Web Intelligence and Agent Systems: An International Journal 11 (2013) 149–170
DOI 10.3233/WIA-130268
IOS Press

Multi-label automatic indexing of music by cascade classifiers

Wenxin Jiang [a] and Zbigniew W. Ras [b,c,*]
[a] *Fred Hutchinson Cancer Research Center, Seattle, WA 98109, USA*
E-mail: wjiang2@fhcrc.org
[b] *Department of Computer Science, University of North Carolina, Charlotte, NC 28223, USA*
[c] *Institute of Computer Science, Warsaw University of Technology, 00-665 Warsaw, Poland*
E-mail: ras@uncc.edu

Abstract. Recently, numerous successful approaches have been developed for instrument recognition in monophonic sounds. Unfortunately, none of them can be successfully applied to polyphonic sounds. Identification of music instruments in polyphonic sounds is still difficult and challenging. This has stimulated a number of research projects on music sound separation, new features development, and more recently on hierarchically structured classifiers used in content-based music recommender systems. This paper introduces a hierarchically structured cascade classification system to estimate multiple timbre information from the polyphonic sound by classification which is based on acoustic features and short-term power spectrum matching. This cascade system makes a first estimate on the higher level decision attribute which stands for the musical instrument family. Then, the further estimation is done within that specific family range. Experiments showed better performance of a hierarchical system than the traditional flat classification method which directly estimates the instrument without higher level of family information analysis. Traditional hierarchical structures were constructed in human semantics, which are meaningful from human perspective but not appropriate for a cascade system. We introduce a new hierarchical instrument schema according to the clustering results of the acoustic features. This new schema better describes the similarity among different instruments or among different playing techniques of the same instrument. The classification results show the higher accuracy of cascade system with the new schema compared to the traditional schemas.

Keywords: Music information retrieval, automatic indexing, knowledge discovery, classifiers, clustering

1. Introduction

This paper presents the methodology behind the construction of system-driven hierarchical classifiers forming the kernel of our web-based storage and recommender system for music retrieval, called *MIRAI* [28]. It can automatically index musical input (of polyphonic type) into FS-tree type database and answer queries requesting specific musical pieces [http://www.mir.uncc.edu/]. When *MIRAI* receives a musical waveform, it divides that waveform into segments of equal size and then its classifiers identify the most dominating musical instruments and emotions associated with that segment. Our database of musical instrument sounds is constantly growing and it currently has more than 4,000 sound objects representing 59 music instruments and about 1,100 features. Each sound object is represented as a temporal sequence of approximately 150–300 tuples which gives us a temporal database of more than 1,000,000 tuples, each one represented as a vector of about 1,100 features. This database is mainly used to learn classifiers for automatic indexing of musical instrument sounds and the same for a construction of recommender systems for polyphonic music retrieval.

In our previous research, we have already shown that automatic indexing systems based on hierarchical type of classifiers, which are extracted from *MIRAI* database of musical instrument sounds, usually outper-

[*] Corresponding author.

form similar systems based on sound separation techniques (especially when many instruments are playing at the same time) [28]. This research presents further enhancement of *MIRAI* by the introduction and employment of cascade, system-driven hierarchical classifiers.

A variety of methods and classifiers have been applied in musical instrument estimation domain [4,7, 8,11,13,19,20,27,42]. Still, in a classifier based approach, it is a nontrivial problem to choose the one with a optimal performance in terms of estimation accuracy for most western orchestral instruments. A common practice is to try different classifiers on the same training database and select the one which yields the highest confidence. Then, the selected classifier is used for the timbre estimation on analyzed music sounds. There are boosting systems [9,10] consisting of a set of weak classifiers which iteratively are adding them to a final strong classifier. Boosting systems usually achieve a better estimation model by training each given classifier on a different set of samples from the training database, which keeps the same number of features or attributes. In other words, a boosting system assumes that there is a big difference among different group of subsets of the training database so that different classifiers are trained on the corresponding subset based on their expertise. However, due to the homogeneous characteristics across all data samples in a training database, musical data usually could not take full advantage of such panel of learners because none of the given classifiers would get a majority weight. Thus the improvement cannot be achieved by such a combination of classifiers.

Martin and Kim [27] employed the KNN algorithm to a hierarchical classification system with 31 features extracted from Correlograms which is a three-dimensional representation of the signal. With a database of 1023 sounds they achieved 87% of successful classifications at the family level and 61% at the instrument level when no hierarchy was used. The accuracy at the instrument level was increased to 79% by using the hierarchical procedure but it degraded the performance at the family level (79%). Without including the hierarchical procedure performance figures were lower than the ones they obtained with a Bayesian classifier. The fact that the best accuracy figures are around 80% and that Martin have settled into similar figures can be interpreted as an estimation of the limitations of the KNN algorithm (provided that the feature selection has been optimized with genetic or other kind of techniques). Therefore, more powerful techniques should be explored. Bayes Decision Rules and Naive Bayes classifiers are simple probabilistic classifiers by which the probabilities for the classes and the conditional probabilities for a given feature and a given class are estimated based on their frequencies over the training data. They are based on probability models that incorporate strong independence assumptions, which often have no bearing in reality, hence are naive. The resultant rule is formed by counting the frequency of various data instances, and can be used then to classify each new instance. Brown [4] applied this technique to Mel-Cepstral Coefficients by K-means clustering algorithm and a set of Gaussian mixture models. Each model was used to estimate the probabilities that a coefficient belongs to a cluster. Probabilities of all coefficients were then multiplied together and were used to perform the likelihood ratio test. It then classified 27 short sounds of oboe and 31 short sounds of sax with an accuracy rate of 85% for oboe and 92% for sax. Neural networks process information with a large number of highly interconnected processing neurons working in parallel to solve a specific problem. Neural networks learn by example. Cosi [5] developed a timbre classification system based on auditory processing and Kohonen self-organizing neural networks. Data were preprocessed by peripheral auditory transformations to extract perception features, then were fed to the network to build the map, and finally were compared in clusters with human subjective similarity judgments. In the system, nodes were used to represent clusters of the input spaces. The map was used to generalize similarity criteria even to vectors not utilized during the training phase. All 12 instruments in the test were quite well distinguished by the map. Hidden Markov Model is a statistical model by which the extracted model parameters can be used to do sensitive database searching. In a regular Markov model, the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters. A hidden Markov model adds outputs: each state has a probability distribution over the possible output tokens. This technique has been successfully applied in speech recognition [21] and natural language processing [6]. Paulus and Virtanen [31] developed a system with this technique for automatic transcription of drum instruments from polyphonic music signals. A background model with only one state is created for each instrument to describe the sound when the target instrument is played. Since drum transcription requires a good temporal resolution, the length of the analysis frame is 24 ms with 75% overlap between consecutive frames, leading into temporal resolution of 6 ms. The most likely model sequence of sound presence and absence is determined by decoding the signal on the location where the instrument is hit, and the second models all the other parts of the signal. The feature distributions in each state are modeled with Gaussian-mixture models (GMMs). Three types of instruments have been evaluated in their experiments: kick drum, snare drum, and hi-hats. Total average classification rate was from 44% to 59.7%. The drawbacks of the system include modeling in the location of a hit with a fixed context length instead of with a sound properties oriented context length, and limitation of features used in the experiment. This technique was used to deduce the most useful attributes in classifying sounds and to compare different resultant sound classes by different attributes. Binary Tree is a data structure in which each node contains one parent and not more than 2 children. It has been pervasively used in classification and pattern recognition research. Binary Trees are constructed top-down with the most informative attributes as roots to minimize entropy. Jensen and Arnspang [14] proposed an adapted Binary Tree with real-valued attributes for instrument classification regardless of pitch of the instrument in the sample. Various classifiers for a small number of instruments have been used in musical instrument estimation domain in the literature; yet it is a non-trivial problem to choose the one with optimal performance in terms of estimation rate for most western orchestral instruments. It is common to apply the different classifiers on the training data based on a specific group of features extracted from raw audio files and get the winner with the highest confidence for the testing music sounds. However, different instruments have different acoustic characters and they usually need different features to model the classifiers. In this paper, we will address that issue.

In many cases, the speed of classification is an important issue. It is computationally expensive to classify audio signals of CD quality based on the raw spectrum. To achieve the applicable classification time while preserving high classification accuracy, we introduce a cascade classifier which could further improve the instruments recognition of a Music Information Retrieval (MIR) system.

Cascade classifiers have been investigated in the domain of handwritten digit recognition. Thabtah [3] used filter-and-refine processes and combined them with KNN to give the rough but fast classification with lower dimensionality of features at filter step and to rematch the objects marked by the previous filter with the higher accuracy by increasing dimensionality of features. Also, Lienhart [22] used CART trees as base classifiers to build a boosted cascade of simple feature classifiers to achieve rapid object detection.

To our best knowledge, not much work has been done in investigating the applicability and usefulness of cascade classifiers in MIR area [16,22]. However, it is possible to construct a simple instrument family classifier with a low false recognition rate, which is called a classification pre-filter. When classification pre-filter assigns a specific family label to a musical frame, further classification of that frame is based on classifiers trained only on samples representing this specific family (samples representing other families are immediately discarded). Since the number of training samples is reduced, the computational complexity is reduced while the recognition rate still remains high.

Due to the existence of different characteristics for different features, we introduce a new method applicable to the music domain, which is to train different classifiers on different feature sets instead of different data samples. For instance, both MFCC and harmonic peaks are composed of serial real values, which are in the form of numeric vectors and therefore work well with KNN instead of Decision tree. On the other hand, features such as zero crossing, spectrum centroid, roll-off, attack-time and so on, are acoustic features in the form of single values, which could be combined to produce better rules after applied with decision tree or Bayes Decision Rules.

2. Timbre relevant features

We use audio data from the system MIRAI [28] which contains more than 4000 sounds mostly taken from the MUMS (McGill University Master Samples). The descriptions of these sounds are in terms of standard musical features which definitions are provided by MPEG7, in terms of non-MPEG7 features used earlier by other researchers, and in terms of new temporal features proposed in our system MIRAI (http://www.mir.uncc.edu). The audio data represent the following instruments: alto flute, bach trumpet, bass clarinet, bassoon, bass trombone, Bb trumpet, b-flat clarinet, cello, bowed cello, cello martele, cello mute, cello pizzicato, contrabass clarinet, contrabassoon, crotales, ctrumpet, ctrumpet harmonStemOut, bowed double bass, double bass martele, double bass mute, double bass pizzicato, e-flat clarinet, electric bass, electric guitar, English horn, flute, French horn, French Horn mute, glockenspiel, marimba crescendo, marimba single stroke, oboe, piano-9ft, piano-hamburg, piccolo, piccolo flutter, soprano saxophone, tenor saxophone, steel drums, tenor trombone, tenor trombone mute, tuba, tubular bells, bowed vibraphone, vibraphone hard mallet, bowed viola, viola martele, viola mute, viola natural, viola pizzicato, violin artificial, bowed violin, violin ensemble, violin mute, violin natural harmonics, xylophone.

