ranking, they have utilized audio-based similarity in the ranking schema.

Our framework proposed in this paper takes a similar approach in which both content-based and context-based information are effectively utilized. However, we have several characteristic points to be noted.

The clustering problem in [12] is *purpose-directed* in the sense that we have to designate in advance which kind of clusters we try to detect (e.g., artistic style clusters) and prepare our dataset suitable for the purpose. We, therefore, would not suffer from issues of interpretation for clustering results which is the main concern in our framework.

In [13], a retrieval result is obtained by finding plausible paths in a similarity graph. That is, we can find solutions by directly searching the graph. A similarity graph plays an important role also in our proposed method. However, our similarity graph cannot provide any solution directly. As is different from [13], it is used for just checking whether a candidate of our solution is acceptable or not. Our similarity graph prescribes an additional constraint our solutions must satisfy.

The main purpose of combining acoustic similarity in [14] is to improve ranking quality based solely on textual information. In other words, both acoustic and textual information are associatively utilized. On the other hand, those information are independently used in our framework. Based solely on our similarity graph, we strictly reject undesirable candidates of solutions.

In more general perspective, a dataset often comprises numerical and categorical features in many application domains. Such a dataset is called *mixed data*. Since clustering mixed data would be a challenging task, various clustering algorithms designed for mixed data have already been developed. The literature [15] provides an extensive survey of the state-of-the-art algorithms.

In the proposed framework, each music object is necessarily assumed to have its own linguistic information like annotation-tags. It would be an inevitable limitation of our method. As has been pointed out as *cold start problem* in recommendation systems, our method would suffer from the same kind of problem. In the field of MIR, importance of text-based information has been recognized and several approaches to obtaining such information for music objects have been investigated and compared [16–18]. Those approaches are surely helpful for our method.

## 3 Music Formal Concepts

In this section, we discuss a notion of *music formal concepts* with which we are concerned in this paper. We first introduce the basic terminology of *Formal Concept Analysis* [19, 20].

### 3.1 Formal Concept Analysis

Let $\mathcal{O}$ be a set of *data objects* (or simply *objects*) and $\mathcal{A}$ a set of *attributes*. For a binary relation $R \subseteq \mathcal{O} \times \mathcal{A}$, a *formal context* $\mathcal{C}$ is defined as a triple $\mathcal{C} = \langle \mathcal{O}, \mathcal{A}, R \rangle$, where for $(o, a) \in R$, we say that the object $o$ has the attribute $a$. For an object $o \in \mathcal{O}$, the set of attributes associated with $o$ is denoted by $o'$, that is,

$$o' = \{a \mid a \in \mathcal{A} \ \text{ and } \ (o, a) \in R\},$$

where "$\,'\,$" is called the *derivation operator*.

Similarly, for an attribute $a \in \mathcal{A}$, the set of objects having $a$ is also denoted by $a'$, that is,

$$a' = \{o \mid o \in \mathcal{O} \ \text{ and } \ (o, a) \in R\}.$$

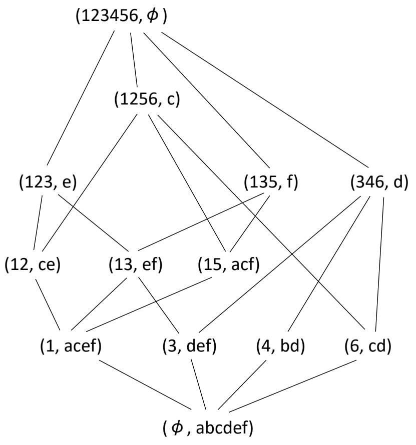
It is easy to extend the derivation operator for sets of objects and attributes. More precisely speaking, for a set of objects $O \subseteq \mathcal{O}$ and a set of attributes $A \subseteq \mathcal{A}$, we have $O' = \bigcap_{o \in O} o'$ and $A' = \bigcap_{a \in A} a'$, respectively.

For a set of objects $O$ and a set of attributes $A$, if and only if $O' = A$ and $A' = O$, then the pair $(O, A)$ is called a *formal concept* (or simply a *concept*) in the context $\mathcal{C}$ [19], where $O$ is called the *extent* and $A$ the *intent* of the concept.

