Inteligencia Artificial 22(64) (2019), 14-35
doi: 10.4114/intartif.vol22iss64pp14-35

INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

Feature Learning with Multi-objective Evolutionary Computation in the generation of Acoustic Features

José Menezes[1,A], Giordano Cabral[1,2,B], Bruno Gomes[2,C], Paulo Pereira[2,D]

[1]UFRPE - Federal Rural University of Pernambuco, Recife 52171-900, Brazil
[2]UFPE - Federal University of Pernambuco, Recife 50670-901, Brazil

[A]joseantonio.menezes@ufrpe.br, [B]grec@cin.ufpe.br, [C]btmg@cin.ufpe.br, [D]prps@cin.ufpe.br

Abstract To choice audio features has been a very interesting theme for audio classification experts. This process is probably the most important to solve the classification problem. In this sense, techniques of *Feature Learning* generate attributes more appropriate for classification model. Generally these techniques do not depend on knowledge domain and can apply in various types of data. Yet, less agnostic approaches learn a knowledge restricted to the area studded and audio data requires a specific knowledge. Many techniques aim to improve the performance in generation of new acoustic features, among there is the technique based in evolutionary algorithms to explore analytical space of function. Despite the efforts made, there are still opportunities for improvement. This work proposes and evaluates a multi-objective alternative to the exploitation of analytical audio features. Experiments were arranged to validate the method, with the help a computational prototype implementing the proposed solution. Then it was verified the model effectiveness and was shown there is still opportunity for improvement in the chosen segment.

Resumen Elegir características de audio ha sido un tema muy interesante para los expertos en clasificación de audio. Este proceso es probablemente el más importante para resolver el problema de clasificación. En este sentido, las técnicas de Feature Learning generan atributos más apropiados para el modelo de clasificación. En general, estas técnicas no dependen del dominio del conocimiento y pueden aplicarse a diversos tipos de datos. Sin embargo, los enfoques menos agnósticos aprenden un conocimiento restringido al área tachada y los datos de audio requieren un conocimiento específico. Muchas técnicas tienen como objetivo mejorar el rendimiento en la generación de nuevas características acústicas, entre ellas, la técnica basada en algoritmos evolutivos para explorar el espacio analítico de la función. A pesar de los esfuerzos realizados, todavía hay oportunidades de mejora. Este trabajo propone y evalúa una alternativa multi-objetivo a la explotación de las características de audio analíticas. Se organizaron experimentos para validar el método, con la ayuda de un prototipo computacional que implementó la solución propuesta. Luego se verificó la efectividad del modelo y se mostró que todavía hay oportunidades de mejora en el segmento elegido.

Keywords: Automatic audio classification, feature learning, analytical space, evolutionary algorithms, multi-objective optimization.
Palabras clave: Clasificación automática de audio, aprendizaje de características, espacio analítico, algoritmos evolutivos, optimización multiobjetivo.

ISSN: 1137-3601 (print), 1988-3064 (on-line)

1 Introduction

Automatic Audio Classification (AAC) is a subject of great interest to Music Information Retrieval (MIR) specialists. The process involves many concepts of computational intelligence, among them Feature Learning [1].

It is known the choice of good features is determinant for the efficiency of classification tasks, because they reasonably delimit each class of the problem [2], [3]. However, it is not always easy to determine the best features to solve a problem. Therefore, strategies for designing new features are important. There is an interest in Feature Learning techniques, such as: PCA (*Principal Components Analysis*) [4], ICA (*Independent Components Analysis*) [5] and *Deep Learning* [6], among others. Feature Learning is also important in reducing of extraction cost and overfitting, since it reduces the data dimension and allows the classification mechanism to generalize observations.

Due to the complexity of the real AAC problems, the process of composing acoustic attributes often becomes handcrafted, demanding specialized knowledge and design time. Consequently, analytical approaches are promising since they dispense specialist knowledge. They are scalable, less costly and are adaptable for reuse.

Beside providing good results, it is our interest that the search for good features be intelligible. Although the aforementioned Feature Learning alternatives are effective, they are also generic in the field of knowledge which they can be applied, being useful in a wide range of fields: image processing, video, sensors, etc. But the audio has particularities that suggest other alternatives, which can obtain significant gains in terms of classification performance, usability, project or execution time.