Currently in MIRAI we use a set of acoustical and statistical features to describe sound and timbre. The features we extract are a mixture of MPEG-7 features, as well as non-MPEG temporal features. The spectrum features have two different frequency domains: Hz frequency and Mel frequency. Frame size was carefully designed to be 120ms, so that the 0th octave G (the lowest pitch in our audio database) can be detected. The hop size is 40ms with a overlapping of 80ms. Since the sample frequency of all the music objects is 44,100Hz, the frame size is 5292. A hamming window was applied to all STFT transforms to avoid jittering in the spectrum. These features are briefly described in the subsections below. We use our own software, developed for MIRAI project (see: http://www.mir.uncc.edu/), to extract all these features.

2.1. MPEG7 features

Below are the MPEG-7 standard audio features, both in the frequency domain and in the time domain [29].

- Spectrum Centroid: Describes the center-of-gravity of a log-frequency power spectrum. It economically indicates the pre-dominant frequency range.
- Spectrum Spread: The Root of Mean Square value of the deviation of the Log frequency power spectrum with respect to the gravity center in a frame. Like Spectrum Centroid, it is an economic way to describe the shape of the power spectrum.
- Spectrum Basis Functions: These are used to reduce the dimensionality by projecting the spectrum from high dimensional space to low dimensional space with compact salient statistical information.
- Spectrum Projection Functions: A vector to represent a reduced feature set by the projection against a reduced rank basis.
- Harmonic Peaks: A sequence of local peaks of harmonics of each frame.
- Log Attack Time: This feature is defined as the decimal logarithm of the time duration from the time instant when the signal starts to the time when it reaches its maximum value, or when it reaches its sustained part, whichever comes first.

2.2. Non-MPEG7 features

- SpecCentroid: Calculated as Harmonic Spectral Centroid from MPEG-7, representing power-weighted average of the frequency of the bins in the linear power spectrum, and averaged over all the frames of the steady state of the sound.
- Zero crossing: Counts the number of times that the signal sample data changes signs in a frame [38].
- Roll-off: averaged (over all frames) frequency below which an experimentally chosen percentage of the accumulated magnitudes of the spectrum is concentrated.
- Spectral Flux: This is used to describe the spectral rate of the signal. It is computed by the total difference between the magnitude of the FFT points in a frame and its successive frame [38].
- Mel frequency cepstral coefficients (MFCC): These coefficients describe the spectrum according to the human perception system in the Mel scale. They are computed by grouping the STFT points of each frame into a set of 40 coefficients by a set of 40 weighting curves with logarithmic transform and a discrete cosine transform (DCT) [24]. We use the MFCC functions from the Julius software toolkit [1].
- Tristimulus: Describes the ratio of the energy of 3 groups of harmonic partials to the total energy of harmonic partials. The following groups are used: fundamental, medium partials (2, 3, and 4) and higher partials (the rest) [33].

3. Hierarchical structure of decision attribute

It is easier for a person to tell the difference between violin and piano than violin and viola. Because violin and piano belong to different instrument families, then they have quite different timbre qualities. Violin and viola fall into the same instrument family which indicates that they share a similar timbre quality. If we can build the classifiers both on the family level and the instrument level, the polyphonic music sound is first classified at the instrument family level. After a

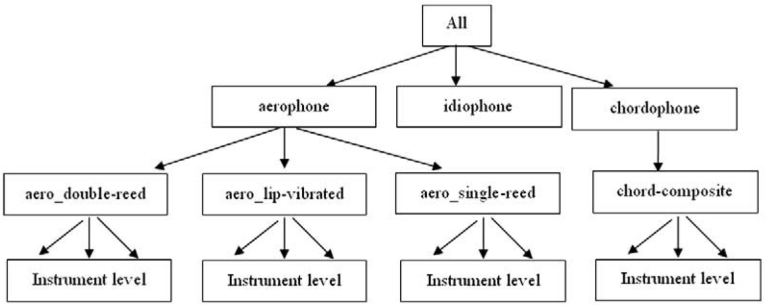

Fig. 1. Hornbostel-Sachs hierarchical structure.

specific instrument family label is assigned to the analyzed sound by the classifier, it is further classified at the instrument level by another classifier which is built on the training data of that specific instrument family. Since there are fewer instruments in this family, the classifier learned from this family has the expertise of identifying the instruments within this family. Before we discuss how to build classifiers on the different levels, let us first look at the hierarchical structure of the western instruments.

Erich von Hornbostel and Curt Sachs published an extensive scheme for musical instrument classification in 1914. Their scheme is widely used today, and is most often known as the Hornbostel-Sachs system. This system includes aerophones (wind instruments), chordophones (string instruments), idiophones (made of solid, non-stretchable, resonant material), and membranophones (mainly drums). Idiophones and membranophones are together considered as percussion. Additional groups include electrophones, i.e. instruments where the acoustical vibrations are produced by electric or electronic means (electric guitars, keyboards, synthesizers), complex mechanical instruments (including pianos, organs, and other mechanical music makers), and special instruments (include bullroarers, but they can be classified as free aerophones).

Idiophones subcategories include: Struck (claves, clapper, castanets, and finger cymbals). **Membranophones** include the following different kind of drums: Cylindrical drums, Conical drum, Barrel drum, Hourglass drum, Goblet drum, Footed drum, Long drum, Kettle or pot drum, Frame drums, and Friction drums. **Chordophones** include: zither, mandolins, guitars, ukuleles, Lute (bowed) – viols (fretted neck), fiddles, violin, viola, cello, double bass, and hurdy-gurdy (no frets), Harp. **Aerophones** are classified as single reed (such as clarinet, saxophones), double reed (such as oboe, bassoon) and lip vibrated (trumpet or horn) according to the mouthpiece used to set air in motion to produce sound. Some of the Aerophones subcategories are also called woodwinds or brass, but this criterion is not based on the material the instrument is made of, but rather on the method of sound production. In woodwinds, the change of pitch is mainly obtained by the change of the length of the column of the vibrating air. Additionally, over-blow is applied to obtain the second, third or fourth harmonic to become the fundamental. In brass instruments, over-blows are very easy because of wide bell and narrow pipe, and therefore over-blows are the main method of pitch changing. Sounds can be also classified according to the **articulation**. It can be performed in three ways: (1) sustained or non-sustained sounds, (2) muted or not muted sounds, (3) vibrated and not vibrated sounds. This classification may be difficult, since the vibration may not appear in the entire sound; some changes may be visible, but no clear vibration. Also, brass is sometimes played with moving the mute in and out of the bell. Most of musical instrument sounds of definite pitch have some noises/continuity in their spectra. According to MPEG7 classification [14], there are four classes of musical instrument sounds: (1) Harmonic, sustained, coherent sounds – well detailed in MPEG7, (2) Non-harmonic, sustained, coherent sounds, (3) Percussive, non-sustained sounds – well detailed in MPEG7, (4) Non-coherent, sustained sounds.

Figure 1 shows the simplified Hornbostel/Sachs tree. We do not include membranophones here because the instruments of this family usually do not produce the harmonic sound so that they need special tech-

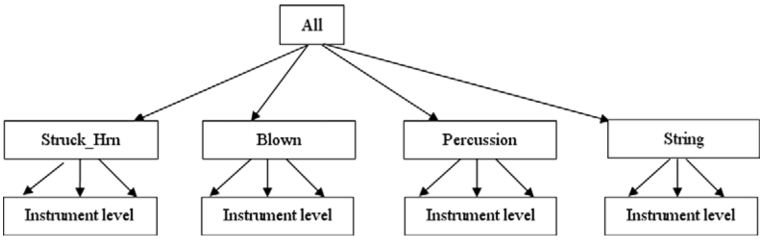

Fig. 2. Hierarchical structure according to playing method.

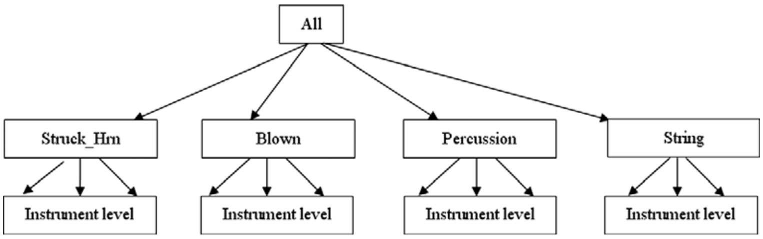

Fig. 3. Hierarchical decision attributes.

niques to be identified. This paper focuses on the harmonic instruments which fall into the other three families.

Figure 2 shows us another hierarchical structure of instrument family which is grouped by the way how the musical instruments are played.

In [43], a multi-hierarchical decision system S with a large number of descriptors built for describing music sound objects is described. The decision attributes in S are hierarchical and they include Hornbostel-Sachs classification and classification of instruments with respect to a playing method. The information richness hidden in these descriptors has strong implication on the confidence of classifiers built from S and used as a tool by the content-based Automatic Indexing Systems (AIS). Because decision attributes are hierarchical, the indexing and timbre estimation can be done with respect to different granularity levels of music instrument classes. We can then identify not only the instruments playing in a given music piece but also classes of instruments. In this paper we propose a methodology of building cascade classifiers from musical datasets.

4. Cascade hierarchical decision systems

In hierarchical decision systems, the initial group of classifiers is trained using all objects in an information system S partitioned by values of the decision attribute d at all granularity levels (one classifier per level). Only values of the highest granularity level (corresponding granules are the largest) are used to split S into information sub-systems where each one is built by selecting objects in S of the same decision value. These sub-systems are used for training new classifiers at all granularity levels of its decision attribute. Next, we split each sub-system further by sub-values of its decision value. The obtained tree-type structure with groups of classifiers assigned to each of its nodes is called a cascade classifier.

Let $S(d) = (X, A \cup \{d\}, V)$ be a decision system as introduced in [16,32], where X is a set of musical objects, A is the set of features used as classification attributes, d is a hierarchical decision attribute, V_d is a set of values of the decision attribute, and V_A is a set of values of all classification attributes. Figure 3 shows an example of a hierarchical decision attribute.

Let $V_d = \{d[1], d[2], \ldots, d[k]\}$ is a set of all values of the attribute d at level 1 of its tree representation. Let $X_i = \{x \in X : d(x) = d[i]\}$ and $S_i[d_i] = (X_i, A \cup \{d[i]\}, V)$, for any $1 \leq i \leq n$. Now, assume that $CR(S)$ denotes a tree of height one. System S is its root and $S_i(d[i])$, $(1 \leq i \leq n)$, are its children. The outgoing edge from S to $S_i(d[i])$ is labeled by $d[i]$, for any $1 \leq i \leq n$.