It should be noted that a formal concept $(O, A)$ provides a clear interpretation of the extent and intent. The extent means that every object in $O$ shares all of the attributes in $A$. Moreover, the intent means there exists no object having every attribute in $A$ except for ones in $O$. In other words, the extent is regarded as a *cluster of similar objects* for which we can clearly state the reason why they are similar in terms of the intent.

For a formal context $\mathcal{C}$, we refer to the set of all formal concepts in $\mathcal{C}$ as $\mathcal{FC}_{\mathcal{C}}$. We here assume an ordering $\prec$ on $\mathcal{FC}_{\mathcal{C}}$ such that for any pair of concepts $FC_i = (O_i, A_i)$ and $FC_j = (O_j, A_j)$ in $\mathcal{FC}_{\mathcal{C}}$ $(i \neq j)$, $FC_i \prec FC_j$ if and only if $O_i \subset O_j$ (dually

$A_i \supset A_j$), where $FC_i$ is said to be more *specific* than $FC_j$ and conversely $FC_j$ more *general* than $FC_i$. Then, the ordered set $(\mathcal{FC}_{\mathcal{C}}, \prec)$ forms a lattice, called a *formal concept lattice.*



| object | attribute |
|---|---|
| 1 | a c e f |
| 2 | c e |
| 3 | d e f |
| 4 | b d |
| 5 | a c f |
| 6 | c d |

**Fig. 1.** Example of formal concept lattice

Figure 1 shows the formal concept lattice for a small example of formal context with the sets of objects and attributes, { 1, 2, 3, 4, 5, 6 } and { a, b, c, d, e, f }, respectively. In the figure, each concept is represented in a simplified form. For example, the concept ({ 1, 3 }, { e, f }) is abbreviated as (13, ef). Moreover, general concepts are placed on the upper side.

### 3.2 Music Formal Concept

In this paper, we assume that our music object owns two kinds of information, *audio signal-based information* and *linguistic information.*

For the former, a music object is usually represented (or stored) in a standard audio format like WAV and MP3. From those music objects, we can then extract some audio features, such as Mel-Frequency Cepstrum Coefficient (MFCC) and Chroma, with the help of useful techniques of signal processing.

On the other hand, for the latter, we expect that music objects are usually provided several linguistic labels including playing artists, composers, song writers, genres, etc. Furthermore, those objects would often be freely assigned user-tags by active users of popular music online services. In order to obtain formal concepts for music objects, therefore, we can consider a formal context whose attributes are based on the linguistic information.

Let $\mathcal{M}$ be a set of our music objects in some audio formats and $\mathcal{L}$ a vocabulary (a set of terms) to express linguistic information on $\mathcal{M}$, that is, we assume each music object in $\mathcal{M}$ is annotated with some of terms in $\mathcal{L}$. Then, we define our *music formal context* $\mathcal{MC}$ as $\mathcal{MC} = \langle \mathcal{M}, \mathcal{L}, R \rangle$, where $R \subseteq \mathcal{M} \times \mathcal{L}$ and $R = \{(m, t) \mid m \in \mathcal{M}, t \in \mathcal{L}, m \text{ is annotated with } t\}$. We can now extract formal concepts from $\mathcal{MC}$, called *music formal concepts*.

It is, in general, well known that we often find a large number of formal concepts in a given formal context. We actually have 13 concepts even in the small context shown in Figure 1. Needless to say, it would be quite impractical to examine all of them in order to obtain preferable ones. In some case, unfortunately, most of the extracted FCs would not be satisfiable to us.

In the next section, we try to improve quality of music FCs by taking acoustic information of music objects into account.

## 4 Finding Music Formal Concepts Consistent with Acoustic Feature Similarity

Since a music formal context is defined based on linguistic information, music FCs provides us clusters of similar music objects from linguistic point of view. This means that our music FCs would not always reflect acoustic similarity. As a result, we could often find many music FCs uncomfortable and unsatisfiable. In order to exclude such undesirable FCs, we additionally impose a constraint upon our target FCs to be extracted.

As has been mentioned above, we can usually extract several kinds of acoustic information from our music objects with useful techniques of signal processing. Since such an information is provided in a form of real-valued feature vectors, we assume each of our music objects has its own acoustic feature vector with dimension of $d$. For a music object $m_i \in \mathcal{M}$, its feature vector is referred to as $\boldsymbol{v}_i$.