It is in this context that techniques of audio feature learning arise, such as the automatic exploration of the analytical space of acoustic attributes (Section 3). In this sense, EDS (Extractor Discovery System) [7] stands out. It uses genetic programming to explore an analytic space of functions and find new audio features. The technique evolves individually features and return the featureset explored by the search. This approach has stagnated, despite the evolution over the following years [8], [9]. Possibly due to advances of Deep Learning, which has been increasingly applied in MIR problems [10]. Deep Learning is feature learning capable and is understood in the context of this work as a Feature Learning alternative. However, it is presented as a black-box option, and it is not possible to extract meaning from the features learned. Then it not cooperate with specialists to understand what characterizes the classes of his problem.

In order to design intelligible acoustic features, we identify improvement opportunities in the EDS. They involve the use of multi-objective evolutionary algorithms, which can restore the innovative character of the solution and to remove from the inertia the analytical development of acoustic attributes.

Some AAC problems require the exclusive minimization of the false positives or negatives. Simple genetic algorithm methods, such as EDS, do not meet this need. By these methods, the goal can only be achieved by a side effect: the solution obtains good accuracy, consequently minimizing errors. We believe this particularity can be treated more narrowly with multi-objective algoritms.

In addition, EDS process involves two steps in choosing attributes: optimization of audio features and selection of the most relevant them. The use of multi-objective algorithms also possibility to simplify the process of choosing features, simultaneously performing tasks that EDS does in two stages.

For the purpose of to achieve less agnostic approaches of feature learning, we have the questions: "Is it possible to improve the performance of an audio featureset used in AAC tasks through the multi-objective optimization of this set?", "Can it help in the exclusive minimization of false positives or negatives?". This work proposes to answer these questions. It analyzes the potential of multi-objective algorithms in the generation of audio features, composed according to the EDS model.

It was necessary to develop a computational tool that implemented such requirements, because there is no such proposal in the literature. It was compared with a mono-objective technique (EDS-like) and another domain-independent feature learning technique (PCA). We were able to confirm the initial hypothesis that it is possible to obtain better results with the approach proposed. It was also possible to note the potential to improve objectives related to the matrix of confusion.

Experiment was carried out with a database of a real problem: shot recognition. It was verified that the use of multi-objective optimization of audio features can improve the accuracy of the classification model. Besides it being more adapted to the specific needs of the problems.

2 Problem Details

In applications that demand intelligence in digital audio manipulation, AAC has been widely used. Figure 1 presents an ontology of subproblems and their respective applications.

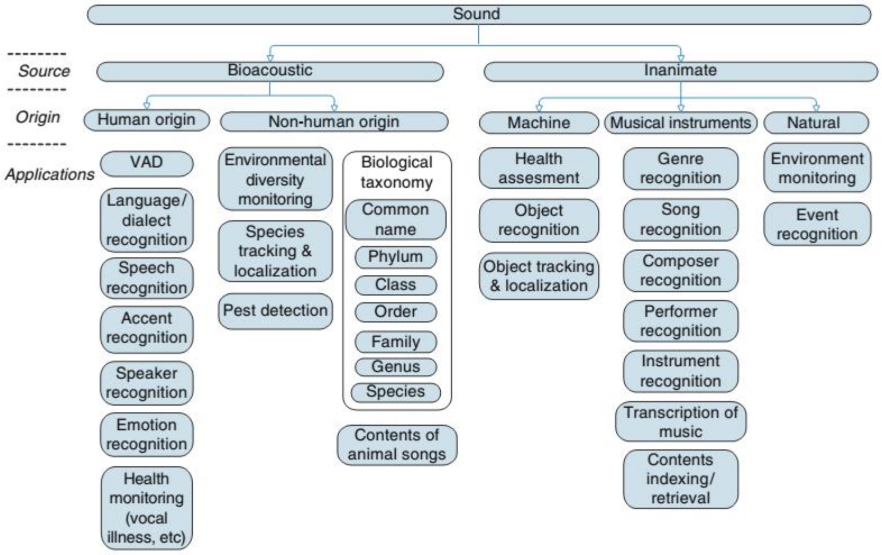


Figure 1. Taxonomy of sounds in the perspective of human applications [11]

