Application of temporal feature extraction to musical instrument sound recognition

Alicja A. Wieczorkowska (alicja@pjwstk.edu.pl), Jakub Wróblewski (jakubw@pjwstk.edu.pl), Piotr Synak (synak@pjwstk.edu.pl) and Dominik Ślęzak (slezak@pjwstk.edu.pl)
Polish-Japanese Institute of Information Technology, ul. Koszykowa 2, 02-008 Warsaw, Poland

Abstract. Automatic content extraction from multimedia files is a hot topic nowadays. Moving Picture Experts Group develops MPEG-7 standard, which aims to define a unified interface for multimedia content description, including audio data. Audio description in MPEG-7 comprises features that can be useful for any content-based search of sound files. In this paper, we investigate how to optimize sound representation in terms of musical instrument recognition purposes. We propose to trace trends in evolution of values of MPEG-7 descriptors in time, as well as their combinations. Described process is a typical example of KDD application, consisting of data preparation, feature extraction and decision model construction. Discussion of efficiency of applied classifiers illustrates capabilities of further progress in optimization of sound representation. We believe that further research in this area would provide background for automatic multimedia content description.

Keywords: knowledge discovery in databases, music content processing, multimedia content description, MPEG-7

1. Introduction

Automatic extraction of multimedia information from files is recently of great interest. Usually multimedia data available for end users are labeled with some information (title, time, author, etc.), but in most cases it is insufficient for content-based searching. For instance, the user cannot find automatically all segments with his favorite tune played by the flute in the audio CD. To address the task of automatic content-based searching, descriptors need to be assigned at various levels to segments of multimedia files. Moving Picture Experts Group is finishing works on MPEG-7 standard, named "Multimedia Content Description Interface" (ISO/IEC, 2002), that defines a universal mechanism for exchanging the descriptors. However, neither feature (descriptor) extraction nor searching algorithms are encompassed in MPEG-7. Therefore, automatic extraction of multimedia content, including musical information, should be a subject of study.

All descriptors used so far reflect specific features of sound, describing spectrum, time envelope, etc. In our paper, we propose a different approach: we suggest observation of feature changes in time and taking as new descriptors patterns in trends observed for particular features. We discuss how to achieve it by applying data preprocessing and mining tools developed within the theory of rough sets introduced in (Pawlak, 1991).

The analyzed database origins from audio CD's MUMS (Opolko and Wapnick, 1987). These CD's contain sounds of musical instruments, played with various articulation techniques. We processed samples representing woodwind instruments, brass, and strings of contemporary orchestra. The obtained database was divided into 18 classes, where each class represents single instrument and selected articulation technique.

2. Sound descriptors

2.1. MPEG-7 descriptors

Descriptors of musical instruments should allow to recognize instruments independently on pitch and articulation. Sound features included in MPEG-7 Audio are based on research performed so far in this area and they comprise technologies for musical instrument timbre description, sound recognition, and melody description. Audio description framework in MPEG-7 includes 17 temporal and spectral descriptors divided into the following groups (cf. (ISO/IEC, 2002)):

- basic: instantaneous waveform, power values
- basic spectral: log-frequency power spectrum, spectral centroid, spectral spread, spectral flatness
- signal parameters: fundamental frequency, harmonicity of signals
- timbral temporal: log attack time and temporal centroid
- timbral spectral: spectral centroid, harmonic spectral centroid, spectral deviation, spectral spread, spectral variation
- spectral basis representations: spectrum basis, spectrum projection

Each of these features can be used to describe a segment with a summary value that applies to the entire segment or with a series of sampled values. An exception is the timbral temporal group, as its values apply only to segments as a whole.

2.2. Other applied descriptors

Although descriptors included in MPEG-7 are based on published research, number of the descriptors included in the standard has been limited to 17, in order to obtain compact representation of audio content for search purposes and other applications. Apart from the features included in MPEG-7 (Peeters et al., 2000), the following descriptors have been used in the research:

- duration of the attack, quasi-steady state and ending transient of the sound in proportion to the total time (Kostek and Wieczorkowska, 1997)
- moments of time wave (Brown et al., 2001)
- pitch variance - vibrato (Martin and Kim, 1998), (Wieczorkowska, 1999b)
- contents of the selected groups of harmonics in spectrum (Kostek and Wieczorkowska, 1997), like even/odd harmonics Ev/Od

  $$Ev = \frac{\sqrt{\sum_{k=1}^{M} A_{2k}^2}}{\sqrt{\sum_{n=1}^{N} A_n^2}} \qquad Od = \frac{\sqrt{\sum_{k=2}^{L} A_{2k-1}^2}}{\sqrt{\sum_{n=1}^{N} A_n^2}} \tag{1}$$

  and lower/middle/higher harmonics $Tr_1/Tr_2/Tr_3$ (Tristimulus parameters (Pollard and Jansson, 1982), used in various versions)

  $$Tr_1 = \frac{A_1^2}{\sum_{n=1}^{N} A_n^2} \quad Tr_2 = \frac{\sum_{n=2,3,4} A_n^2}{\sum_{n=1}^{N} A_n^2} \quad Tr_3 = \frac{\sum_{n=5}^{N} A_n^2}{\sum_{n=1}^{N} A_n^2} \tag{2}$$

  where A_n denotes the amplitude of the n^{th} harmonic, N is the number of harmonics available in spectrum, $M = \lfloor N/2 \rfloor$ and $L = \lfloor N/2 + 1 \rfloor$
- statistical properties of sound spectrum, including average amplitude and frequency deviations, average spectrum, standard deviations, autocorrelation and cross-correlation functions (Ando and Yamaguchi, 1993), (Brown et al., 2001)
- various properties of the spectrum, including higher order moments, such as skewness and kurtosis, spectral irregularity (Fujinaga and McMillan, 2000)
- constant-Q coefficients (Brown, 1999), (Kaminskyj, 2000)

- cepstral and mel-cepstrum coefficients and derivatives (Batlle and Cano, 2000), (Brown, 1999), (Eronen and Klapuri, 2000)
- multidimensional scaling analysis trajectories (Kaminskyj, 2000)
- descriptors based on wavelet analysis (Wieczorkowska, 1999a), (Kostek and Czyzewski, 2001), Bark bands (Eronen and Klapuri, 2000) and other (Herrera et al., 2000), (Wieczorkowska and Raś, 2001)

2.3. Descriptors used in our research

The main goal of this paper is to verify, how much one can gain by analyzing widely used descriptors by means of the dynamics of their behavior in time. We restrict ourselves to a small part of the known descriptors, to be able to compare the results obtained with and without analysis of temporal behavior more clearly. We begin the analysis process with the following descriptors:

Temporal descriptors:

- Signal length, denoted as *Length*
- Relative length of the attack (till reaching 75% of maximal amplitude), quasi-steady (after the end of attack, till the final fall under 75% of maximal amplitude) and decay time (the rest of the signal), denoted, respectively, by *Attack*, *Steady* and *Decay*
- Moment of reaching maximal amplitude, denoted by $Maximum$

Spectral descriptors:

- Harmonics defined by (1), denoted by $EvenHarm$ and $OddHarm$
- Brightness and Irregularity (see e.g. (Wieczorkowska, 1999a))
- Tristimulus parameters given by (2), denoted by $Tristimulus1, 2, 3$
- Fundamental frequency, denoted by $Frequency$

It's worth mentioning that the above spectral descriptors were used so far in literature only in purpose of reflecting specific static features. In the foregoing sections, we propose to consider the same features but calculated over the chains of reasonably small time intervals. It allows to observe the sound's behavior in time, what is especially interesting for the attack time.

3. Musical instrument sound recognition

3.1. Classification models

One of the main goals of data analysis is to construct models, which properly classify objects (described by some attributes) to some predefined classes. Reasoning with data can be stated as a classification problem, concerning prediction of decision class basing on information provided by attributes. For this purpose, one stores data in so called decision tables, where each training case drops into one of predefined decision classes.