Based on those feature vectors, we can now evaluate similarity between any two music objects from acoustic viewpoint. For music objects $m_i, m_j \in \mathcal{M}$, we calculate similarity between $m_i$ and $m_j$, denoted as $sim(m_i, m_j)$, by *Cosine Similarity* [21], that is,

$$sim(m_i, m_j) = \frac{\boldsymbol{v}_i \cdot \boldsymbol{v}_j}{|\boldsymbol{v}_i||\boldsymbol{v}_j|}. \tag{1}$$

In order to bring acoustic similarity of music objects in our target FCs, we take a graph-theoretic approach for efficient computation.

Assuming a threshold $\theta$ as the lower bound of similarity, we create a *similarity graph*, $G(\theta)$, for our music objects. It is formally defined as $G(\theta) = (\mathcal{M}, E(\theta))$, where

$$E(\theta) = \{(m_i, m_j) \mid m_i, m_j \in \mathcal{M}, i \neq j, sim(m_i, m_j) \geq \theta\}. \tag{2}$$

That is, any pair of music objects are connected by an edge if they have a certain degree of similarity with respect to their acoustic feature vectors.

It is easy to see from the definition that a clique in $G(\theta)$ gives a set of music objects pairwise similar. For a music FC, therefore, if we additionally require the extent to form a clique in $G(\theta)$, our FC can reflect acoustic similarity as well as linguistic one. In other words, by the additional requirement, we can exclude any music FC whose extent shows *inconsistency of acoustic similarity.* As the result, it would be expected that we can reasonably obtain more preferable FCs. In what follows, we refer to a music FC consistent with acoustic feature similarity as a music FC again.

We now formalize our problem of finding music FCs.

**Definition 1.** ***(Problem of Finding Music FCs)***
*Let $\mathcal{M}$ be a set of music objects, $\mathcal{L}$ a vocabulary annotating those objects, $\mathcal{MC} = \langle \mathcal{M}, \mathcal{L}, R \rangle$ a music formal context corresponding to the annotation and $G(\theta) = (\mathcal{M}, E(\theta))$ a similarity graph. Then, a* problem of finding music formal concepts *is to enumerate every formal concept $(M, L)$ in $\mathcal{MC}$ such that $M$ must be a clique in $G(\theta)$.* ■

An algorithm for the problem is presented below. We first provide our basic search strategy for computing ordinary FCs and then incorporate the additional requirement into our search process.

**Basic Search Strategy** Let $\mathcal{MC} = \langle \mathcal{M}, \mathcal{L}, R \rangle$ be a music formal context. We here assume some total ordering $\prec$ on $\mathcal{M}$ and for any subset $M \subseteq \mathcal{M}$, the objects in $M$ are always sorted in the ordering.

Based on $\prec$, the power set of $\mathcal{M}$, $2^{\mathcal{M}}$, can be arranged in a form of *set enumeration tree* [22], where the root node is $\emptyset$ and for a node $M$, a child of $M$ is defined as $M \cup \{m\}$ such that $tail(M) \prec m$, referring to the last object of $M$ as $tail(M)$.

It is easy from the definition to see that for each FC $(M, L)$ in $\mathcal{MC}$, we can always find a set of objects $X \subseteq \mathcal{M}$ such that $M = X''$ and $L = X'$. By traversing the set enumeration tree, thus, it is possible to meet every FC by computing $(X'', X')$ for an $X$ in the tree.

More concretely speaking, as a basic process, we try to expand a set of objects $X$ into $X \cup \{m\}$ with an object $m$ such that $tail(X) \prec m$. We then compute $((X \cup \{m\})'', (X \cup \{m\})')$ to obtain an FC. Such an object $m$ we try to add is called a *candidate* and is selected from the set of candidates, $cand(X)$, formally defined as

$$cand(X) = \{m \mid m \in (\mathcal{M} \setminus X'') \ \text{ and } \ tail(X) \prec m\}.$$

Initializing $X$ with $\emptyset$, we recursively iterate our expansion process in *depth-first manner* until no $X$ can be expanded.