The applications are in areas such as telecommunications, security, health, biology, music and so on. These applications include the detection of vocal activity [12], speech recognition [13], languages detection [14], speaker recognition [15], emotion recognition [16], health monitoring (vocal chords, respiration, etc.), localization of sound sources [17], pest detection [18], biological taxonomy [19], animal communication, audio monitoring, sound event recognition such as baby crying [20] and a range of musical applications. The latter are promising due to the increasing amount of musical content produced and shared. These applications include the recognition of musical styles [21] music recognition, recognition and modeling of composers styles, detection of musical instruments [22], separation of sources, musical transcription (chords, notes, rhythms, attacks, rhythm, tonality) [23] and so on [11].

With regard to the implementation of a new classifier model, three steps are involved: selection / extraction of audio features, training and validation. In the selection of features, the sound must be represented by a set of attributes (wave frequency, frequency power, spectral, etc.) [24]. In traditional methods this step works with pre-defined audio attributes. After that, those which best represent problem classes are chosen to train a new classifier using some audio base. Finally the obtained model is validated using another base.

Choice of good features is decisive for the mapping between audio samples and classes of the problem. However, some Feature Learning strategies are particular since some audio attribute do not apply to images, texts, or any other form of data.

Classifier, in turn, is chosen according to its efficiency. But the determination of which is the best one is not exact and depends on the nature of the problem, as well as sound attributes involved. To search for the best classifier one can optimize the hyper-parameters of classification algorithms, a solution investigated by [25], which employs SMAC (Sequencial Model-based Algorithm Selection) [26] to find the best classifier and fit its parameters.

The efforts required in feature selection and classifier selection result in two distinct approaches:

- *Bag-of-Frame* [27], [28]: Here the sound samples are divided into frames. For each frame a feature vector is computed. These vectors are aggregated (hence the bag) and used in the rest of the process: selection of feature subset, training and validation of the classifier. Currently this approach serves a wide range of problems such as musical genre classification, instrument recognition, nasality detection, voice or mood identification; etc.

- *Ad-hoc* [8]: Although Bag-of-Frame to be efficient in many cases, problems of lesser abstraction are more difficult to solve with it. For example, it is easy to distinguish between Rock and Jazz, but it is difficult to distinguish between Be-bop and Hard-bop (subgenres of the Jazz), because there is a lot of similarity and subtle differences between styles. It is hard even for humans. An Ad-hoc approach aims to devise a new features set that make this distinction possible. This can be done by applying two or more functions on the signal (e.g. applying a filter on the signal and then an FFT to get the maximum power). Disadvantages are that this process requires specialist knowledge, is costly and is by trial and error. Moreover, reuse is rare. Therefore the analytical features obtained by this process are unlikely to serve another problem.

It is in this context that techniques of automatic generation of features gains importance, especially when they use evolutionary algorithms [29], [30], [7] and achieve optimal solutions with less computational effort.

2.1 Analytical space of audio features

In an audio classification approach that uses generic features (preexisting high-level attributes, e.g.: Zero Crossing, Root Main Square – RMS, Mel Frequency Cepstral Coefficient – MFCC, etc.) [24] there is no concern with evolution these features. Designer only selects those most relevant using dimensionality reduction techniques. We will call the preexisting feature set of generic space.

Mean difference in the use of analytical features, in contrast to the previous approach, is that they consist of attributes generated from the combination of two or more generic features. Interest is to improve their performances through analysis. So another approach would be the *heuristic search in the analytic space*, which aims to find new audio features starting from a generic set.

As the analytical space has high complexity, not every method is feasible to search for the optimal attributes. Evolutionary algorithms are interesting option. In audio feature learning, the use of Evolutionary Computing is still very little explored. However the theme is comprehensive and its potential in AAC has been neglected. Results from theme exploration are particularly interesting because they are specific to the knowledge domain of MIR such as these used by EDS (Extractor Discovery System) [31], [7], [8].

There are three challenges to AAC that are relevant to this paper. They are: the need to find meaningful features; the difficulty in designing them and the possibility of improving in Evolutionary Computing as a method of solving these problems.