A decision table takes the form of $\mathbf{A} = (U, A \cup \{d\})$, where each attribute $a \in A$ is identified with a function $a : U \to V_a$ from the universe of objects U into the set V_a of all possible values on a. Values $v_d \in V_d$ correspond to mutually disjoint decision classes of objects. In case of the analysis of the musical instrument sound data (Opolko and Wapnick, 1987), one deals with a decision table consisting of 667 records corresponding to samples of musical recordings. We have 18 decision classes corresponding to various kinds of musical instruments – flute, oboe, clarinet, violin, viola, cello, double bass, trumpet, trombone, French horn, tuba – and their articulation – vibrato, pizzicato, muted (Wieczorkowska, 1999b). These classes define decision attribute d.

Methods for construction of classifiers can be regarded as tools for data generalization. These methods include rule-based classifiers, decision trees, k-nearest neighbor classifiers, neural nets, etc. Problem of musical instrument sound recognition has been approached in several research studies, applying various methods. The most common one is k-nearest neighbor algorithm, applied in (Martin and Kim, 1998), (Fujinaga and McMillan, 2000), (Eronen and Klapuri, 2000), (Kaminskyj, 2000). To obtain better results, Fujinaga and MacMillan (2000) applied k-nearest neighbor classifier to weighted feature vectors and a genetic algorithm to set the optimal set of weights. Brown in her research (Brown, 1999) applied clustering and Bayes decision rules, using k-means algorithm to calculate clusters, and forming Gaussian probability density functions from the mean and variance of each of the clusters. Martin and Kim (1998) used maximum a posteriori classifiers, constructed based on Gaussian models obtained through Fisher multiple-discriminant analysis. Gaussian classifier was also used by Eronen and Klapuri (2000). Apart from statistical methods, machine learning tools have also been applied to musical instrument sound classification. For instance, classification based on binary trees was used in (Wieczorkowska, 1999a). Another popular approach to musical instrument sound classification is based on various neural network techniques, see (Cosi et al., 1994), (Toiviainen, 1996), (Wieczorkowska, 1999a). Research based on hidden Markov models was reported in (Batlle and Cano, 2000), whereas Wieczorkowska (1999b) applied rough set approach to musical sound classification. Extensive review of classification methods applied to this research, including the above mentioned and other (for instance, support vector machines) is given in (Herrera et al., 2000).

3.2. KDD framework

All the above approaches are based on adapting the well known classifier construction methods to the specific domain of musical instrument sounds. Actually, the process of analyzing data cannot be restricted just to the classifier construction. In the particular case of the musical instrument analysis, one has to extract a decision table itself – to choose the most appropriate set of attributes-descriptors A, as well as to calculate values $a(u) \in V_a$, $a \in A$, for particular objects-samples $u \in U$. Thus, it is better to write about this task in terms of a broader methodology.

Knowledge Discovery in Databases (KDD) is a process, which, according to widely accepted scheme, consists of several steps (see e.g. (Düntsch et al., 2000)), such as

- understanding application domain
- determining a goal
- creating/selecting a target data set
- preprocessing
- data reduction and transformation
- selection of data mining method, algorithms and parameters
- model construction (data mining)
- interpretation of results

In case of classification of musical instruments, the first two steps comprise of the musical domain analysis. Next, proper selection (Liu and Motoda, 1998) and reduction (Pawlak, 1991) of the set of features is crucial for efficiency of classification algorithm. In some cases, a set of attributes is worth transforming into more suitable form before it is used to model the data. For instance, when the data set is described by decision rules, one may transform attribute values to gain higher support of rules, keeping their accuracy, and increasing generality of a model. The need of such a transformation is shown for various kinds of feature domains: numeric, symbolic, as well as, e.g., for time series (see e.g. (Nguyen, 2000), (Ślęzak and Wróblewski, 1999), (Synak, 2000), (Wróblewski, 2000)).

3.3. Extraction of temporal features

One can realize that the above mentioned methods for feature extraction can be of crucial importance while considering musical instruments. Because of the nature of the musical sounds, the methods concerned with time series analysis seem to be of a special interest. Generally, it is difficult to find numerical description of musical instrument sounds that allows correct classification of instrument for sound of various pitch and/or articulation. Listener needs transients (especially the beginning of sound) to correctly classify musical instrument sounds, but during transients sound features change dramatically and they usually differ from sound features for the steady state. It is illustrated by Figure 1, where fragments of time domain of oboe sound a^1 of frequency 440Hz are presented.

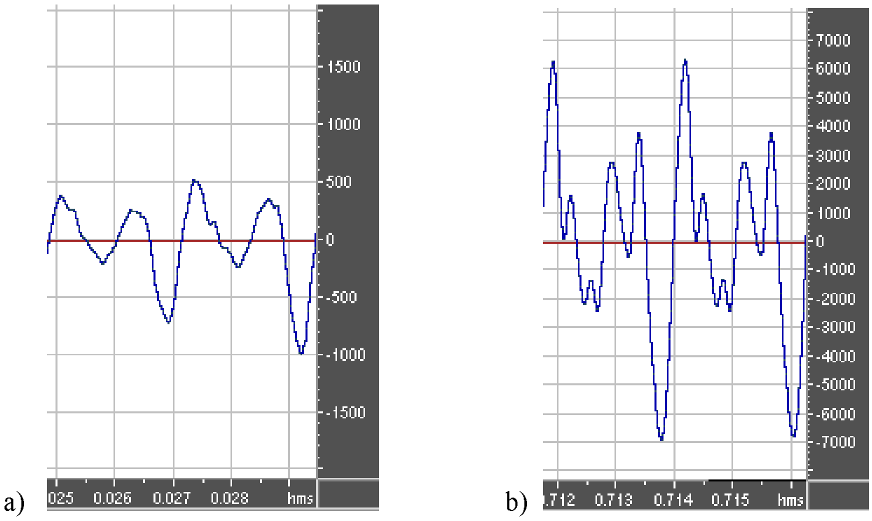


Figure 1. Time domain for 2 periods of the oboe sound a^1 = 440Hz during the attack of the sound (a) and the quasi-steady state (b).

As we can observe, the beginning (attack) of the sound significantly differs from the quasi-steady state. During the attack, changes are very rapid, but in the quasi-steady some change may happen as well, especially when the sound is vibrated. Feature vectors used so far in the research reflect mainly quasi-steady state and the attack of sounds. The features used are based on time domain analysis, spectral analysis, and some other approaches (for example, wavelet analysis). Time domain analysis can describe basic features applicable to any sounds, like basic descriptors from MPEG-7, or features specific for the whole sound, like timbral temporal features from MPEG-7.

4. Preprocessing of musical sound data

4.1. Data description

In purpose of learning classifiers for the musical instrument sound recognition, we need to prepare the training data in the form of decision table $\mathbf{A} = (U, A \cup \{d\})$, where each element $u \in U$ corresponds to a sound sample, each element $a \in A$ is a numeric feature corresponding to one of sound descriptors and decision attribute $d \notin A$ labels particular object-sound with integer codes adequate to instrument. For such a preparation, we need a framework for preprocessing original data, in particular, for extracting features most relevant to the task of the sound recognition.

The sound data are taken from MUMS audio CD's that contain samples of broad range of musical instruments, including orchestral ones, piano, jazz instruments, organ etc. (Opolko and Wapnick, 1987). These CD's are widely used in musical instrument sound research, see (Cosi et al., 1994), (Martin and Kim, 1998), (Wieczorkowska, 1999b), (Fujinaga and McMillan, 2000), (Kaminskyj, 2000), (Eronen and Klapuri, 2000), so we consider they became a standard. Our database consists of 667 samples of recordings, divided into the following 18 classes: violin vibrato, violin pizzicato, viola vibrato, viola pizzicato, cello vibrato, cello pizzicato, double bass vibrato, double bass vibrato, double bass pizzicato; flute, oboe and b-flat clarinet; trumpet, trumpet muted, trombone, trombone muted, French horn, French horn muted, and tuba.