It is noted that based on the ordering $\prec$, we can avoid a considerable number of duplicate generations of each individual FC.

More concretely speaking, when we expand $X$ with a candidate $m \in \mathit{cand}(X)$, if $(X \cup \{m\})'' \setminus X''$ includes some object $x$ such that $x \prec m$, then the FC $((X \cup \{m\})'', (X \cup \{m\})')$ and those obtained from any descendant of $X \cup \{m\}$ are completely useless because those concepts have already been obtained in our depth-first search. Therefore, we can immediately stop further expansions of $X \cup \{m\}$ and backtrack to the next candidate.

**Pruning Useless Music FCs** According to the basic strategy, we can surely extract every ordinary FC in $\mathcal{MC}$. Since our final goal is to find every FC whose extent must form a clique in $G(\theta)$, we incorporate the requirement into our search process.

As a simple observation, it is easy to see that any subset of a clique in $G(\theta)$ is also a clique. This implies that if a set of music objects $X \subseteq \mathcal{M}$ cannot form a clique in $G(\theta)$, any superset of $X$ can never be a clique. This observation brings us a simple pruning rule we can enjoy during our search process.

For an (ordinary) FC $MC$, if its extent does not form a clique, then any FCs succeeding to $MC$ in our depth-first search tree can safely be pruned as useless ones because their extents do not also form cliques and therefore can never be our target FCs. Whenever we find such a violation of the requirement, we can immediately stop our expansion process and then backtrack.

**Algorithm Description** We present a simple depth-first algorithm for finding our target music FCs. Its pseudo-code is shown in Figure 2.

In the figure, the head (first) element of a set $S$ is referred to as $\mathit{head}(S)$. Moreover, we refer to the original index of object $o$ in $\mathcal{O}_{\mathcal{MC}}$ as $\mathit{index}(o)$. The **if** statement at the beginning of **procedure** FCFind is for avoiding duplicate generations of the same FC and the **else if** for pruning useless expansions.

## 5 Experimental Results

In this section, we present our experimental results. We have implemented our algorithm for finding music formal concepts consistent with audio features and conducted several experimentations to verify its usefulness. Our system has been coded in C and executed on a PC with Intel® Core™ i5 (1.6 GHz) processor and 16 GB main memory.

```
[Input]   $\mathcal{MC} = \langle \mathcal{M}, \mathcal{L}, R \rangle$ : a music formal context
          $G$ : a similarity graph for $\mathcal{M}$ based on acoustic feature vectors
[Output]  $\mathcal{MFC}$ : the set of music formal concepts consistent with
                          acoustic feature similarity
procedure Main($\mathcal{MC}$, $G$) :
  $\mathcal{MFC} \leftarrow \emptyset$ ;
  Fix a total ordering $\prec$ on $\mathcal{M}$ ;
  $C \leftarrow \mathcal{M}$ ;
  while $C \neq \emptyset$ do
    begin
      $m \leftarrow head(C)$ ;
      $C \leftarrow (C \setminus \{m\})$ ;
      MusicFCFind($\{m\}$, $\emptyset$, $C$) ;
    end
  return $\mathcal{FC}$ ;

procedure MusicFCFind($X$, $PrevExt$, $Cand$) :
  $MFC \leftarrow (Ext = X'', X')$ ; // music FC
  if $\exists x \in (Ext \setminus PrevExt)$ such that $x \prec tail(X)$ then
    return; // discard duplicate music FC
  else if $Ext$ is not a clique in $G$ then
    return; // discard music FC violating cliqueness
  endif
  $\mathcal{MFC} \leftarrow \mathcal{MFC} \cup \{MFC\}$ ;
  while $Cand \neq \emptyset$ do
    begin
      $m \leftarrow head(Cand)$ ;
      $Cand \leftarrow Cand \setminus \{m\}$ ; // removing $m$ from $Cand$ ;
      $NewCand \leftarrow Cand \setminus PrevExt$ ; // new candidate objects.
      if $NewCand = \emptyset$ then continue ;
      MusicFCFind($X \cup \{m\}$, $Ext$, $NewCand$) ;
    end
```

**Fig. 2.** Algorithm for Finding Music Formal Concepts Consistent with Acoustic Feature Similarity

### 5.1 Dataset

In our experimentation, we have used "*The MagnaTagATune Dataset*" [9], a dataset publicly available [1].