Cascade representation of $S(d)$ is defined as a tree with a root $S(d)$ and all its descendants being built by executing the instruction [if $card(V_d) > 1$, then replace $S(d)$ by $CR(S(d))$] recursively, starting from the root and then repeating for all leaves of a constructed tree.

Table 1
Example of hierarchical decision attribute

	a	b	c	d
x_1	2	1	2	$d[1,1]$
x_2	2	1	3	$d[1,1]$
x_3	1	1	0	$d[1,2]$
x_4	1	1	3	$d[1,2]$
x_5	2	2	2	$d[2,1]$
x_6	2	2	3	$d[2,1]$
x_7	1	1	1	$d[1,1]$
x_8	1	1	1	$d[1,1]$
x_9	2	2	1	$d[2,1]$
x_{10}	2	2	0	$d[2,1]$
x_{11}	1	1	2	$d[2,2]$
x_{12}	1	1	1	$d[2,2]$

5. Example

Let us look at the example of a decision system $S(d)$ represented as Table 1. Its attributes are $\{a,b,c\}$. d is the decision attribute.

To build a cascade representation of $S(d)$, we take its subsystems:

$$\begin{aligned}
&S^*(d) = (\{x_i : 1 \leq i \leq 12\}, \{a,b,c,d\}, V),\\
&S_{[1]}(d[1]) = (\{x_i : i = 1,2,3,4,7,8\},\\
&\qquad \{a,b,c,d\}, V),\\
&S_{[2]}(d[2]) = (\{x_i : i = 5,6,9,10,11,12\},\\
&\qquad \{a,b,c,d\}, V),\\
&S_{[1,1]}(d[1,1]) = (\{x_i : i = 1,2,7,8\},\\
&\qquad \{a,b,c,d\}, V),\\
&S_{[1,2]}(d[1,2]) = (\{x_i : i = 3,4\}, \{a,b,c,d\}, V),\\
&S_{[2,1]}(d[2,1]) = (\{x_i : i = 5,6,9,10\},\\
&\qquad \{a,b,c,d\}, V),\\
&S_{[2,2]}(d[2,2]) = (\{x_i : i = 11,12\}, \{a,b,c,d\}, V).
\end{aligned}$$

Now, the corresponding cascade representation of $S(d)$, denoted as $(\{S^*(d)\} \cup \{S_k(d) : k \in J\}, \prec)$, where $J = \{[1],[2],[1,1],[1,2],[2,1],[2,2]\}$ and "$\prec$" means parent-child relation, is represented by a tree similar to Fig. 4.

The partition of objects in $S(d)$ can be driven by an optimization function or it can be predefined, as it is done in MIRAI [30], by following either Hornbostel-Sachs classification or classification of instruments with respect to a playing method.

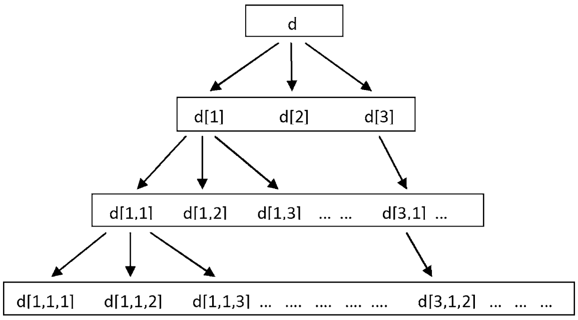

Fig. 4. Cascade representation of $S(d)$.

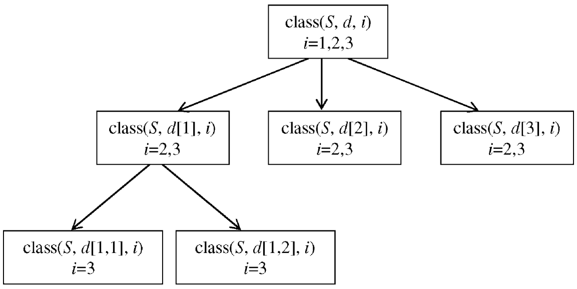

Fig. 5. Cascade classifier for $S(d)$.

6. Cascade classifiers

Let $S(d) = (X, F \cup \{d\})$ be a decision system, where d is a hierarchical attribute. We follow the notation of the previous section to represent its values, with $d[i]$ referring to a child of d and $d[i,j]$ to its grandchild. $F = \{f_1, \ldots, f_m\}$ is the set of all available features which are extracted from the input signal and then used by the classifiers respectively to identify the analyzed frame. $X = \{x_1, \ldots, x_t\}$ is the set of all segmented frames from the analyzed audio sound. $Casc(S(d)) = \{S_k(d) : k \in J\}$ is a cascade representation of $S(d)$, where J is the index set of all nodes of the hierarchical tree, such as [1], $[1,1]$, $[1,2]$ and so on. A sample representation structure for a cascade classifier is given in Fig. 5.

In this sample, three classifiers are associated with the root level of the tree. The first one (with $i = 1$) is trained by S with values of the decision attribute interpreted as the largest granules (largest generalizations of instruments). The last one (with $i = 3$) is based on decision attribute values interpreted as the smallest granules (single instruments). For each frame x_i, the whole process is started by the classification at the root of hierarchical tree and followed by the classification at other lower levels of the tree. The system selects the appropriate classifier $class(S_{[k]}, d, i)$ and feature

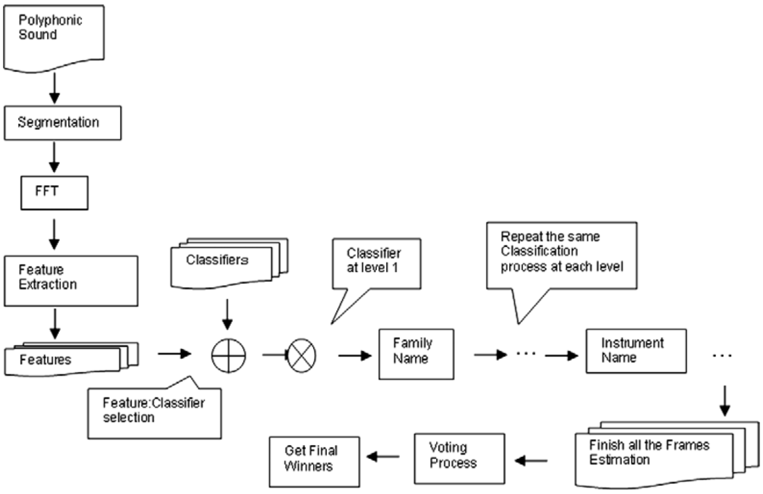

Fig. 6. Timbre estimation framework based on the cascade hierarchical classification system with classifier and feature selection.

Table 2
Cascade classifier for $S(d)$

Root	Classname	Classifier	Support	Confidence
d	All-instruments	Class($S, d, 2$)	771	96.97%
d	All-instruments	Class($S, d, 1$)	764	96.02%
d	All-instruments	Class($S, d, 3$)	730	91.80%
$d[1]$	Aerophone	Class($S, d[1], 2$)	269	98.26%
$d[1]$	Aerophone	Class($S, d[1], 3$)	265	96.84%
$d[2]$	Chordophone	Class($S, d[2], 2$)	497	98.83%
$d[2]$	Chordophone	Class($S, d[2], 3$)	466	92.75%
$d[3]$	Idiophone	Class($S, d[3], 2$)	19	95.95%
$d[3]$	Idiophone	Class($S, d[3], 3$)	19	95.95%
$d[1,1]$	Aero-double-reed	Class($S, d[1,1], 3$)	70	98.94%
$d[1,2]$	Aero-lip-reed	Class($S, d[1,2], 3$)	113	95.66%
$d[1,3]$	Aero-side	Class($S, d[1,3], 3$)	10	90.91%
$d[1,4]$	Aero-single-reed	Class($S, d[1,4], 3$)	72	99.54%
$d[2,1]$	Chord-composite	Class($S, d[2,1], 3$)	410	93.18%

$f(S_{[k]}, d, i)$ to perform classification at each possible level $S[k]$ from the top to the bottom. The confidence of classification at each level is $conf(x_i, S_{[k]})$ which has to be either equal or above the confidence threshold λ_1. Otherwise, the classification process fails. After the classification process reaches the bottom level, which is the instrument level, we have the final instrument estimations $\{d_p\}$ for the frame x_i. The overall confidence of each instrument identified in the frame x_i is calculated by multiplying the confidence obtained at each node $conf(x_i, d_p) = \prod conf(x_i, S_{[k]})$ leading to that instrument. After all the individual frames are estimated by the classification system, a smoothing process is performed by calculating the average confidence of each possible instrument within the indexing window $Conf(d_p) = \frac{\sum\{conf(x_i,d_p):i\in w\}}{s}$, where w is the set of indexes of all frames in a window and s is the number of frames in w. For each window, we use 120ms as the frame size and 40ms as the hop size. Only instruments satisfying the confidence threshold λ_2 are listed in the final result within the indexing window. Figure 6 shows the MIR timbre estimation framework based on the cascade hierarchical classification system.

7. Experiments and results based on all features

We build a hierarchical decision system S with all acoustic features described earlier in this paper which are also listed in Table 3. The decision attributes in S

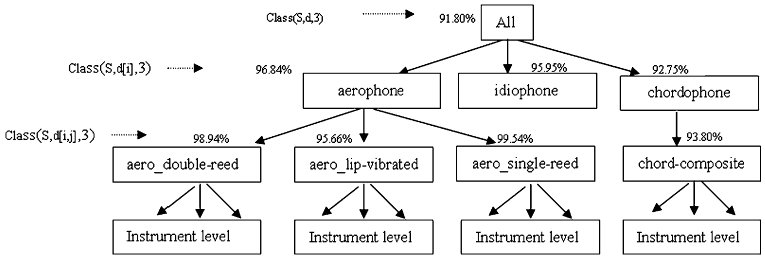

Fig. 7. Non-cascade classifier for Hornbostel/Sachs classification of instruments and their accuracy.

Table 3
Feature group

Group	Feature description
A	33 Flatness Coefficients
B	13 MFCC Coefficients
C	28 Harmonic Peaks
D	38 Spectrum projection coefficients
E	Log spectral centroid, spread, flux, rolloff, zerocrossing

are hierarchical and they represent Hornbostel-Sachs classification. The information richness hidden in descriptors has strong implication on the confidence of classifiers built from the decision system S. Hierarchical decision attributes allow us to have the indexing done on different granularity levels of classes of musical instruments. We can identify not only the instruments but also classes of instruments if the instrument level identification fails. We show that cascade classifiers outperform standard classifiers. Table 2 represents the cascade classifier for Hombostel-Sachs classification of instruments and its confidence in identifying them.