4.2. Envelope descriptors

Attributes $a \in A$ can be put into $\mathbf{A} = (U, A \cup \{d\})$ in various ways. They can be based on analysis of various descriptors, their changes in time, their mutual dependencies, etc. Let us begin with the following example of a new, temporal feature. Consider a given sound sample, which is referred as object $u \in U$. We can split it onto, say, 7 intervals

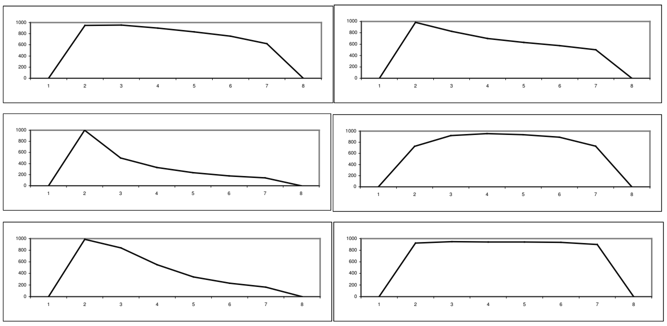


Figure 2. Centroids (the most typical shapes) of sound envelopes, used in clustering.

of equal length. Average values of amplitudes within these intervals are referred, respectively, as $Val_{Amp}1, \ldots, 7$. Sequence

$$\overrightarrow{Val}_{Amp}(u)\, \langle Val_{Amp}1(u), \ldots, Val_{Amp}7(u) \rangle \tag{3}$$

corresponds to a kind of envelope, approximating the behavior of amplitude of each particular u in time. We can consider, e.g., Euclidean distance over the space of such approximations. Then we can apply one of basic clustering or grouping methods to find the most representative envelopes. At Fig. 2 we show representative envelopes as centroids obtained from algorithm dividing data onto 6 clusters. We obtain a new group of attributes, labeling each sample-object $u \in U$ with amplitude envelope parameters:

Envelope descriptors:

1. Each sample was split onto 7 intervals of equal length. Average values of amplitudes within these intervals are referred, respectively, as $Val_{Amp}1, \ldots, 7$.

2. Area under the curve of envelope (approximated by means of values $Val_{Amp}1, \ldots, 7$), denoted by $EnvFill$

3. Number of envelope based cluster ($Cluster$) is the number of the closest of 6 representative envelope curves, shown at Fig. 2.

Another sound descriptors' evolution were analyzed using envelopes in our experiments, but experiments show, that the envelope of amplitude is the most useful of them.

4.3. Fundamental frequency and time frames

The main difficulty of sound analysis is that many useful attributes of sound are not concerned with the whole sample. E.g. spectrum-based attributes (tristimulus parameters, pitch, etc.) describe rather a selected time frame on which the spectrum was calculated than the whole sound (moreover, these attributes may change from one segment of time to another). One can take a frame from quasi-steady part of a sample and treat it as a representative of the whole sound but in this case we may loose too much information about the sample.

Our approach is to take into account both the sample based attributes and the window based ones. We propose to consider small time windows, of the length equal to the fundamental sound period times four. Within each such window, we can calculate local values of spectral (and also other) descriptors. For each particular attribute, its local window based values create the time series, which can be further analyzed in various ways. For instance, we can consider envelope based attributes similar to those introduced in the previous subsection. Such envelopes, however, would be referring not to the amplitudes, but to the dynamics of changes observed for spectral descriptors in time.

Usually, the analyzing window length is constant for the whole analysis, with the most common length is about 20-30 ms. For example, Eronen and Klapuri (2000) used 20 ms window, Brown (1999) reported 23 ms analyzing window, Batlle and Cano (2000) applied 25 ms window, and Brown et al. (2001) used 32 ms window. Such a window is sufficient for most sounds, since it contains at least a few periods of the recorded sounds. However, such a window is too short for analysis of the lowest sounds we used, and long for analysis of short pizzicato sounds, where changes are very fast, especially in case of higher sound. This is why we decided to set up the length of those intervals as 4 times the fundamental period of the sound. We decompose each musical sound sample onto such intervals and calculate value sequences and final features for each of them.

Spectral time series should be stored within an additional table, where each record corresponds to a small window taken from a sound sample. Hence, we first need to extract the lengths of windows for particular sounds. It corresponds to the well known problem of extracting fundamental frequency from data. Given frequency, we could calculate, for each particular sound sample, fundamental periods and derive necessary window based attributes.

There are numerous mathematical approaches for approximation of fundamental signal frequency by means of the frequency domain or estimation of the length of period (and fundamental as an inverse) by means of the time domain. The methods used in musical frequency tracking include autocorrelation, maximum likelihood, cepstral analysis, Average Magnitude Difference Function (AMDF), methods based on zero-crossing of the sound wave etc., see (Brown and Zhang, 1991), (Doval and Rodet, 1991), (Cook et al., 1992), (Beauchamp et al., 1993), (Cooper and Ng, 1994), (de la Cuadra et al., 2001); most of these methods originate from speech processing. Frequency tracking methods applied to musical instrument sounds are usually tuned to the characteristics of spectrum (sometimes assumption about the frequency are required), and octave errors are common problem here. Therefore frequency estimation instrument-independent is quite difficult. However, with the development of MPEG-7 standard, we can expect that audio databases with labeled with the frequency can be available in close in the foreseeable. For our research purposes, we have used AMDF in the following form (see (Cook et al., 1992)):

$$AMDF(i) = \frac{1}{N} \sum_{k=0}^{N} |A_k - A_{i+k}| \tag{4}$$

where N is the length of interval taken for estimation and A_k is the amplitude of the signal. One can calculate the values of $AMDF(i)$ within the interval of a few admissible period lengths and approximate the period for a given sound by taking the minimal value of $AMDF(i)$. The problem is that during the attack time the values of $AMDF$ are less reasonable than in case of the rest of the signal, after stabilization. On the other hand, the most interesting behavior of local values of many descriptors can be observed during the attack time. We cope with this problem by evaluating the approximate period length within the stable part of the sound and then – tuning it with respect to the part corresponding to the attack phase (Wieczorkowska, 1999a). In experiments we use a mixed approach to approximate periods – based both on searching for stable minima of $AMDF$ and maxima of spectrum obtained using DFT (Discrete Fourier Transform).

4.4. The final structure of database

Given a method for extracting frequencies, we can accomplish the preprocessing stage and concentrate on data reduction and transformation. The obtained database framework for these operations is illustrated at Fig. 3.

- Table SAMPLE (667 records, 18 columns) gathers temporal and spectral descriptors. It has additional column *Instrument* which

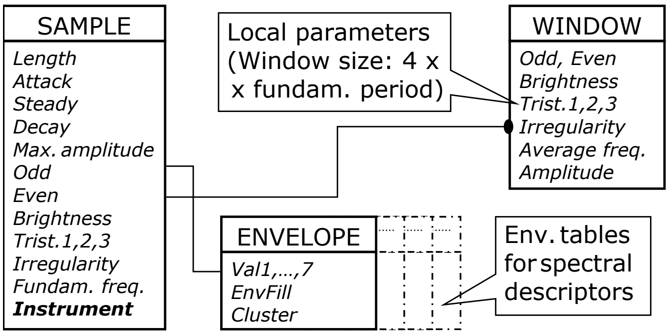


Figure 3. Relational musical sound database after the preprocessing stage.

states the code of musical instrument, together with its articulation (18 values).

- Table ENVELOPE (667 records, 7 columns) is linked in 1:1 manner with Table SAMPLE. By default, its columns correspond to attributes derived by considering amplitudes. However, we define analogous ENVELOPE tables also for other descriptors.