The dataset contains 25,863 audio clips in MP3 format, where each of the clips has length of 30 seconds. The number of the original music works (titles) from which those clips have been extracted is 6385.

For most of the clips, two kinds of audio features, *pitch* and *timbre*, have already been provided in the dataset. More concretely speaking, for each audio clip, a couple of sequences (time-series) of 12-dimensional vectors have been prepared for both audio features.

[1] `http://mirg.city.ac.uk/codeapps/the-magnatagatune-dataset`

The dataset also contains annotation data for the audio clips. Each of the clips except for 4,221 has been annotated with several tags out of 188 possible ones.

### 5.2 Music Formal Context and Similarity Graphs

For preparation of our music formal context and similarity graphs for audio features, we have to select only audio clips from the dataset each of which is assigned at least one annotation tag and has its corresponding feature vectors. We have found 21,618 audio clips out of 25,863 satisfying the conditions.

Based on the selected 21,618 music audio clips and their annotation data, we have created our music formal context $\mathcal{MC} = \langle \mathcal{M}, \mathcal{A}, R \rangle$, where $\mathcal{M}$ is the set of 21,618 audio clips as our data objects and $\mathcal{A}$ the set of 188 possible annotation tags as our attributes. Furthermore, $R$ is defined as $R = \{(m, a) \mid m \in \mathcal{M}, a \in \mathcal{A}, m \text{ is annotated with } a\}$.

Our similarity graphs for audio features have also been created from the selected 21,618 audio clips and their audio feature vectors. As has been stated above, each audio clip has its corresponding two time-series of 12-dimensional feature vectors for pitch and timbre. As standard processing for (music) audio data, we average each dimension of time-series to get a single feature vector. Moreover, we also compute standard deviation of each dimension. Thus, for each audio clip $m_i \in \mathcal{M}$, we can obtain four single 12-dimensional vectors, $\boldsymbol{v}_i^{p\text{-}avg}$, $\boldsymbol{v}_i^{p\text{-}std}$, $\boldsymbol{v}_i^{t\text{-}avg}$ and $\boldsymbol{v}_i^{t\text{-}std}$, for averaged pitch, standard deviation of pitch, averaged timbre and standard deviation of timbre, respectively.

Assuming $\mathcal{M}$ as the set of vertices, given a threshold $\theta$ for similarity of audio features, our similarity graph for averaged pitch, denoted by $G^{p\text{-}avg}(\theta)$, has been constructed as $G^{p\text{-}avg}(\theta) = (\mathcal{M}, E^{p\text{-}avg}(\theta))$, where $E^{p\text{-}avg}(\theta)$ is defined with vectors $\boldsymbol{v}_i^{p\text{-}avg}$ according to the equations (1) and (2). As similarity graphs for standard deviation of pitch, averaged timbre and standard deviation of timbre, we can construct $G^{p\text{-}std}(\theta)$, $G^{t\text{-}avg}(\theta)$ and $G^{t\text{-}std}(\theta)$, respectively, in the same manner.

We have set $\theta$ to each value in the range from 0.9 to 1.0 with a step of 0.01.

### 5.3 Examples of Music Formal Concepts

We present here two music formal concepts. One is an example of our target FCs actually extracted by the proposed system and the other a negative example rejected due to inconsistency of acoustic similarity.

In Figure 3(a), we present a music FC actually found as accepted one. The FC satisfies the requirement of acoustic similarity based on standard deviation of pitch, where $\theta$ has been set to 0.95.