The testing was done for musical instrument sounds of pitch B3. The results in Table 2 show the confidence of the classifiers trained on different subsets which correspond to the different nodes in the hierarchical tree. The decision attributes of these classifiers are at the instrument level.

Figure 7 shows the confidence of the classifiers trained on the whole dataset (the largest granule). These classifiers describe values of the decision attribute at its different granularity levels represented by nodes of the tree. The confidence of a standard classifier $class(S, d, 3)$ for Hombostel-Sachs classification of instruments is 91.80%. However, we get better results by following the cascade approach. When we use the classifier $class(S, d, 2)$ followed by the classifier $class(S, d[1, 1], 3)$, the precision in recognizing musical instruments in "aero double reed" class is equal to $96.02\% * 98.94\% = 95.00\%$. Also, its confidence in recognizing instruments in "aero single reed" class is equal to $96.02\% * 99.54\% = 95.57\%$. It has to be noted that this improvement in classification confidence is obtained without increasing the number of attributes in the subsystems of S.

Again, from Table 2 and Fig. 7, when we compare different classifiers which are built from the same training dataset but on different granularity levels of decision attribute, we find that generic classifiers usually have the higher confidence than the peculiar one (Fig. 8).

Following this strategy, we get high classification confidence for single instrument estimation in comparison to a regular non-cascade classification approach.

8. Classifier selection based on different features

Cascade classification system allows different classifiers and different features to be used at different levels of the hierarchical structure. In order to investigate how the classifier and feature selection affect the cascade system, two experiments of classification based on KNN and Decision Tree were conducted: 1) classification with each feature group; 2) classification with the combination of different feature groups.

The training dataset of middle C includes 26 different instruments: Electric Guitar, Bassoon, Oboe, B-flat clarinet, Marimba, C-Trumpet, E-flat Clarinet, Tenor Trombone, French horn, Flute, Viola, Violin, English horn, Vibraphone, Accordion, Electric Bass, Cello, Tenor saxophone, B-Flat Trumpet, Bass flute, Double bass, Alto flute, Piano, Bach trumpet, Tuba, and Bass Clarinet.

These instruments cover the typical western musical instruments families which are played by the orchestra. There are 2762 frames extracted from those instrument sound objects. We tested different features with different classifiers to get the optimal pair of them. The results of all the experiments in this paper are generated by 10-fold cross validation

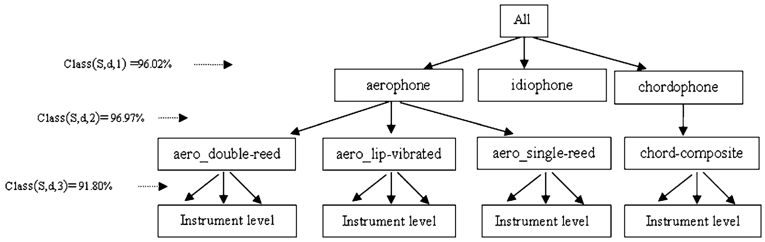

Fig. 8. Confidence of classifiers built on different levels of decision attribute (pitch 3B).

Table 4

Classification of each feature group

Feature group	Classifier	Confidence (%)
A	KNN	99.23
A	Decision Tree	94.69
B	KNN	98.19
B	Decision Tree	93.57
C	KNN	86.60
C	Decision Tree	91.29
D	KNN	47.45
D	Decision Tree	31.81
E	KNN	99.34
E	Decision Tree	99.77

8.1. Classification on each feature group

In experiment 1, we divided the features into the following 5 groups (Table 3).

Among the groups A to D, each represents one single feature vector of multiple numeric values. Group E includes all the statistical single-value features. Classifiers of KNN and Decision Tree are applied to the dataset with each feature group. For decision tree classifier, we choose J48 in Weka [6], which is the implementation of C4.5 decision tree algorithm [29]. The confidence factor used for pruning tree (smaller values incur more pruning) is 0.25. The minimum number of instances per leaf is 10. For K-nearest neighbor classifier [9], we have chosen IBK in Weka [6], which is the brute force search algorithm for nearest neighbor search. The number of neighbors is 3. Euclidean distance is used as similarity function. Confidence is defined as the ratio of the correct classified instances over the total number of instances.

The result in Table 4 shows that some features work better with KNN than decision tree, such as Flatness Coefficients (Group A), MFCC (Group B), and spectrum projection coefficients (Group D), Decision tree works better with harmonic peaks (Group C). The statistical features (Group E) show little difference between the two classifiers.

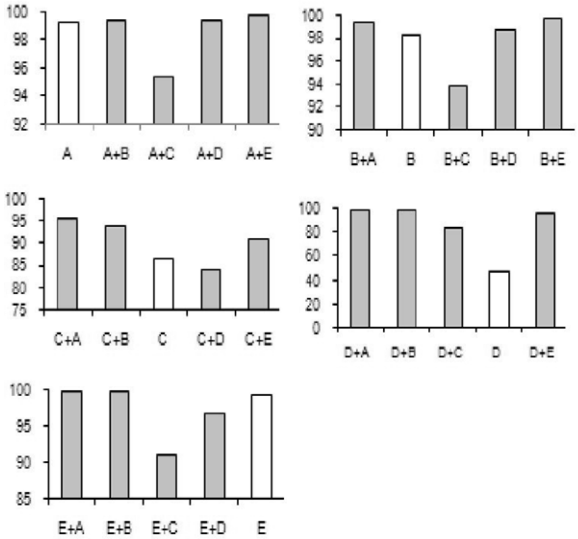

Fig. 9. Classification based on KNN in experiment 2.

8.2. Classification on the combination of different feature groups

In order to further explore the relationship between feature groups and classifiers, we merge every two feature groups into larger feature groups and test them with different classifiers. Figure 9 shows the result of KNN classification.

The confidence of KNN changes minimally when more features are added. And when Group C (harmonic-Peaks) is added to Group A, B, D, and E, the classification results deteriorate. This result further validates the conclusion from Experiment 1 that harmonic peaks do not fit KNN classifier well.

Figure 10 shows the result of decision tree classification. We observe that group E improves other feature groups when it is added. However those results do not improve much compared to the classification result of single Group E. It means Group E yields the best result for decision tree classifier.

From those two experiments, we see that KNN classifier works better with feature vectors such as

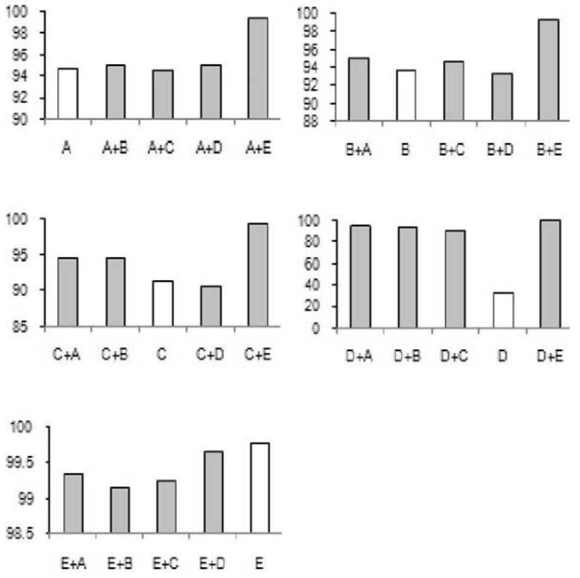

Fig. 10. Classification of decision tree in experiment 2.

spectral flatness coefficients, projection coefficients and MFCC. Decision tree works better with harmonic peaks and statistical features. Simply adding more features together does not improve the classifiers and sometime even worsens the classification results (such as adding harmonic to other feature groups). In cascade system, it is a non-trivial task to perform feature selection for different classifiers to optimize the timbre estimation.

9. Feature and classifier selection at each level of cascade system

According to the previous discussion and conclusion, cascade system has to select the appropriate feature and classifier to achieve the best estimation result at each level of cascade classification. We test four feature groups (A, B, C, D) with three different classifiers (NaiveBayes, KNN, Decision Tree). From the classification results, we try to learn how to perform feature selection and classifier selection based on the information hidden in the current training database. We use the same algorithms from Weka for KNN and decision tree classifiers as in the previous section. NaiveBayes classifier [11] is added to this experiment.

According to the results shown in Fig. 11, at the top level of Hornbostel/Sachs hierarchical tree (decision attribute is on class1 level) KNN classifier with feature A yields the best estimation confidence. At the beginning the system should use Flatness Coefficients and KNN to discover the family that the analyzed sound object belongs to. In order to perform the further estimation on the lower level of the instrument family, the system also needs to know how to select the feature and classifier at that particular level.

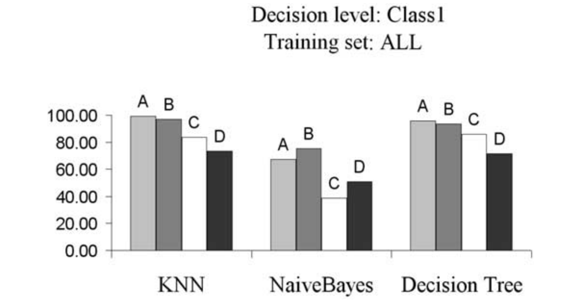

Fig. 11. Feature and classifier selection at top level.

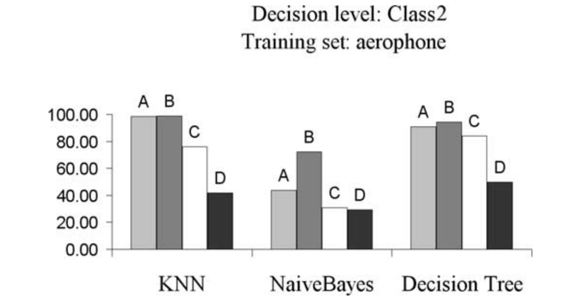

Fig. 12. Feature and classifier selection at second level for aerophones.

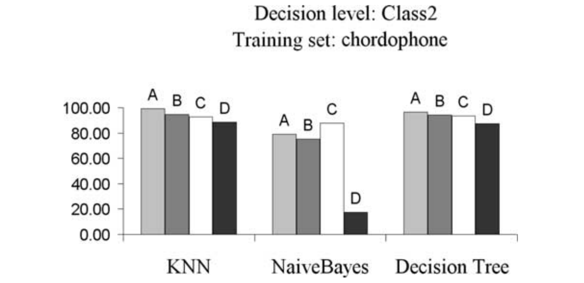

Fig. 13. Feature and classifier selection at second level for chordophones.

Figures 12, 13 and 14 tell us that KNN classifier and feature A (Flatness Coefficients) are still the best choice for the instruments falling in the families Chordophone or Idiophone. If the analyzed sound object is estimated as Aerophone, feature B (MFCC) is the better choice than others. Table 5 concludes the feature and classifier selection more clearly.