- Table WINDOW (190800 records, 10 columns) gathers records corresponding to local windows. According to the previous subsection, for each sample we obtain (Length∗Frequency/4) records. Each record is labeled with spectral descriptors defined in the same way as for Table SAMPLE but calculated locally.

As a result, we obtain the relational database, where tables SAMPLE and WINDOW are linked in 1:n manner, by the code of the instrument sample (primary key for SAMPLE and foreign key for WINDOW). All additional data tables used in our experiments were derived from this main relational database. E.g. we have created (and collected in additional data table) envelopes of some spectral features of sound (see section 7) by calculating average values of subsequent records from WINDOW table in 6 intervals of equal width. Temporal templates and episodes (see section 6) are also created using this database.

5. Automatic Extraction of New Attributes

5.1. Relational Approach

Given the database structure as illustrated by Fig. 3, one can move to the next stage of the knowledge discovery process – the feature extraction. In the particular case of this application, we need features – attributes – describing the sound samples. Hence, we need to use information stored within the database to create new, reasonable columns within the Table SAMPLE, where records correspond to the objects we are interested in.

One can see that the ENVELOPE tables considered in the previous subsection consist of some new attributes describing musical samples. The values of these attributes were extracted from the raw data at the preprocessing level. Still, some new attributes can be added in a more automatic way. Namely, one can use relations between already existing tables. One can construct a parameterized space of possible features based on available relations. Then one can search for optimal features in an adaptive way, verifying which of them seem to be the best for constructing decision models.

Such a process has been already implemented for SQL-like aggregations in (Wróblewski, 2000). Exemplary features, found automatically as SQL-like aggregations from table WINDOW, can be of the following nature:

- *sum of LocFreq from WINDOW*
- *average of LocIrr from WINDOW*
- *sum of LocIrr from WINDOW*
- *sum of LocOdd from WINDOW*
- *sum of LocTri1 from WINDOW where LocTri1* > 25
- *sum of LocTri3 from WINDOW where LocTri3* $<$ *LocTri2*

E.g., *sum of LocIrr* is the new attribute which value is equal to the sum of *irregularity* parameters (section 2.3) calculated over the whole set of frames corresponding to one sample. *LocFreq*, *LocOdd* and *LocTri* correspond to (calculated locally on frames) other parameters: sound *frequency*, *OddHarm* and *Tristimulus*.

The goal of the searching algorithm is here to extract aggregations of potential importance while distinguishing instrument decision classes. The example presented above collects the best new attributes (according to a quality measure presented in (Wróblewski, 2000)). Such attributes are then added as new columns to table SAMPLE. In some situations adding such new features improves and simplifies the laws of reasoning about new cases. Still, in this particular case of database with only one 1:n relation, the usage of this approach does not provide much valuable knowledge. Moreover, the temporal nature of Table WINDOW requires a slightly different approach than that directly based on SQL-like aggregations. We go back to this issue in Section 6.

The advantage of the presented approach is especially visible when using rule based data mining methods (rough set based data mining algorithm, see (Ślęzak, 2001), (Wróblewski, 2001b)). Adding 9 new attributes to the original data table increased the recognition rate to 49.7%, comparing with 48.5% obtained with the same algorithm without any new features. On the other hand, plain k-NN algorithm (with no further feature selection step) cannot utilize the new information: results of classification were not better than the original ones.

5.2. Single Table Based Extraction

Automatic extraction of significantly new features is possible also for single data tables, not embedded into any relational structure. In case of numerical features, such techniques as discretization, hyperplanes, clustering, and principle component analysis (see e.g. (Nguyen, 2000)), are used to transform the original domains into more general or more descriptive ones. One can treat the analysis process over transformed data either as a modeling of a new data table (extended by new attributes given as a function of original ones) or, equivalently, as an extension of model language. The latter means, e.g., change of metric definition in k-NN algorithm or extension of language of rules or templates.

In our approach the original data set is extended by a number of new attributes defined as a linear combination of existing ones. Let $B = b_1, \ldots, b_m \subseteq A$ be a subset of attributes, $|B| = m$, and let $\alpha = (\alpha_1, \ldots, \alpha_m) \in \mathbf{R}^m$ be a vector of coefficients. Let $h : U \to \mathbf{R}$ be a function defined as:

$$h(u) = \alpha_1 b_1(u) + \ldots + \alpha_m b_m(u) \tag{5}$$

Usefulness of new attribute defined as $\overline{a}(u) = h(u)$ depends on proper selection of parameters B and α. The new attribute $\overline{a}$ is useful, when the model of data (e.g. decision rules) based on discretized values of $\overline{a}$ becomes more general (without loss of accuracy).

Evolution strategy algorithm optimizes the coefficients of $\overline{a}$ using various quality functions. Three of them are implemented in the current version of the Rough Set Expert System RSES (Bazan et al., 2002). Theoretical foundations of their usage are described in (Ślęzak and Wróblewski, 1999), as well as (Ślęzak, 2001; Wróblewski, 2001b).

Let L be a straight line in $\mathbf{R}^m$ defined by given linear combination h. The general idea of the mentioned evaluation measures is given below.

- The *distance measure* is average (normalized) distance of objects from different decision classes in terms of $\overline{a}$ (i.e. projected onto L). The value of the distance measure can be expressed as follows:

  $$Dist(\overline{a}) = \sum_{i=1,\ldots,N} \sum_{j=i+1,\ldots,N: d(u_i) \neq d(u_j)} \frac{|\overline{a}\,(u_i) - \overline{a}\,(u_j)|}{max(\overline{a}) - min(\overline{a})} \tag{6}$$

  where $max(\overline{a})$ and $min(\overline{a})$ are maximal and minimal values of $\overline{a}$ over objects $u \in U$. In (Ślęzak, 2001) it is shown that this measure is equivalent to the average measure of rough set based quality of cuts defined over the domain of $\overline{a}$.

- The *discernibility measure* takes into account two components: distance (as above) and average discernibility, defined as a sum of squares of cardinalities of decision-uniform intervals defined on L. This measure turned out to be effective for classification of the benchmark data sets in (Ślęzak and Wróblewski, 1999). In its simplified form, without considering the distance coefficient, it can be defined as follows (cf. (Ślęzak, 2001)):

  $$Disc(\overline{a}) = \sum_{i=1}^{M} r_i^2 \tag{7}$$

  where $r_i = |\{u \in U : c_i < \overline{a}(u) \leq c_{i+1}\}|$ is a number of objects included in the i-th interval, $c_{M+1} = +\infty$, and $M = |C_a|$ for the minimal possible set of cuts $C_{\overline{a}} = \{c_1, \ldots, c_M\}$, which enables to construct consistent decision rules based on $\overline{a}$, i.e. such that

  $$\forall_{u,u'} \left[\overline{a}(u) < \overline{a}(u') \wedge d(u) \neq d(u') \Rightarrow \exists_{i=1,\ldots,M} (\overline{a}(u) < c_i \leq \overline{a}(u'))\right] \tag{8}$$

  This quality function also refers to intuition that a model with lower number of (consistent) decision rules is better than the others (cf. (Bazan et al., 2000), (Pawlak, 1991)).

- The *predictive measure*. This measure is an estimate of expected classifier's prediction quality when using only $\overline{a}$. It is constructed with use of the probabilistic methods for approximating the expected values of coverage and sensibility (ability to assign the objects to proper classes; cf. (Wróblewski, 2001a; Wróblewski, 2001b)). Using the same notation:

$$Pred(\overline{a}) = 1 - \prod_{i=1}^{M} \left(1 - \frac{r_i - 1}{|U| - 1}\right) \tag{9}$$

This measure is more suitable for rule based data mining methods rather than for distance based ones (e.g. k-NN).