The extent consists of 6 music objects all of which are annotated with (at least) the 6 tags in the intent, where each object is expressed in the form of
"*Artist-AlbumTitle-TrackNum-TackTitle.*" Listening to those music objects, it is

| | |
|---|---|
| Extent | 1. zilla-egg-07-rufus |
| | 2. aba_structure-tektonik_illusion-03-pipe |
| | 3. magnatune_compilation-electronica-10-introspekt_mekhanix |
| | 4. hoxman-synthesis_of_five-11-nighty_girl |
| | 5. strojovna_07-iii-04-loopatchka |
| | 6. strojovna_07-Number_1-05-bycygel |
| Intent | fast drums techno synth funky upbeat |

(a) Accepted Music FC

| | |
|---|---|
| Extent | 1. saros-soundscapes-03-symphony_of_force |
| | 2. dj_markitos-evolution_of_the_mind-01-sunset_endless_night_journey_remix |
| | 3. burning_babylon-knives_to_the_treble-12-double_axe |
| | 4. belief_systems-eponyms-05-talk_box |
| | 5. hands_upon_black_earth-hands_upon_black_earth-11-priest |
| Intent | techno synth trance bass |

(b) Rejected Music FC

**Fig. 3.** Examples of Music FCs

found the concept provides a nice cluster in which they are certainly similar acoustically and have a clear interpretation given by the intent.

On the other hand, as a negative example, Figure 3(b) shows a music FC rejected by our algorithm due to inconsistency of any acoustic similarity provided for our experimentation. For the concept, each music object of the extent is surely annotated with all tags in the intent. Listening their audio samples, however, we would have an impression that the cluster given by the concept seems slightly ambiguous as a homogeneous group. For example, the music 2 of *DJ MARKITO* is a typical techno sound with clear beat of high tempo, while the music 5 of *Hands Upon Black Earth* is a illusional sound of synthesizers with no beat. With the help of acoustic similarity, such an undesirable cluster (FC) can be excluded in our framework.

### 5.4 Computational Performance

We here discuss computational performance of the proposed system. Concretely speaking, we have executed our system for each of the constructed graphs and observed computation times and numbers of extracted music FCs.

Figure 4 shows behavior of computation times for extracting music FCs consistent with acoustic similarity given by $G^{p\text{-}avg}(\theta)$, $G^{p\text{-}std}(\theta)$, $G^{t\text{-}avg}(\theta)$ and $G^{t\text{-}std}(\theta)$, respectively. In the figure, for example, the performance curve referred to as `t-std` is for $G^{t\text{-}std}(\theta)$ with each value of $\theta$. In order to see effectiveness of incorporating acoustic similarity, we have also put a dotted line, referred to as `NoSim`, corresponding to the performance curve in case without the additional requirement.

It is clearly stated that the requirement of acoustic similarity effectively improves efficiency of our computation. This means that the pruning based on the requirement can work well in our search.

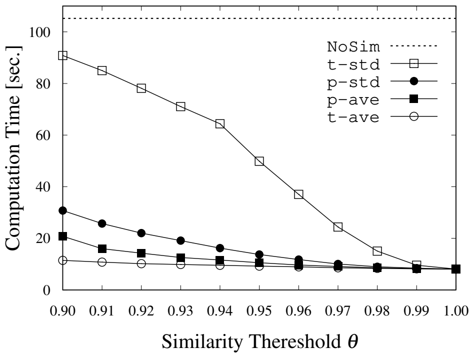


**Fig. 4.** Computation Times (sec)

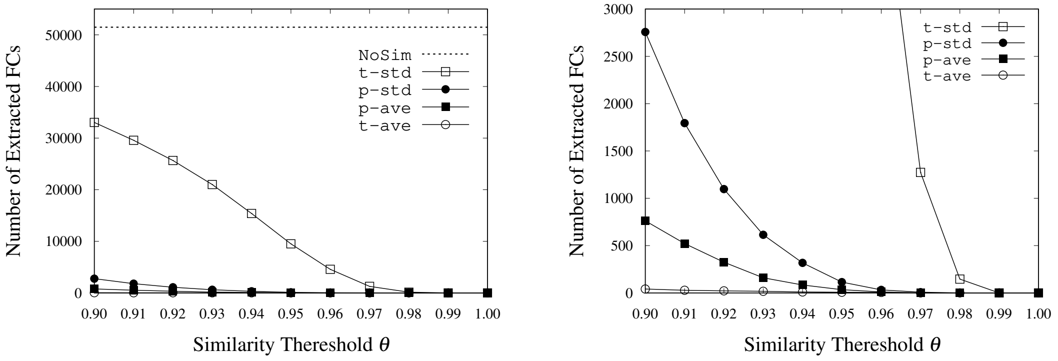
We can enjoy sufficient degree of improvement as the value of $\theta$ becomes larger (requiring stronger acoustic similarity) even in case of $G^{t\text{-}std}(\theta)$, that is, acoustic similarity based on standard deviation of timbre vectors. At the setting of $\theta$ =0.95 whose corresponding angle is about 18 degree, we get reductions of at least 50 % in any case and thus reasonable computation times.