Table 5

Feature and classifier selection table for level 1

Node	Feature	Classifier
Chordophone	Flatness Coefficients	KNN
Aerophone	MFCC Coefficients	KNN
Idiophone	Flatness Coefficients	KNN

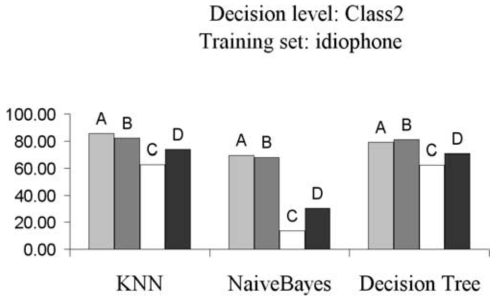

Fig. 14. Feature and classifier selection at second level for idiophones.

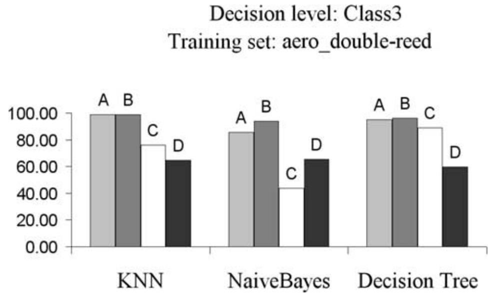

Fig. 15. Feature and classifier selection at third level for aero double reed.

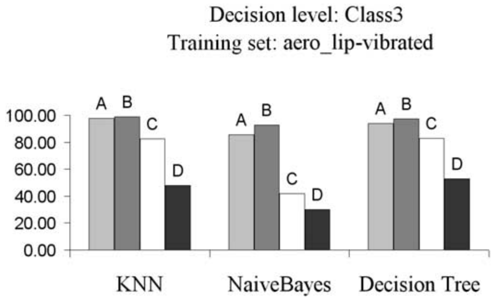

Fig. 16. Feature and classifier selection at third level for aero lip vibrated.

We continue to perform the classification on the different subsets at the third level of Hornbostel-Sachs hierarchical tree and get the classification results shown in Figs 15–20.

The instrument name is eventually estimated by the classifiers at the bottom level. Table 6 shows the classifier and feature selection results from the classification experiments at second level of the hierarchical tree. A notable result is that the subset of "Aero-single-reed" does not inherit the same character from the parent node of "Aerophone". Instead of selecting Feature B(MFCC) and KNN, the decision tree along with Feature A(Flatness Coefficients) yields the better classification result.

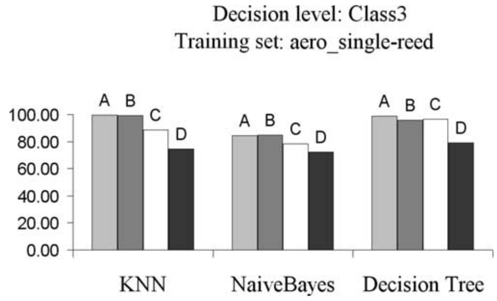

Fig. 17. Feature and classifier selection at third level for aero single reed.

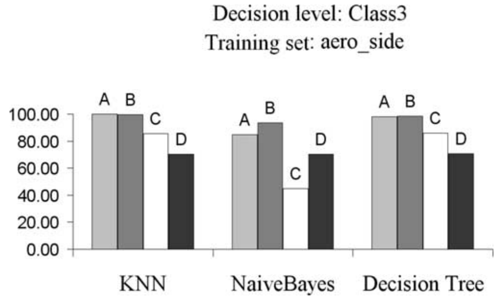

Fig. 18. Feature and classifier selection at third level for aero side.

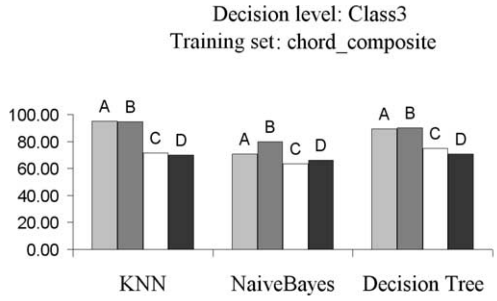

Fig. 19. Feature and classifier selection at third level for chord composite.

According to the above knowledge derived from the training database, we can optimize the feature selection and classifier selection at each level of hierarchical tree and further improve the overall estimation result for cascade classification system.

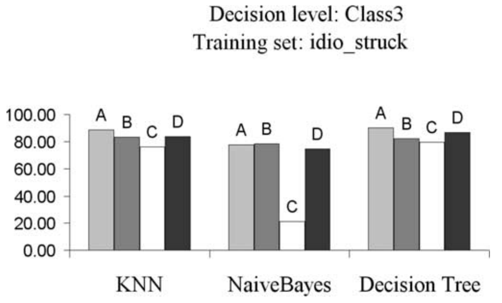

Fig. 20. Feature and classifier selection at third level for idio struck.

Table 6

Feature and classifier selection table for level 2

Node	Feature	Classifier
Chrd-composite	Flatness Coefficients	KNN
Aero-double-reed	MFCC Coefficients	KNN
Aero-lip-vibrated	MFCC Coefficients	KNN
Aero-side	MFCC Coefficients	KNN
Aero-single-reed	Flatness Coefficients	Decision Tree
Idio-struck	Flatness Coefficients	KNN

10. Hierarchical structure built by clustering analysis

Clustering is the method that divides data objects into similarity groups (clusters) according to a defined distance measure. Clustering is widely used as an important technique of machine learning and pattern recognition in the fields of biology, genomics and image analysis. However it has not been well investigated in the music domain since the category information of musical instruments has already been defined by musicians as the two hierarchical structures demonstrated in the last section. Those structures group the musical instruments according to their semantic similarity which is concluded from the human experience. However the instruments that are assigned to the same family or subfamily by those hierarchical structures often sound quite different from another. On the other hand, instruments that have very similar timbre qualities can be assigned to very different groups by those hierarchical structures. The inconsistency between the timbre quality and the family information causes the incorrect timbre estimation of cascade classification system. For instance, the trombone belongs to the aerophone family, but the system often classifies it as the chordophone instruments, such as violin. In order to take full advantage of the cascade classification strategy, we build the new hierarchical structure of musical instruments by the machine learning technique. Cluster analysis is commonly used to search for groups in data. This is most effective when the groups are not already known. We use the cluster analysis methods to reorganize the instrument group according to the similarity of timbre relevant features among the instruments.

10.1. Clustering analysis methods

There exist many clustering algorithms. Basically, all the clustering algorithms can be divided into two categories: partitional clustering and hierarchical clustering. Partitional clustering algorithms determine all clusters at once without hierarchical merging or dividing process. K-means clustering is most common method in this category [26]; K is the empirical parameter. Basically, it randomly assigns instances to K clusters. Next, new centroid for each of the K clusters and the distance of all items to these K centroids are calculated. Items are re-assigned to the closest centroid and the whole process is repeated until cluster assignments are stable. Hierarchical clustering generates a hierarchical structure of clusters which may be represented in a structure called a dendrogram. The root of the dendrogram consists of a single cluster containing all the instances, and the leaves correspond to individual instances. Hierarchical clustering can be further divided into two types according to whether the tree structure is constructed by following agglomerative or divisive approach. Agglomerative approach works in the bottom-up manner, it recursively merges smaller clusters into larger ones till some stoping condition is reached. Algorithms based on divisive (or top-down) approach begin with the whole set and then recursively split this set into smaller ones till some stoping condition is reached.

We have chosen the hierarchical clustering method to learn the new hierarchical schema for music instruments, since it fits our scenario well. There are many options to compute the distance between two clusters. The most common methods are the following [40]: Single linkage (nearest neighbor), Complete linkage (furthest neighbor), Unweighted pair-group method using arithmetic averages (UPGMA), Weighted pair-group method using arithmetic averages (WPGMA) [39], Unweighted pair-group method using the centroid average (UPGMC), Weighted pair-group method using the centroid average (WPGMC), Ward's method.

To complete the above definitions of a distance measure between two clusters, we also have to define the distance between their instances or centroids. Here are some most common distance measures between two objects:

1. Euclidean: Usual square distance between the two vectors. Disadvantages: not scale invariant, not for negative correlations

   $$d_{xy} = \sqrt{\sum (x_i - y_i)^2}.$$

2. Manhattan: Absolute distance between the two vectors.

   $$d_{xy} = \sum |x_i - y_i|.$$

3. Maximum: Maximum distance between any two components of x and y

   $$d_{xy} = \max |x_i - y_i|.$$

4. Canberra: Canberra distance examines the sum of series of a fraction differences between coordinates of a pair of objects. Each term of fraction difference has value between 0 and 1. If one of coordinates is zero, the term corresponding to this coordinate become unity regardless the other value, thus the distance will not be affected; if both coordinates are zero, then the term is defined as zero.

   $$d_{xy} = \sum \frac{|x_i - y_i|}{|x_i| + |y_i|}.$$

5. Pearson correlation coefficient (PCC) is a correlation-based distance. It measures the degree of association between two variables.

   $$\rho_{xy} = \frac{[cov(X,Y)]^2}{var(X)var(Y)}, \quad d_{xy} = 1 - \rho_{xy}$$

   where $cov(X,Y)$ is the covariance of the two variables, $var(X)$ and $var(Y)$ – the variance of each variable.
6. Spearman's rank correlation coefficient is another correlation based distance.

   $$\rho_{xy} = 1 - \frac{6\sum d_i^2}{n(n^2-1)}, \quad d_{xy} = 1 - \rho_{xy}$$

   where $d_i = x_i - y_i$ is the difference between the ranks of corresponding values x_i and y_i, and n is the number of values in each data set (same for both sets). Rank is calculated in the following way: 1 is assigned to the smallest element of each data, 2 to the second smallest element, and so on; the average ranking is calculated if there is a tie among different elements.

It is critical to choose an appropriate distance measure for objects in a musical domain because different measures may produce different shapes of clusters which represent different schema of instrument family. Different features also require the appropriate measures to be chosen in order to give better description of feature variation. The inappropriate measure could distort the characteristics of timbre which may cause the incorrect clustering.