6. Time domain features

6.1. Temporal vs. relational features

The basis of musical sound recognition process is a properly chosen set of descriptors that potentially contains relevant features distinguishing one instrument from another. It seems to be very important to choose not only descriptors characterizing the whole sample at once, but also those describing how parameters change in time (see Figure 4). Features described in Section 2 can be used to describe a segment with a summary value or with a series of sampled values. Descriptors can be stored as a sequence corresponding to the dynamic behavior of a given feature over the sound sample. Analysis of regularities and trends occurring within such a temporary sequence can provide the values of conditional features labeling objects-sounds in the final decision table. Especially interesting trends are supposed to be observed during the attack part of signal.

Extraction of temporal templates or temporal clusters can be regarded as a special case of using 1:n connection between data tables. Here, new aggregated columns are understood in terms of deriving descriptors corresponding to trends in behavior of values of some locally defined columns (in our case: spectral columns belonging to table WINDOW), ordered by the time column. The difference with respect to the aggregations described in the previous section is that temporal aggregations cannot be expressed in SQL-like language. One of the main goals of the future research is to automatize the process of defining temporal attributes, to get ability of massive search through the space of all possibilities of temporal descriptors. Then, one would obtain an extended model of relational feature extraction developed in (Wróblewski, 2000) and (Wróblewski, 2001b), meeting the needs of the modern database analysis.

We propose to search for temporal patterns that can potentially be specific for one instrument or a group of instruments. Such patterns can be further used as new descriptors like *Cluster* in table ENVELOPE.

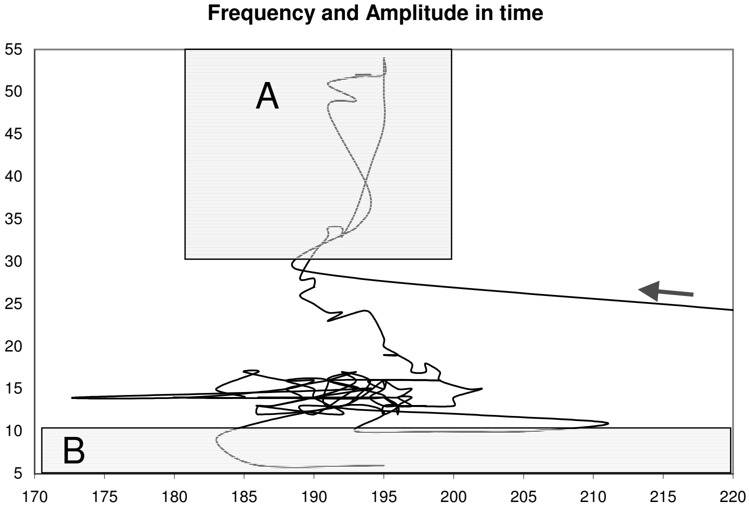


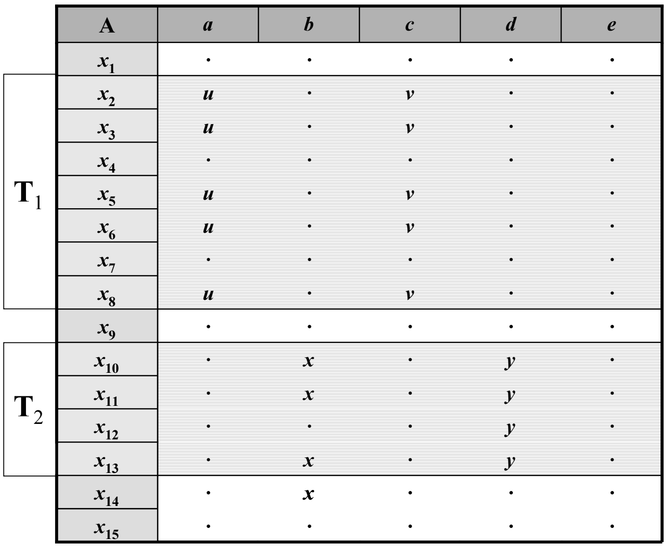
Figure 4. Evolution of two sound parameters in time. Examples of templates: $A = (180 < Freq < 200,\ 30 < Ampl < 60)$, $B = (Ampl < 10)$. The episode ABB occurs in this sound.

These attributes describe general trends of the amplitude values in time. Results presented in Section 7 show potential importance of such features. Similar analysis can be performed over spectral features stored in table WINDOW, by searching for, e.g., *temporal patterns* (cf. (Synak, 2000)).

6.2. Temporal patterns

Generation of temporal patterns requires the choice of descriptors that would be used to characterize sound samples and a method to measure values of those descriptors in time. For the latter we propose to use time window based technique. We browse a sample with time windows of certain size. For a given time window we compute values of all descriptors within it, and this way generate one object of a *temporal information system* $\mathbf{A} = (\{x_1, x_2, \ldots, x_n\}, A)$, where x_i is a measurement from the i-th window using descriptors from A (actually, we constructed table WINDOWS by repeating this procedure for all samples of sounds). Next, we use it to determine optimal *temporal templates* that respond to temporal patterns. Thus for one sample we compute a sequence of temporal templates.

Temporal templates are built from expressions $(a \in V)$, called *descriptors*, where $a \in A$, $V \subseteq V_a$ and $V \neq \emptyset$. Formally, *template* is a set



	A	*a*	*b*	*c*	*d*	*e*
	x_1	·	·	·	·	·
$\mathbf{T}_1$	x_2	*u*	·	*v*	·	·
	x_3	*u*	·	*v*	·	·
	x_4	·	·	·	·	·
	x_5	*u*	·	*v*	·	·
	x_6	*u*	·	*v*	·	·
	x_7	·	·	·	·	·
	x_8	*u*	·	*v*	·	·
	x_9	·	·	·	·	·
$\mathbf{T}_2$	x_{10}	·	*x*	·	*y*	·
	x_{11}	·	*x*	·	*y*	·
	x_{12}	·	·	·	*y*	·
	x_{13}	·	*x*	·	*y*	·
	x_{14}	·	*x*	·	·	·
	x_{15}	·	·	·	·	·

Figure 5. Temporal templates for temporal information system $\mathbf{A} = (\{x_1, \ldots, x_{15}\}, \{a, b, c, d, e\})$: $\mathbf{T_1} = (\{(a \in \{u\}), (c \in \{v\})\}, 2, 8)$, $\mathbf{T_2} = (\{(b \in \{x\}), (d \in \{y\})\}, 10, 13)$.

of descriptors involving any subset $B \subseteq A$:

$$T = \{(a \in V) : \; a \in B, \, V \subseteq V_a\}. \tag{10}$$

By *temporal template* we understand

$$\mathbf{T} = (T, t_s, t_e), \;\; 1 \le t_s \le t_e \le n, \tag{11}$$

that is template placed in time – with corresponding period $[t_s, t_e]$ of occurrence (see Figure 5). Let us define some basic notions related to temporal templates. First of all, by *width* we understand the length of period of occurrence, i.e. $width(\mathbf{T}) = t_e - t_s + 1$. *Support* is the number of objects from period $[t_s, t_e]$ matching all descriptors from T. Finally, *precision* of temporal template is defined as sum of precisions of all descriptors from T, where precision of descriptor $(a \in V)$ is given by:

$$Precision\,((a \in V)) \begin{cases} \frac{card(V_a) - card(V)}{card(V_a) - 1} & card(V_a) \ge 1 \\ 1 & otherwise \end{cases} \tag{12}$$

We consider quality of temporal template as a function of *width*, *support* and *precision.*

Templates and temporal templates are intensively studied in literature, see (Agrawal et al., 1996), (Nguyen, 2000), (Synak, 2000). To

]

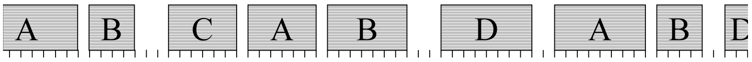


Figure 6. A series of temporal templates.

outline the intuition, which is behind these notions, let us understand template as a strong regularity in data, whereas temporal template as strong regularity occurring in time.