Figure 5(a) shows how effectively the requirement of acoustic similarity can reduce numbers of music FCs to be extracted. It is easy to see that the behavior is almost the same as one in case of computation times. As has been discussed, we can completely discard non-target FCs by detecting just a small part of them defining a boundary between target and non-target in our search. Therefore, computation time of our algorithm is mainly spent for detecting target FCs consistent with acoustic similarity.

In case of `t-std`, although numbers of extracted FCs are surely reduced compared to that in case of `NoSim`, they still seems too large to actually examine them. As is mainly focused on the other three cases in Figure 5(b), we can obtain reasonable numbers of music FCs in case of `p-std` and `p-avg`, and very small numbers of those in case of `t-avg`. Thus, our requirements for acoustic similarity based on timber feature vectors bring us undesirable effects from practical point of view.

### 5.5 Discussion

As has been observed, the requirement for acoustic similarity can certainly reduce computation times and numbers of FCs to be extracted. Needless to say, degree of reductions is directly affected by the threshold $\theta$ adjusted in our construction process of similarity graphs. Although larger values of $\theta$ would bring us drastic reductions still keeping high homogeneity, we often find few music FCs satisfying such a severe requirement. Moreover, if we fortunately detect some FCs for larger



(a) Overview in Entire Range of $\theta$ (b) Details in Lower-Range

**Fig. 5.** Number of Extracted Music Formal Concepts

$\theta$, they would not be interesting for us in the sense that such an FC tends to include many music objects by the same artist or in the same album [2]. Therefore, we have to carefully set $\theta$ to an adequate value. At the moment, we have just an empirical instruction that values around 0.95 would be reasonable from practical viewpoint.

As an extended approach, users can flexibly adjust values of $\theta$ with the help of *user-interaction* so that they can intensively and deeply examine music FCs particularly interesting for them. At an early stage, we try to extract music FCs by setting $\theta$ to a (relatively) small value. In such a setting, since the requirement for acoustic similarity is not severe, it is easy to imagine that we have a large number of FCs. Obviously, showing users all of them is quite impractical. In order to have an overview of our music database, it would be reasonable to present *maximally general* FCs (that is, ones with maximally larger extents) which have a very small part of them. Browsing those maximal FCs, users would mark several promising candidates to further examine. For an increased value of $\theta$, we can again extract music FCs with finer homogeneity and then intensively find ones related to the candidates previously marked.

## 6 Concluding Remarks

In this paper, we discussed a method of finding music formal concepts. Those concepts correspond to meaningful clusters of music objects in the sense that each cluster can clearly be interpreted in terms of its intent and consists of objects acoustically similar. We presented a depth-first algorithm for efficiently extracting music FCs with a simple pruning rule. In our experimentations, we observed use-

[2] In case where several music objects are clipped from a single track, as *The MagnaTagATune Dataset*, we could find most of the objects in an FC are from the same track.

fulness of the proposed method from the viewpoints of quality of extracted FCs and computational efficiency.

Since our current framework assumes that each music object is assigned its own linguistic information like annotation-tags, we have to cope with issues such as cold start problems in recommendation systems. It would be worth incorporating some mechanism of automatic-tagging/labeling into our current system.

The proposed method is a general framework applicable to any domain in which data objects can be represented in numerical vectors and assigned their own linguistic information. Based on the current framework, we can design and develop useful recommendation systems in various application domains.