10.2. Evaluation of different clustering algorithms for different features

As we can see, each clustering method has its own different advantage and disadvantage over others. It is a nontrivial task to decide which one is the most appropriate method for generating the hierarchical instrument classification structure. Not only the specific cluster linkage method needs to be decided in the hierarchical clustering algorithms, but also the good distance measurement has to be chosen in order to generate the good schema that represents the actual relationships among those instruments. We designed quite intensive experiments with the "cluster" package in R system [37]. The R package provides two hierarchical clustering algorithms: `hclust` (agglomerative hierarchical clustering), and `diana` (divisive hierarchical clustering). Table 7 shows all the clustering methods that we tested. We evaluated six different distance measurements (Euclidean, Manhattan, Maximum, Canberra, Pearson correlation coefficient, and Spearman's rank correlation coefficient) for each algorithm. For the agglomerative type of clustering (`hclust`), we also evaluated seven different cluster linkages that are available in this package: Ward, single (single linkage), complete (complete linkage), average (UPGMA), mcquitty (WPGMC), median(WPGMA), and centroid (UPGMC).

We have chosen the middle C pitch group which contains 46 different musical sound objects. We have extracted three different feature sets (MFCC [24], spectral flatness coefficients [23], and harmonic peaks [23]) from those sound objects. Each feature set produces one dataset for clustering. Some sound objects belong to the same instrument. For example, "ctrumpet" and "ctrumpet harmonStemOut" are objects pro-

Table 7

All distance measures and linkage methods tested for agglomerative and divisive clustering

Clustering algorithm	Cluster linkage	Distance measure
hclust	average	6 distance metrics
hclust	centroid	6 distance metrics
hclust	complete	6 distance metrics
hclust	mcquitty	6 distance metrics
hclust	median	6 distance metrics
hclust	single	6 distance metrics
hclust	ward	6 distance metrics
diana	N/A	6 distance metrics

duced by the same instrument: trumpet. We have preserved these particular object labels in our feature database without merging them as the same label because they could have very different timbre quality which the conventional hierarchical structure ignores. We have tried to discover the unknown musical instrument group information solely by the unsupervised machine learning algorithm, instead of applying any human guidance. Each sound object was segmented into multiple 0.12s frames and each frame was stored as an instance in the testing dataset. Since the segmentation is performed with overlap of 2/3 of the frame, there were totally 2884 frames from the 46 objects in each of the three feature datasets.

When our algorithm finishes the clustering job, a particular cluster ID is assigned to each frame. Theoretically, one may expect the same cluster ID to be assigned to all the frames of the same instrument sound object. However, the frames from the same sound object are not uniform and have variations in their feature patterns as the time evolves. Therefore, clustering algorithms do not perfectly identify them as the same cluster. Instead, some frames are assigned into other groups where majority of the frames come from other instrument sounds. As a result, multiple (different) cluster IDs are assigned to the frames of the same instrument object.

Our goal is to cluster the different instruments into the groups according to the similarity of timbre relevant features. Therefore, one important step of the evaluation is to check if a clustering algorithm is able to cluster most frames of an individual instrument sound into one group. In other words, a clustering algorithm should be able to differentiate most of the frames of one instrument sound from the others. It is evaluated by calculating the accuracy of a cluster ID assignment. We use the following example to illustrate this evaluation process. A hierarchical cluster tree T_m is produced by a clustering algorithm A_m. There are totally n instrument sound objects in the dataset ($n = 46$). The clustering package provides function $cutree$ to cut T_m into n clusters. One of these clusters is assigned to each frame. Table 8 shows a contingency table (x_{ij} represent numbers) derived from the clustering results after the $cutree$ is applied. It is a $n \times n$ matrix, where x_{ij} is the number of frames of $instrument_i$ that are labeled by $cluster_j$, and $x_{ij} \geq 0$.

Table 8

Format of the contingency table derived from clustering result

	$Clust_1$		$Clust_j$		$Clust_n$
$Instr_1$	x_{11}	$\cdots$	x_{1j}	$\cdots$	x_{1n}
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
$Instr_i$	x_{i1}	$\cdots$	x_{ij}	$\cdots$	x_{in}
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
$Instr_n$	x_{n1}	$\cdots$	x_{nj}	$\cdots$	x_{nn}

In order to calculate the accuracy of the cluster assignment, we need to decide which cluster ID corresponds to which instrument object. If cluster k is assigned to $instrument_i$, x_{ik} is the number of correct assignments for $instrument_i$, the accuracy of the clustering for $instrument_i$ is $\beta_i = x_{ik}/(\sum_{j=1}^{n} x_{ij})$.

During clustering process, each frame of the sound object is clustered into one particular group, and the group ID (i.e. instrument) is assigned to this frame. For each row, the maximum value is found among n columns, and next the column corresponding to the position of this maximum becomes the class label for frames represented in this row. However, it may happen that the maximum value is found in the same column also for other rows, and then the same group ID is linked to two different sound objects, which means these two different instrument sounds could not be distinguished by this particular clustering scheme. Clearly, we would like to avoid such an ambiguity. On the other hand, we have to cluster many frames of one sound object into a single group. Therefore, we would need permutations to calculate the theoretic best solution for the whole table, but such a large number of computations cannot be performed.

The overall accuracy for the clustering algorithm A_m is the average accuracy of all the instruments $\overline{\beta} = (\sum_{i=1}^{n} \beta_i)/n$. To find the maximum $\overline{\beta}$ among all possible cluster assignments to instruments, we should permute this matrix in order to find the maximum accuracy for the whole matrix (for each row of matrix, there are multiple values that could be selected among n columns), but it is not applicable to perform such a large number of calculations. This is why we have

Table 9

Evaluation result of `hclust` algorithm

Feature	Method	Metric	$\overline{\beta}$	w	Score
Flatness Coefficients	Ward	Pearson	87.3%	37	32.30
Flatness Coefficients	Ward	Euclidean	85.8%	37	31.74
Flatness Coefficients	Ward	Manhattan	85.6%	36	30.83
MFCC	Ward	Kendall	81.0%	36	29.18
MFCC	Ward	Pearson	83.0%	35	29.05
Flatness Coefficients	Ward	Kendall	82.9%	35	29.03
MFCC	Ward	Euclidean	80.5%	35	28.17
MFCC	Ward	Manhattan	80.1%	35	28.04
MFCC	Ward	Spearman	81.3%	34	27.63
Flatness Coefficients	Ward	Spearman	83.7%	33	27.62
Flatness Coefficients	Ward	Maximum	86.1%	32	27.56
MFCC	Ward	Maximum	79.8%	34	27.12
Flatness Coefficients	McQuitty	Euclidean	88.9%	33	26.67
MFCC	ward	Average	87.3%	30	26.20

chosen maximum x_{ij} in each row to approximate the optimal $\overline{\beta}$.

Since it is possible to assign the same cluster to multiple instruments, we have taken the number of clusters as well as accuracy into account. The final measurement to evaluate the performance of clustering is $score_m = \overline{\beta} \cdot w$, where w is the number of clusters, $w \leq n$. This measure reflects how well the algorithm clusters the frames from the same instrument object into the same cluster. It also reflects the ability of algorithm to separate instrument objects from each other.

In the experiments, we used two hierarchical clustering algorithms, `hclust` and `diana`. Table 9 presents 14 results which yielded the highest score among 126 experiments based on `hclust` algorithm.

From the results, the Ward linkage outperforms other methods and it yields the best performance when Pearson distance measure is used on the Flatness Coefficients feature dataset.

Table 10 shows the results from `diana` algorithm. In this algorithm, Euclidean yields the highest score on the mfcc feature dataset.

During the clustering process, we cut the hierarchical clustering result at a certain level, when obtaining groups which could represent instrument objects. If most of the frames from the same instrument object are clustered into one group, then this algorithm is selected to generate the hierarchical tree.

When we compare the two algorithms (`hclust` and `diana`), `hclust` yields better clustering results than `diana`. Therefore, we chose agglomerative clustering algorithm to generate the hierarchical schema for musical instruments, using Ward as the linkage method, Pearson distance measure as the distance metric, and Flatness Coefficients as the feature dataset to perform clustering analysis.

11. New hierarchical tree

Figure 21 shows the dendrogram result generated by the hierarchical clustering algorithm we chose (i.e. agglomerative clustering), as mentioned in Section 10.2. From this new hierarchical classification, we discover some instrument relationships which are not represented in the traditional schemas.

A musical instrument can produce sounds with quite different timbre qualities when different playing techniques are applied. One of the common techniques is muting. A mute is a device fitted to a musical instrument to alter the sound produced. It usually reduces the volume of the sound as well as affects the timbre. There are several different mute types for different instruments. The most common type used with the brass is the straight mute – a hollow, cone-shaped mute that fits into the bell of the instrument. This results in a more metallic, sometimes nasal sound, and when played at loud volumes can result in a very piercing note. The second common brass mute is the cup mute. Cup mutes are similar to straight mutes, but attached to the end of the mute's cone is a large lip that forms a cup over the bell. The result is removal of the upper and lower frequencies and a rounder, more muffled tone. In the case of string instruments of the violin family, the mute takes the form of a comb-shaped device attached to the bridge of the instru-

Table 10

Evaluation result of `Diana` algorithm

Feature	Metric	$\bar{\beta}$	w	Score
Flatness Coefficients	Euclidean	77.3%	24	18.55
Flatness Coefficients	Kendall	75.7%	23	17.40
Flatness Coefficients	Manhattan	76.8%	25	19.20
Flatness Coefficients	Maximum	80.3%	23	18.47
Flatness Coefficients	Pearson	79.9%	26	20.77
MFCC	Euclidean	78.5%	29	22.78
MFCC	Kendall	77.2%	27	20.84
MFCC	Manhattan	77.7%	26	20.21
MFCC	Pearson	83.4%	25	20.86
MFCC	Spearman	81.2%	24	19.48

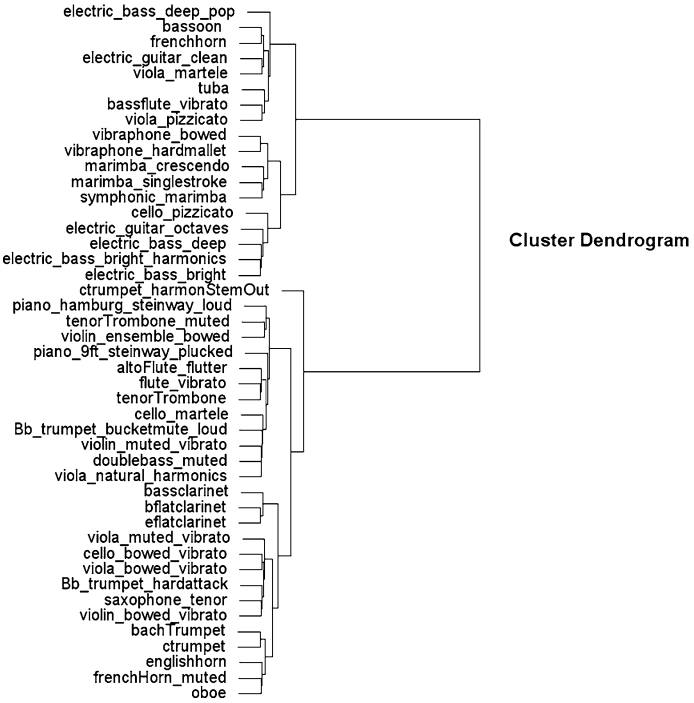

Fig. 21. Clustering result from `hclust` algorithm with Ward linkage method, Pearson distance measure, and Flatness Coefficients used as the feature set.

ment, dampening vibrations and resulting in a “softer” sound.