In one musical sound sample we can find several temporal templates. They can be time dependent, i.e. one can occur before or after another. Though, we can treat them as a sequence of events (see Figure 6 and 4). From such a sequence we can discover frequent *episodes* – collections of templates occurring together, see e.g. (Mannila et al., 1998), (Synak, 2000). We expect some of such episodes to be specific only for particular instrument or group of instruments.

6.3. Extraction of temporal features

We propose the following scheme for new attribute generation from sound samples:

1. For each training sample we generate a sequence of temporal templates. As the input we take objects from table WINDOW (see Section 4), i.e. spectral features of samples measured in windows of size equal to four times the fundamental period of sound. Because these attributes are real, we discretize them first using uniform scale quantization – number of intervals is a parameter.

2. Number of different templates (in terms of descriptors only) found for all samples is relatively large. Therefore, to search for any template dependencies, characteristic for samples of one kind, we have to assure that their total number is rather small. For this purpose we generate a number of representative templates, i.e. templates of highest quality with respect to some measure. Number of such representatives is another parameter of described method. We propose to use the following measures of quality. One can see that these are just examples of measures based on theories of machine and statistical learning, as well as rough sets, which can be applied at this stage of the process.

   Measure $BayesDist(T)$, defined by

$$BayesDist(T) = \sum_k |P(T/k) - P(T)| \qquad (13)$$

describes the impact of decision classes onto probability of T. By $P(T)$ we mean simply the prior probability that elements of the universe satisfy T. By $P(T/k)$ we mean the same probability, but derived from the k-th decision class. If we regard T as the left side of an inexact decision rule $T \Rightarrow d = k$, then $P(T/k)$ describes its sensitivity (cf. (Mitchell, 1998)). The quantities of the form $|P(T/k) - P(T)|$ expresses a kind of degree of information we gain about T given knowledge about membership of the analyzed objects to particular decision classes. According to the Bayesian principles (cf. (Box and Tiao, 1992)), $BayesDist(T)$ provides the degree of information we gain about decision probabilistic distribution, given additional knowledge about satisfaction of T.

Measure $RoughDisc(T)$, defined by

$$RoughDisc(T) = P(T) \cdot (1 - P(T)) - \sum_{k} P(T,k) \cdot (P(k) - P(T,k)) \tag{14}$$

is an adaptation of one of the rough set measures used e.g. in (Nguyen, 1997) and (Ślęzak, 2001) to express the number of pairs of objects belonging to different decision classes, being discerned by a specified condition. Normalization of that number, understood as dividing by the number of all possible pairs, provides quantity (14).

3. Using template representatives we replace each found template with the closest representative. The measure of closeness is the following one:

$$DIST(T_1, T_2) = \sum_{a \in A} \left(1 - \frac{|V_a^{T_1} \cap V_a^{T_2}|}{|V_a^{T_1} \cup V_a^{T_2}|} \right). \tag{15}$$

4. Each sequence of temporal templates, found for each sample, is now expressed in terms of a number of representative templates. We can expect some regularities in those sequences, possibly specific for one or more classes of instruments. To find those regularities we propose an algorithm, based on A-priori (see e.g. (Agrawal and Srikant, 1994), (Mannila et al., 1998)), which discovers "frequent" episodes with respect to some frequency measure. The difference, comparing e.g. to Winepi algorithm (Mannila et al., 1998), is that here we are looking for episodes across many series of events.

   By an episode we understand a sequence of templates. An episode occurs in a sequence of templates if each element (template) of episode exists in a sequence and order of occurrence is preserved. For example, episode AAC occurs in sequence $B\underline{A}C\underline{A}BB\underline{C}$. On the input of the algorithm we have a set of template sequences and frequency threshold τ.

```
Frequent episode detection
1. $\mathcal{F}_1 = \{$frequent 1-sequences$\}$
2. for $(l = 2;\ \mathcal{F}_{l-1} \neq \emptyset;\ l{+}{+})$ {
3.     $\mathcal{C}_l = GenCandidates(\mathcal{F}_1, \mathcal{F}_{l-1}, l)$
4.     $\mathcal{F}_l = \{c \in \mathcal{C}_l : Frequency(c) \geq \tau\}$
5. }
6. return $\bigcup_l \mathcal{F}_l$
```

At first, we check which templates occur in all sequences with frequency at least τ. That forms set $\mathcal{F}_1$ of frequent episodes of length one. We can consider several measures of frequency. The fundamental one is just the number of occurrences, however, being "frequent" we can also understand as frequent occurrence in one class of instruments and rare occurrence in another classes. Therefore, we can adapt measures (13), (14) to definition of episode's frequency (function $Frequency()$).

Next, we recursively create a set of candidates $\mathcal{C}_l$ by combining frequent templates ($\mathcal{F}_l$) with frequent episodes of size $l-1$ ($\mathcal{F}_{l-1}$).

The last step is to verify the set of candidates $\mathcal{C}_l$ and eliminate infrequent episodes.

5. We generate two attributes related to occurrence of frequent episodes in a series of templates found in a sound sample. The first one is an episode of highest frequency (with respect to chosen frequency measure) out of all episodes that occur in a given sequence. The second one is the longest episode – if there is more than one such episode we choose that with highest frequency.

Presented method requires evaluation of many parameters. The most important ones are: window size (when generating temporal templates), scaling factor, number of representative templates, quality and frequency measure, frequency threshold.

7. Results of experiments

7.1. Results known from literature

The research on musical instrument sound classification is performed all over the world and there are some results. However, the data differ from one experiment to another and it is almost impossible to compare the results of experiments. The most common data come from McGill University Master Samples (MUMS) CD collection (Opolko and Wapnick, 1987).

Experiment carried out so far operate on various number of instruments and classes. Some experiments are based on a few instruments, and sometimes only singular sounds of the selected instruments are used. It is also quite common to classify not only instrument, or instrument and a specific articulation, but also instrument classes. final results vary depending on the size of the data, feature vector, and classification method applied.

Brown et al. (2001) reported correct identifications of 79%-84% for 4 classes (oboe, sax, clarinet, flute), with cepstral coefficients, constant-Q coefficients, and autocorrelation coefficients applied to short segments of solo passages from real records. Each instrument was represented by least 25 sounds. The results depended on the training sounds chosen and the number of clusters used in the calculation. Bayes decision rules were applied to the data clustered using k-means algorithm; this method was earlier applied by Brown (1999) to oboe and sax data only.

Kostek and Czyzewski (2001) applied 2-layer feedforward neural networks with momentum method to classify various groups of 4 orchestral instruments, recorded on DAT. Feature vectors consisted of 14 FFT-based or 23 wavelet-based parameters. The results reached 99% for FFT vectors and 91% for wavelet vectors; various testing procedures were applied.

Martin and Kim (1998) identified instrument families (string, woodwind, and brass) with approximately 90% performance, and individual instruments with an overall success rate of approximately 70% for 1023 isolated tones over the full pitch ranges of 14 orchestral instruments. The classifiers were constructed based on Gaussian models, arrived at through Fisher multiple-discriminant analysis, and cross-validated with multiple 70%/30% splits. 31 perceptually salient acoustic features related to the physical properties of source excitation and resonance structure were calculated for MUMS sounds.

Wieczorkowska (1999a), (1999b) applied rough set based algorithms, decision trees, and some other algorithms to the data representing 18 classes (11 orchestral instruments, full pitch range) taken from MUMS

CDs. The results approached 90% for instrument families (string, woodwind, and brass) and 80% for singular instruments with specified articulation. Various testing procedures were used, including 70%/30% and 90%/10% splits. Feature vectors were based on FFT and wavelet analysis, including time-domain features as well.

Fujinaga and MacMillan (2000) reported the recognition rate 50% for the 39-timbre group (23 orchestral instruments) with over 1300 notes from McGill CD library, and 81% for a 3-instrument group (clarinet, trumpet, and bowed violin). They applied k-nearest neighbor classifier and genetic algorithm to seek the optimal set of weights for the features. Standard leave-one-out cross-validation procedure was used to calculate the recognition rate.