## References

1. L. Billard ad E. Diday. *Symbolic Data Analysis*, Wiley, 2006.
2. S. Marsland, *Machine Learning: An Algorithmic Perspective, Second Edition*, CRC Press, 2015.
3. Y. V. S. Murthy and S. G. Koolagudi. Content-Based Music Information Retrieval (CB-MIR) and Its Applications toward the Music Industry: A Review, *ACM Computing Surveys*, 51(3), Article 45, 2018.
4. T. Li, M. Ogihara and G. Tzanetakis (eds.). *Music Data Mining*, CRC Press, 2012.
5. D. Paul and S. Kundu. A Survey of Music Recommendation Systems with a Proposed Music Recommendation System, *Emerging Technology in Modelling and Graphics*, AISC-937, pp. 279 – 285, Springer, 2020.
6. Y. Song, S. Dixon and M. Pearce. A Survey of Music Recommendation Systems and Future Perspectives, In *Proc. of the 9th Int'l Symp. on Computer Music Modeling and Retrieval - CMMR'12*, pp. 395 – 410, 2012.
7. D. Lin and S. Jayarathna. Automated Playlist Generation from Personal Music Libraries, In *Proc. of 2018 IEEE Int'l Conf. on Information Reuse and Integration for Data Science*, pp. 217 – 224, 2018.
8. F. Mörchen, A. Ultsch, M. Nöcker and C. Samm. Visual Mining in Music Collections, *From Data and Information Analysis to Knowledge Engineering*, M. Spiliopoulou, R. Kruse, C. Borgelt, A. Nürnberger and W. Gaul (eds.), pp. 724 – 731, Springer, 2006.
9. E. Law, K. West, M. Mandel, M. Bay and J. S. Downie. Evaluation of Algorithms Using Games: The Case of Music Tagging, In *Proc. of the 10th Int'l Conf. on Music Information Retrieval - ISMIR'09*, pp. 387 – 392, 2009.
10. P. Knees and M. Schedl. A Survey of Music Similarity and Recommendation from Music Context Data, *ACM Transactions on Multimedia Computing, Communication and Applications*, 10(1), Article 2, 2013.
11. F. Pachet. Knowledge Management and Musical Metadata, In *Encyclopedia of Knowledge Management*, 2005.
12. D. Wang, T. Li and M. Ogihara. Are Tags Better Than Audio Features? The Effect of Joint Use of Tags and Audio Content Features for Artistic Style Clustering, In *Proc. of the 11th Int'l Society for Music Information Retrieval Conference - ISMIR'10*, pp. 57 – 62, 2010.
13. R. Miotto and N. Orio. A Probabilistic Model to Combine Tags and Acoustic Similarity for Music Retrieval, *ACM Transactions on Information Systems*, 30(2), Article 8, 2012.
14. P. Knees, T. Pohle, M. Schedl, D. Schnitzer, K. Seyerlehner and G. Widmer. Augmenting Text-Based Music Retrieval with Audio Similarity, In *Proc. of the 10th Int'l Society for Music Information Retrieval Conference - ISMIR'09*, pp. 579 – 584, 2009.

15. A. Ahmad and S. S. Khan. Survey of State-of-the-Art Mixed Data Clustering Algorithms, *IEEE Access*, 7, pp. 31883 – 31902, 2019.
16. K. M. Ibrahim, J. Royo-Letelier, E. V. Epure, G. Peeters and G. Richard. Audio-Based Auto-Tagging With Contextual Tags for Music, In *Proc. of 2020 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP 2020*, pp. 16 – 20, 2020.
17. L. Kumar A. Mitra, M. Mittal, V. Sanghvi, S. Roy and S. K. Setua. Music Tagging and Similarity Analysis for Recommendation System, *Computational Intelligence in Pattern Recognition*, AISC-999, pp 477 – 485, Springer, 2020.
18. D. Turnbull, L. Barrington and G. Lanckriet. Five Approaches to Collecting Tags for Music, In *Proc. of the 9th Int'l Society for Music Information Retrieval Conference - ISMIR'08*, pp. 225 – 230, 2008.
19. B. Ganter and R. Wille. *Formal Concept Analysis – Mathematical Foundations*, 284 pages, Springer, 1999.
20. B. Ganter, G. Stumme and R. Wille (Eds). *Formal Concept Analysis – Foundations and Applications*, LNAI-3626, 348 pages, Springer, 2005.
21. P. Knees and M. Schedl. *Music Similarity and Retrieval*, Springer, 2016.
22. R. Rymon. Search through Systematic Set Enumeration, In *Proc. of Int'l Conf. on Principles of Knowledge Representation Reasoning - KR'92*, pp. 539 – 550, 1992.