In the hierarchical structure shown in Fig. 21, “trumpet” and “ctrumpet harmonStemOut” represent two different sounds produced by the trumpet. “ctrumpet harmonStemOut” is produced when a particular mute is applied, called Harmon mute (different from the common straight or cup mutes). It is a hollow, bulbous metal device placed in the bell of the trumpet. All air is forced through the middle of the mute. This gives the mute a nasal quality. Protruding at the end of the device, there is a detachable stem extending through the centre of the mute. The stem can be removed completely or can be inserted to varying degrees. Name of this instrument sound object shows whether the stem is extended or completely removed, which darken the original piercing, strident timbre quality.

From the spectra of various sound objects (Fig. 22), we can clearly observe big differences between them. The spectra also show that “Bach trumpet” has more similar spectral pattern to “trumpet”. The relationships between C trumpet, C trumpet muted (Harmon, stem out) and Bach trumpet are accurately represented in the

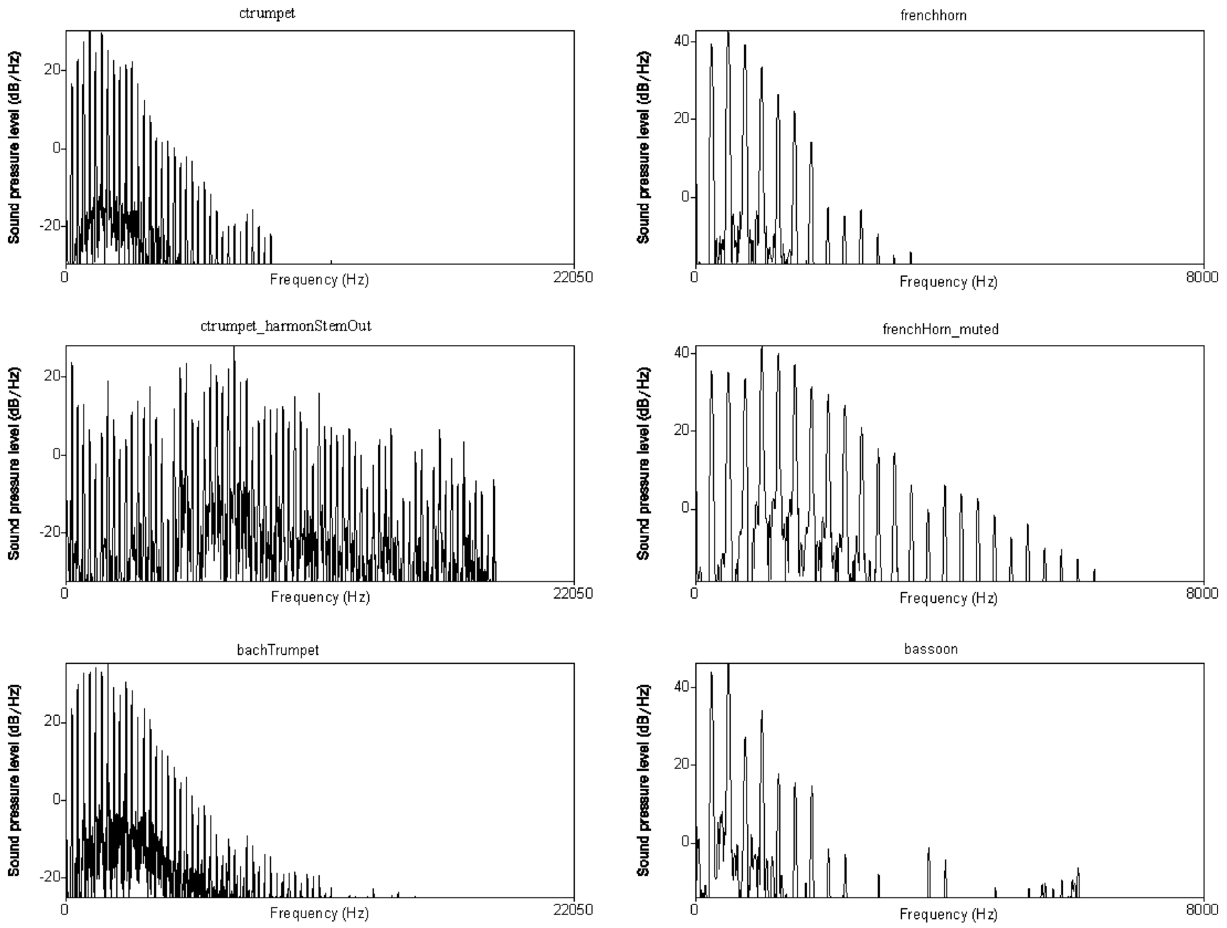

Fig. 22. Spectrum comparison of different instrument objects. On the left hand side: C Trumpet, C Trumpet muted (Harmon, Stem out) and Bach Trumpet; on the right hand side: French horn, French horn muted, bassoon.

new hierarchical schema. Figure 21 shows that "ctrumpet" and "bachtrumpet" are clustered into the same group. "ctrumpet harmonStemOut" is clustered in one single group instead of merging with "ctrumpet" since it has a very unique spectral pattern. The new schema also discovers the relationships among "French horn", "French horn muted" and "bassoon". Instead of clustering two "French horn" sounds in one group as the conventional schema does, bassoon is considered as the sibling of the regular French horn. "French horn muted" is clustered in another different group together with "English Horn" and "Oboe" (the extent of the difference between groups is measured by the distance between the nodes in the hierarchical tree).

According to this result, the new schema is more accurate than the traditional schema, because it represents the actual similarity of timbre qualities of musical instruments. Not only it better describes the differences between instruments, but it also distinguishes the sounds produced by the same instrument that have quite different timbre qualities due to different playing techniques.

12. Experiments and evaluation

In order to evaluate the new schema, we developed the cascade classification system based on the multi-label classification method and tested it with the new schema, as well as with the two previous conventional hierarchical schemas: Hornbostel-Sachs and Playing Method. The system used MS SQLSERVER2005 database system to store training dataset and MS SQLSERVER analysis server as the data mining server to build decision tree and process the classification request.

Training data The audio files used in this research consist of stereo musical pieces from the McGill University Master Samples (MUMS, [30]). Each file has two channels: left channel and right channel, in .au (or .snd) format. These audio data files are treated as mono-channel, where only left channel is taken into consideration, since successful methods for the left channel can also be applied to the right channel, or any channel if more channels are available. 2917 single instrument sound files were used, representing 45 different instruments.

Each sound stands for one note played by a specific instrument. Many instruments can produce different timbres when they are played using different techniques. Therefore, sounds of various pitch and articulation were investigated for each of these 45 instruments.

Power spectrum and 33 spectral flatness coefficients were extracted from each frame of these single instrument sounds, according to the equations described by the MPEG-7 standard [23]. The frame size was 120 ms and the overlap between two adjacent frames was 80ms, to reduce the information loss caused by windowing function (therefore, the hop size was 40ms). The total number of frames for the entire feature database reaches to about one million, since each sound is analyzed in many frames. For instance, the instrument sound which only lasts three seconds is segmented into 75 overlapped frames. The classifier is trained by the obtained feature database.

Testing data 308 mixed sounds were synthesized by randomly choosing two single instrument sounds from 2917 training data files. Spectral flatness coefficients were extracted from the frames of mixes, in order to perform instrument family estimation on the higher level of the hierarchical tree. After reaching the bottom level of the hierarchical tree, we used the power spectrum from the frames representing mixes, in order to match against the reference spectral database. Since the spectrum matching is performed in a small subgroup, the computation complexity is reduced. The same analyzing frame size and hop size were used for the mixes as in the case of training data. We use precision, recall, and F-score to evaluate the performance of classifiers. The definitions of recall and precision are shown in Fig. 23. I_1 is the number of actual instruments playing in the analyzed sound. I_2 is the number of instruments estimated by the system. I_3 is the number of correct estimations.

Recall is the measurement to evaluate the recognition rate and precision is to evaluate the recognition accuracy. The F-score is often used in the field of information retrieval for measuring search, document classification, and query classification performance. It is the harmonic mean of precision and recall. F-score is calculated as

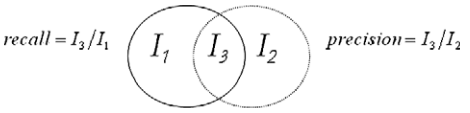

Fig. 23. Precision and recall.

$$\texttt{F-score} = \frac{2 \times precision \times recall}{precision + recall}.$$

The average recall, precision and F-score of all the 308 sounds estimations were calculated to evaluate each method.

Since timbre estimation was performed for indexing segments (smoothing window), containing multiple frames, as described in Section 6, the measures mentioned above were calculated for indexing segments of size 1 second.

K-Nearest Neighbor (KNN) [18] was used as the classifier, with $k = 3$. As shown in Table 11, in Experiment 1 we applied the multiple label classification [16] based on features representing spectral flatness coefficients only. In Experiment 2 we used the power spectrum matching method, instead of features [15]. In Experiment 3 and Experiment 4 we used two traditional hierarchical structures (Hornbostel-Sachs and play method) in order to perform cascade classification based on both the power spectrum and spectral flatness coefficients. In Experiment 5 we applied the new hierarchical structure as the basis of the cascade system. The indexed window size for all the experiments is one second, and the output of total number of estimations for each indexed window is controlled by confidence threshold $\lambda = 0.4$, which is the minimum average confidence of instrument candidates.