Kaminskyj (2000) obtained overall accuracy of 82% using combined nearest neighbor classifiers with different k values and using the leave-one-out classification scheme. The data describe 19 musical instruments of definite pitch, taken from MUMS CDs. Kaminskyj used the following features: RMS amplitude envelope, constant Q transform frequency spectrum and multidimensional scaling analysis trajectories.

Eronen and Klapuri (2000) recognized instrument families (string, brass, and woodwind) with 94% accuracy and individual instruments with 80% rate using 1498 samples covering the full ranges of 30 orchestral instruments, played with various articulation techniques. 44 spectral and temporal features were calculated for sounds mostly taken MUMS collection, and guitar and piano by amateur players recorded on DAT. The classification was cross-validated with 70%/30% splits of train and test data.

Extensive comparison of results of experiments on musical instrument sound classification worldwide is presented in (Herrera et al., 2000).

7.2. Experiments based on the proposed approaches

Fig. 7 presents the results of classification of sounds with respect to the kinds of instruments and their usage. We consider 18 decision classes and 667 records. We use standard CV-5 method for evaluation of resulting decision models. Presented results correspond to two approaches to constructing classifiers:

- Best k-NN: Standard implementation with tuning parameter k. The best results among different values of k as well as different metrics (Euclidean, Manhattan) is presented.

Attributes	**Best k-NN**	**RS-decision rules**
Envelope	36.3%	17.6%
Envelope with linear combinations	42.1%	11.1%
Temporal	54.3%	39.4%
Spectral	34.2%	14.6%
Temporal + Spectral	68.4%	48.5%
Temporal + Spectral + Relational	64.8%	49.7%
Spectral envelopes	32.1%	–
– linear combinations	32.1%	–
Spectral env. (clustered)	31.3%	–
– linear combinations	31.3%	–

Figure 7. Experimental results.

- RS-decision rules: Algorithm presented in (Bazan et al., 2000) for finding optimal ensembles of decision rules, based on the theory of rough sets (Pawlak, 1991)

Particular rows of the table in Fig. 7 correspond to performance of the above algorithms over decision tables consisting of various sets of conditional attributes. Groups of features correspond to notation introduced in Section 4:

- Envelope: 36% of correct classification of new cases into 18 possible decision classes – a good result in case of k-NN over 7 quite naive conditional features.

- Envelope with linear combinations: Improvement of correct classification in case of k-NN after adding linear combinations over original Envelope of dimensions, found by the approach discussed in Section 5. This confirms the thesis about importance of searching for optimal linear combinations over semantically consistent original features, stated in (Ślęzak and Wróblewski, 1999). On the other hand, one can see that extension of the set of envelope based attributes is not good in combination with RS-decision rules – 11.1% is not much better than random choice.

- Temporal: Incredible result for just a few, very simple descriptors, ignoring almost the whole knowledge concerning the analysis of music instrument sounds. Still k-NN (54.3%) better than RS-decision rules (39.4%). In general, one can see that k-NN is a better approach for this specific data (although it's not always the case – see e.g. (Polkowski and Skowron, 1998)). Obviously, it would be still better to base, at least partially, on decision rules while searching for intuitive explanation of the reasoning process.

– Spectral: Classical descriptors related to spectrum analysis seem to be not sufficient to this type of task. From this perspective, the results obtained for Temporal features are even more surprising.

– Temporal + Spectral: Our best result, 68.4% for k-NN, still needing further improvement. Again, performance of RS-decision rules is worse (48.5%), although other rough set based methods provide better results – e.g., application of the algorithm for the RSES library (see (Bazan and Szczuka, 2000)) gives 50.3%.

– Temporal + Spectral + Relational: Another rough set based classification algorithms, described in (Ślęzak, 2001) and (Wróblewski, 2001b), provide – if taken together with new (automatically created) features listed in Section 5 – up to 49.7%.

– Spectral envelopes: A general shape (calculated over 6 intervals and normalized) of change of spectral parameters in time. There are 5 spectral features (*Brightness*, *Irregularity*, *Tristimulus1,2,3*) which evolution is described by 30 numerical values. Relatively low recognition rate (32.1%), especially compared with the result for amplitude envelope (only 6 numerical values), shows that changes of spectral features are not specific enough. Optimized linear combinations of these 30 numerical values give the same recognition quality (which may be regarded as a success, since a number of attributes has been limited to 15). Initial results of experiments using rule based system were not promising (below 15%, probably because all of these features are numerical) and this method was not used in the further experiments.

– Spectral envelopes (clustered): The 5 envelopes used in the previous experiment was clustered (into 5 groups each), then a distance to a centroid of each cluster was calculated for an object. These distances (5 numerical values for each spectral attribute) was collected in a decision table described by 25 conditional attributes. Results (and discussion) are similar to the previous experiment.

8. Conclusions

We focus on methodology of musical instrument sound recognition, related to KDD process of the training data analysis. We propose a novel approach, being a step towards automatic extraction of musical information within multimedia contents. We suggest to build classifiers by basing on appropriately extracted features calculated for particular sound samples – objects in a relational database. We use features similar to descriptors from MPEG-7, but also consider the time series framework, by taking as new descriptors temporal clusters and patterns observed for particular features. Experience from both signal analysis and other data mining applications suggests us to use additional techniques for automatic new feature extraction as well.

The most important for further research is to perform more experiments with classification of new cases by basing on decision models derived from training data in terms of introduced data structure. It seems that the need of transformation is obvious in case of attributes which are neither numerical nor discrete, e.g. when objects are described by time series. In the future we plan to combine the clustering methods with time trends analysis, to achieve an efficient framework for expressing and deriving the dynamics of the changes of complex feature values in time.

Acknowledgements

Supported by Polish National Committee for Scientific Research (KBN) in the form of PJIIT Project No. (nasz nowy grant extrakcyjny)

References

Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., and Verkamo, I. (1996). Fast Discovery of Association Rules. In *Proc. of the Advances in Knowledge Discovery and Data Mining* (pp. 307–328). AAAI Press / The MIT Press, CA.

Agrawal, R., Srikant, R. (1994). Fast Algorithms for Mining Association Rules. In *Proc. of the VLDB Conference*, Santiago, Chile.

Ando, S. and Yamaguchi, K. (1993). Statistical Study of Spectral Parameters in Musical Instrument Tones. *J. Acoust. Soc. of America*, 94(1), 37–45.

Batlle, E. and Cano, P. (2000). Automatic Segmentation for Music Classification using Competitive Hidden Markov Models. *Proceedings of International Symposium on Music Information Retrieval.* Plymouth, MA. Available at http://www.iua.upf.es/mtg/publications/ismir2000-eloi.pdf.

Bazan, J. G., Nguyen, H. S., Nguyen, S. H, Synak, P., and Wróblewski, J. (2000). Rough Set Algorithms in Classification Problem. In L. Polkowski, S. Tsumoto, and T.Y. Lin (Eds.), *Rough Set Methods and Applications: New Developments in Knowledge Discovery in Information Systems.* Physica-Verlag, 49–88.

Bazan, J. G. and Szczuka, M. (2000). RSES and RSESlib - A collection of tools for rough set computations. In W. Ziarko and Y. Y. Yao (Eds.), *Proc. of RSCTC'00*, Banff, Canada. See also: http://alfa.mimuw.edu.pl/˜rses/.

Bazan, J. G., Szczuka, M., and Wróblewski, J. (2002). A New Version of Rough Set Exploration System. In *Proc. of RSCTC'02*. See also: http://alfa.mimuw.edu.pl/˜rses/.

Beauchamp, J. W., Maher, R., and Brown, R. (1993). Detection of Musical Pitch from Recorded Solo Performances. 94th AES Convention, preprint 3541, Berlin.

Box, G. E. P., and Tiao, G. C. (1992). Bayesian Inference in Statistical Analysis. Wiley.

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

Brown, J. C. and Zhang, B. (1991). Musical Frequency Tracking using the Methods of Conventional and 'Narrowed' Autocorrelation. *J. Acoust. Soc. Am.*, 89, 2346–2354.

Brown, J. C., Houix, O., and McAdams, S. (2001). Feature dependence in the automatic identification of musical woodwind instruments. *J. Acoust. Soc. of America*, 109, 1064–1072.

Cook, P. R., Morrill, D., and Smith, J. O. (1992). An Automatic Pitch Detection and MIDI Control System for Brass Instruments. Invited for special session on Automatic Pitch Detection, Acoustical Society of America, New Orleans.

Cooper, D. and Ng, K. C. (1994). A monophonic pitch tracking algorithm. Available at http://citeseer.nj.nec.com/cooper94monophonic.html.

Cosi, P., De Poli, G., and Lauzzana, G. (1994). Auditory Modelling and Self-Organizing Neural Networks for Timbre Classification *Journal of New Music Research*, 23, 71–98.

de la Cuadra, P., Master, A., and Sapp, C. (2001). Efficient Pitch Detection Techniques for Interactive Music. *ICMC*. Available at http://www-ccrma.stanford.edu/ pdelac/PitchDetection/icmc01-pitch.pdf.

Doval, B. and Rodet, X. (1991). Estimation of Fundamental Frequency of Musical Sound Signals. *IEEE*, A2.11, 3657–3660.

Düntsch I., Gediga G., and Nguyen H. S. (2000). Rough set data analysis in the KDD process. In *Proc. of IPMU 2000*, 1, (pp. 220–226). Madrid, Spain.

Eronen, A. and Klapuri, A. (2000) Musical Instrument Recognition Using Cepstral Coefficients and Temporal Features. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2000* (753–756). Plymouth, MA.

Fujinaga, I. and McMillan, K. (2000). Realtime recognition of orchestral instruments. *Proceedings of the International Computer Music Conference* (141–143).

Herrera, P., Amatriain, X., Batlle, E., and Serra X. (2000). Towards instrument segmentation for music content description: a critical review of instrument classification techniques. In *Proc. of International Symposium on Music Information Retrieval (ISMIR 2000)*, Plymouth, MA.

ISO/IEC JTC1/SC29/WG11 (2002). MPEG-7 Overview. Available at http://mpeg.telecomitalialab.com/standards/mpeg-7/mpeg-7.htm.

Kaminskyj, I. (2000). Multi-feature Musical Instrument Classifier. *MikroPolyphonie* 6 (online journal at http://farben.latrobe.edu.au/).

Kostek, B. and Czyzewski, A. (2001). Representing Musical Instrument Sounds for Their Automatic Classification. *J. Audio Eng. Soc.*, 49(9), 768–785.

Kostek, B. and Wieczorkowska, A. (1997). Parametric Representation of Musical Sounds. *Archive of Acoustics*, 22(1), Institute of Fundamental Technological Research, Warsaw, Poland, 3–26.

Lindsay, A. T. and Herre, J. (2001). MPEG-7 and MPEG-7 Audio – An Overview. *J. Audio Eng. Soc.*, 49(7/8), 589–594.

Liu, H. and Motoda, H. (Eds.) (1998). *Feature extraction, construction and selection – a data mining perspective.* Kluwer Academic Publishers, Dordrecht.

Mannila, H., Toivonen, H., and Verkamo, A. I. (1998). Discovery of frequent episodes in event sequences. Report C-1997-15, University of Helsinki, Finland.

Martin, K. D. and Kim, Y. E. (1998). 2pMU9. Musical instrument identification: A pattern-recognition approach. 136-th meeting of the Acoustical Soc. of America, Norfolk, VA.

Mitchell, T. (1998). Machine Learning. Mc Graw Hill.

Nguyen, H.S.: Discretization od Real Value Attributes: Boolean Reasoning Approach. Rozprawa doktorska. Uniwersytet Warszawski (1997).

Nguyen H. S. (1997). Discretization od Real Value Attributes: Boolean Reasoning Approach. Ph.D. Dissertation, Warsaw University, Poland.

Nguyen S. H. (2000). Regularity Analysis And Its Applications In Data Mining. Ph.D. Dissertation, Warsaw University, Poland.

Opolko, F. and Wapnick, J. (1987). MUMS – McGill University Master Samples. CD's.

Pawlak, Z. (1991). *Rough sets – Theoretical aspects of reasoning about data.* Kluwer Academic Publishers, Dordrecht.

Peeters, G., McAdams, S., and Herrera, P. (2000). Instrument Sound Description in the Context of MPEG-7. In *Proc. International Computer Music Conf. (ICMC'2000)*, Berlin. Av. at http://www.iua.upf.es/mtg/publications/icmc00-perfe.pdf

Polkowski, L. and Skowron, A. (Eds.) (1998). *Rough Sets in Knowledge Discovery* **1**, **2**. Physica-Verlag, Heidelberg.

Pollard, H. F. and Jansson, E. V. (1982). A Tristimulus Method for the Specification of Musical Timbre. *Acustica*, 51, 162–171.

Ślęzak, D. (2001). Approximate decision reducts. Ph.D. thesis, Institute of Mathematics, Warsaw University.

Ślęzak, D., Synak, P., Wieczorkowska, A. A., and Wróblewski, J. (2002). KDD-based approach to musical instrument sound recognition. In M.-S. Hacid, Z. W. Ras, D. Zighed, and Y. Kodratoff (Eds.), *Foundations of Intelligent Systems* (pp. 29–37), LNCS/LNAI 2366, Springer.

Ślęzak, D. and Wróblewski, J. (1999). Classification algorithms based on linear combinations of features. In *Proc. of PKDD'99* (pp. 548–553). Praga, Czech Republik: LNAI 1704, Springer, Heidelberg. Available at http://alfa.mimuw.edu.pl/˜jakubw/bibliography/bibliography.html.

Synak, P. (2000). Temporal templates and analysis of time related data. In W. Ziarko and Y. Y. Yao (Eds.), *Proc. of RSCTC'00*, Banff, Canada.

Toiviainen, P. (1996). Optimizing Self-Organizing Timbre Maps: Two Approaches. *Joint International Conference, II Int. Conf. on Cognitive Musicology* (pp. 264–271), College of Europe at Brugge, Belgium.

Wieczorkowska, A. A. (1999a). The recognition efficiency of musical instrument sounds depending on parameterization and type of a classifier. PhD thesis (in Polish), Technical University of Gdansk, Poland.

Wieczorkowska, A. (1999b). Rough Sets as a Tool for Audio Signal Classification. In Z. W. Ras, A. Skowron (Eds.), *Foundations of Intelligent Systems* (pp. 367–375). LNCS/LNAI 1609, Springer.

Wieczorkowska, A. A. and Raś, Z. W. (2001). Audio Content Description in Sound Databases. In N. Zhong, Y. Yao, J. Liu, and S. Ohsuga (Eds.), *Web Intelligence: Research and Development* (pp. 175–183). LNCS/LNAI 2198, Springer.

Wróblewski, J. (2000). Analyzing relational databases using rough set based methods. In *Proc. of IPMU'00* **1** (pp. 256–262), Madrid, Spain. Available at http://alfa.mimuw.edu.pl/˜jakubw/bibliography/bibliography.html.

Wróblewski, J. (2001a). Ensembles of classifiers based on approximate reducts. Fundamenta Informaticae **47** (3,4), IOS Press (pp. 351–360). Available at http://alfa.mimuw.edu.pl/˜jakubw/bibliography/bibliography.html.

Wróblewski, J. (2001b). Adaptive methods of object classification. Ph.D. thesis, Institute of Mathematics, Warsaw University. Available at http://alfa.mimuw.edu.pl/˜jakubw/bibliography/bibliography.html.