Figure 24 shows that generally the cascade classification improves the recall compared to the non-cascade methods. The non-cascade classification based on spectrum-match (Experiment 2) shows higher recall than the cascade classification approaches based on the traditional hierarchical schema (Experiment 3 and Experiment 4). However, the cascade classification based on the new schema learned by the clustering analysis (Experiment 5) outperforms the non-cascade classification. It increases the recall by 8 percent points, precision by 12 percent points and general F-score by 9 percent points. This shows that the new schema yields sig-

Table 11
Comparison between non-cascade classification and cascade classification with different hierarchical schemas

Experiment	Classification method	Description	Recall	Precision	F-Score
1	Non-Cascade	Feature-based	64.3%	44.8%	51.4%
2	Non-Cascade	Spectrum-Match	79.4%	50.8%	60.7%
3	Cascade	Hornbostel-Sachs	75.0%	43.5%	53.4%
4	Cascade	play method	77.8%	53.6%	62.4%
5	Cascade	machine Learned	87.5%	62.3%	69.5%

Table 12
Classification results of 3-instrument mixtures with different algorithms

Experiment	Classifier	Method	Recall	Precision	F-Score
1	Non-Cascade	single-label based on sound separation	31.48%	43.06%	36.37%
2	Non-Cascade	feature-based multi-label classification	69.44%	58.64%	63.59%
3	Non-Cascade	spectrum-match multi-label classification	85.51%	55.04%	66.97%
4	Cascade (Hornbostel-Sachs)	multi-label classification	64.49%	63.10%	63.79%
5	Cascade (playmethod)	multi-label classification	66.67%	55.25%	60.43%
6	Cascade (machine learned)	multi-label classification	63.77%	69.67%	66.59%

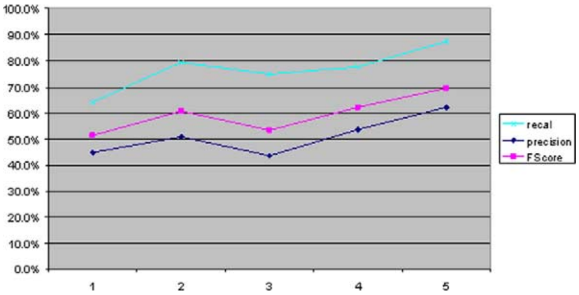

Fig. 24. Comparison between non-cascade classification and cascade classification with different hierarchical schemas. X-axis is the experiment number described in Table 12.

nificant improvement in comparison to the other two traditional schemas. Also, since the hierarchical tree has more levels, the size of the subset on the bottom level is reduced to a very small size, which significantly reduces the cost of spectrum matching.

We evaluated the classification system by the mixed sounds which contain two single instrument sounds. In the real world recordings, there could be more than two instruments playing simultaneously, especially in the orchestra music. Therefore, we also created 49 polyphonic sounds by randomly selecting three different single instrument sounds and mixing them together. Next, we tested those three-instrument mixes, using various classification methods (Table 12).

As we can see from Table 12, the lowest precision and recall is obtained for the algorithm based on sound separation, i.e. separating sounds of mixes and then performing instrument estimation on separated sounds. This is because there is no much information left in the sound mix for the further classification of the third instrument after two signal subtractions corresponding to the first two instrument estimations are made. The cascade method based on multi-label classification again yields high recall and precision of results.

This experiment shows the robustness and effectiveness of the algorithm for the polyphonic sounds which contain more than two timbres. As the dendrogram in Fig. 21 shows, the new schema has more hierarchical levels and looks more complex and obscure to users. However, we only use it as the internal structure for the cascade classification process, and we do not use it in the query interface. Therefore, when the user submits a query to QAS defined in the user's semantic structure (e.g. searching instrument sounds which are close in Hornbostel-Sachs classification, or with respect to the play method), system translates it to the internal schema (based on clustering). After the estimation is done, the answer is converted back to the user semantics. The user does not need to know the difference between French horn and French horn muted since only French horn is returned by the system as the final estimation result. The internal hierarchical representation of musical instrument sound classification is used as an auxiliary tool, assisting answering user's queries.

Acknowledgements

This material is based in part upon work supported by the National Science Foundation under Grant Number IIS-0968647. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] L. Akinobu et al., *Julius software toolkit*, http://julius.sourceforge.jp/en/.

[2] D. Aha and D. Kibler, Instance-based learning algorithms, *Machine Learning* **6** (1991), 37–66.

[3] V. Athitsos, J. Alon, and S. Sclaroff, Efficient nearest neighbor classification using a cascade of approximate similarity measures, in: *Proc. of the 2005. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.

[4] J.C. Brown, Computer identification of musical instruments using pattern recognition with cepstral coefficients as features, *J. Acoust. Soc. Am.* **105** (1999), 1933–1941.

[5] P. Cosi, Auditory modeling and neural networks, in: *A Course on Speech Processing, Recognition, and Artificial Neural Networks*, LNCS, Springer, 1998.

[6] D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun, A practical part-of-speech tagger, in: *Third Conference on Applied Natural Language Processing*, 1992, pp. 133–140.

[7] A. Eronen and A. Klapuri, Musical instrument recognition using Cepstral coefficients and temporal features, in: *Proc. of ICASSP*, 2000.

[8] S. Essid, G. Richard and B. David, Instrument recognition in polyphonic music based on automatic taxonomies, *IEEE Transactions on Audio, Speech and Language Processing* **14**(1) (2006), 68–80.

[9] Y. Freund, Boosting a weak learning algorithm by majority, in: *Proc. of the Third Annual Workshop on Computational Learning Theory*, 1990.

[10] Y. Freund and R.E. Schapire, Decision-theoretic generalization of on-line learning and application to boosting, *Journal of Computer and System Sciences* **55**(1) (1997), 119–139.

[11] F. Fuhrmann, M. Haro and P. Herrera, Scalability, generality and temporal aspects in automatic recognition of predominant musical instruments in polyphonic music, in: *Proc. of ISMIR*, 2009, pp. 321–326.

[12] R.-E. Haskell, Design of hierarchical classifiers, in: *Proc. of The First Great Lakes Computer Science Conference on Computing*, LNCS, Vol. 507, 1989, pp. 118–124.

[13] T. Heittola, A. Klapuri and T. Virtanen, Musical instrument recognition in polyphonic audio using source-filter model for sound separation, in: *Proc. of ISMIR*, 2009, pp. 327–332.

[14] K. Jensen and J. Arnspang, Binary decision tree classification of musical sounds, in: *International Computer Music Conference*, Beijing, China, Oct., 1999.

[15] W. Jiang, A. Wieczorkowska, and A.Z. Ras, Music instrument estimation in polyphonic sound based on short-term spectrum match, in: *Foundations of Computational Intelligence Volume 2*, A.-E. Hassanien et al., eds, Studies in Computational Intelligence, Vol. 202, Springer, 2009, pp. 259–273.

[16] W. Jiang, A. Cohen, and Z. Ras, Polyphonic music information retrieval based on multi-label cascade classification system, in: *Advances in Information & Intelligent Systems*, Z.W. Ras and W. Ribarsky, eds, Studies in Computational Intelligence, Springer, 2009.

[17] G.H. John and P. Langley, Estimating continuous distributions in bayesian classifiers, in: *Eleventh Conference on Uncertainty in Artificial Intelligence*, San Mateo, 1995, pp. 338–345.

[18] I. Kaminskyj, Multi-feature musical instrument sound classifier, *Mikropolyphonie WWW Journal* (2001), http://farben.latrobe.edu.au/mikropol/articles.html.

[19] T. Kitahara, M. Goto, K. Komatani, T. Ogata, and H.G. Uno, Instrument identification in polyphonic music: Feature weighting to minimize influence of sound overlaps, *Journal on Advances in Signal Processing*, Hindawi Publishing Corporation (2007).

[20] B. Kostek and A. Czyzewski, Representing musical instrument soundsfor their automatic classification, *Journal of Audio Eng. Society* **49**(9) (2001), 768–785.

[21] J. Kupiec, Robust part-of-speech tagging using a hidden markov model, *Computer Speech and Language* **6** (1992), 225–242.

[22] R. Lienhart, A. Kuranov, and V. Pisarevsky, Empirical analysis of detection cascades of boosted classifiers for rapid object detection, *Pattern Recognition Journal* (2003), 297–304.

[23] A.T. Lindsay and J. Herre, MPEG-7 and MPEG-7 audio-an overview, *J. Audio Eng. Soc.* **49** (2001), 589–594.

[24] B. Logan, Mel frequency cepstral coefficients for music modeling, in: *1st Ann. Int. Symposium On Music Information Retrieval*, Vol. 28, 2000.

[25] C. Lu and M.S. Drew, Construction of a hierarchical classifier schema using a combination of text-based and image-based approaches, in: *SIGIR Proceedings*, ACM Publications, 2001, pp. 331–336.

[26] J.B. MacQueen, Some methods for classification and analysis of multivariate observations, in: *Proc. of 5th Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, 1967, pp. 281–297.

[27] K.D. Martin and Y.E. Kim, Musical instrument identification, a pattern-recognition approach, in: *Proc. of 136th Meeting of the Acoustical Soc. of America*, Norfolk, VA, 2pMU9, 1998.

[28] Music Information Retrieval System – MIRAI: http://www.mir.uncc.edu.

[29] *MPEG-7 Overview*, http://mpeg.chiariglione.org/standards/mpeg-7/mpeg-7.

[30] F. Opolko and J. Wapnick, *MUMS – McGill University Master Samples CD's*, 1987.

[31] J. Paulus and T. Virtanen, Drum transcription with non-negative spectrogram factorization, in: *Proc. of 13th European Signal Processing Conference*, EUSIPCO, Antalya, Turkey, 4–8 September, 2005.

[32] Z. Pawlak, Information systems – theoretical foundations, *Information Systems Journal* **6** (1981), 205–218.

[33] H.F. Pollard and E.V. Jansson, A tristimulus method for the specification of musical timbre, *Acustica* **51**(5) (1982).

[34] W.H. Press and S.A. Teukolsky, *Numerical Recipes in C*, 2nd edn, Cambridge, 1992.

[35] R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[36] Z. Ras, X. Zhang, and R. Lewis, MIRAI: Multi-hierarchical, FS-tree based music information retrieval system (invited paper), in: *Proc. of the International Conference on Rough Sets and Intelligent System Paradigms (RSEISP 2007)*, LNAI, Vol. 4585, Springer, 2007, pp. 80–89.

[37] R Development Core Team, R: Language and environment for statistical computing, in: *R Foundation for Statistical Computing*, Vienna, Austria, ISBN 3-900051-07-0, http://www.R-project.org, 2005.

[38] E.Scheirer and M. Slaney, Construction and evaluation of a robust multi-feature speech/music discriminator, in: *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997.

[39] P.H.A. Sneath and R.R. Sokal, *Numerical Taxonomy*, Freeman, San Francisco, 1973.

[40] *The Statistics Homepage: Cluster Analysis*, http://statsoft.com/textbook/stathome.html.

[41] I.H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edn, Morgan Kaufmann, San Francisco, 2005.

[42] T. Zhang, Instrument classification in polyphonic music based on timbre analysis, in: *Proc. of SPIEs Conference on Internet Multimedia Management Systems II*, IT-Com, Denver, 2001, pp. 136–147.

[43] X. Zhang, Z.W. Ras, and A. Dardzinska, Discriminant feature analysis for music timbre recognition and automatic indexing, in: *Mining Complex Data, Post-Proceedings of 2007. ECML/PKDD Third International Workshop (MCD 2007)*, LNAI, Vol. 4944, Springer, 2008, pp. 104–115.