The *need* to find new audio features is due to preexisting ones often do not satisfactorily solve restricted classification problems. It is necessary, but costly, to design manually acoustic features more appropriate to the nature of the problem. This artisanal design do not guarantee of satisfactory solutions. And even if one arrives at good attributes, they can hardly be reused in another classification problem. Hence the need for an automatic and fast method of search these features.

Main *difficulty* involves the infinity search space [8] since it is possible to combine generic features and modify its parameters countless times. This makes the optimal solution unlimited and its viability is its viability is method-dependent. In the end, this task can be summarized in a process of trial and error, even if computational methods are used. In this sense, evolutionary algorithms are good options to approach the optimal solution, due to its heuristic nature.

In addition to the previous problems, it is assumed that it is also possible to improve the efficiency in solving any classification problems through the search for new attributes. Because in an audio classification problem, the search space is infinite, so hypothetically, it is possible to find better features than the current ones [8]. Indeed this hypothesis was validated in the works related to EDS [31], [7], [8].

Implementations such as EDS are *possible to improve*. They use a mono-objective algorithm and lack an objective way of treating peculiarities. Often AAC tasks are not only interested in improving the accuracy of the results, but solving the problem in such a way that certain constraints are met. Failure to meet these constraints may even have serious consequences, depending on the application [25]. Take a health monitoring system: the system may even fail to emit a false alarm about a patient's situation, but, depending on what is being monitored, can not fail to send the true alarm in the occurrence of some abnormal event . Thus, the system accuracy is not the main aspect to be improved. In this case the optimization should be directed to reduce errors related to true alarms. And the features that must emerge from the search process must be able to define precisely the critical events, satisfying the constraints.

Moreover, EDS approach needs a complementary process. At the end of the search there is a considerable number of attributes. But it is still necessary to select the most relevant ones. Selection methods are still required. In a strategy that seeks to evolve the feature set and not only one, at the end of the process already it has the best feature set. No post-optimization selection methods are required. Multi-objective evolutionary algorithms are presented as a good alternative to these questions.

2.2 Expected Solution

We expect that a solution will find a set of intelligible attributes for AAC tasks in a viable time. The solution must use evolutionary multi-objective computational techniques.

Among other satisfaction criteria to be achieved are the following:

- *Correctness*. Strategy of optimizing must to improvement accuracy of the classifier model;
- *Adequacy*. Expected solution must to satisfy the constraints of the problems .
- *Reusability and scalability*. It is unusual to apply the audio features of one problem to another, but technique to find them must be adaptable to problems of any kind and scale;
- *Economy of knowledge*. Solutin must be independent of specialized knowledge about signal processing. It would be more accessible to developers of intelligent audio technologies.

We did not find in the state of the art a solution that contemplates all these criteria. Thus we propose alternative one. The following sections present the state of the art, the proposed solution and its validation.

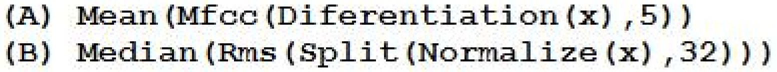
3 State of Art

This section presents Extractor Discovery System. The only solution within the scope of the research: evolutionary computation methods for analytical space exploration of acoustic functions.

3.1 Extractor Discovery System (EDS)

The EDS was developed in the laboratory of Sony CSL in Paris and presents a good proposal in audio feature learning using genetic programming techniques [32].

Starting from a finite set of elementary operators, e.g.: Mathematicians (addition, multiplication by scalar, mean, etc.); signal processing (Fourier transform, filters, spectral centroid, etc.); specific for music (Pitch or ltas), it is possible to combine these operators in a valid way. It obtain expressions as in figure 2 in which (A) can be understood as *the mean of the first five ceptral coefficients (MFCC) of the derivative of the signal 'x'.* And (B) is *Median energy value (RMS) of 32 splits of the normalized signal 'x'.* Combination of these operators defines a new audio feature, also called a function.



```
(A) Mean(Mfcc(Diferentiation(x),5))
(B) Median(Rms(Split(Normalize(x),32)))
```

Figure 2. Feature example [8].

3.1.1 Typing Rules and Heuristic

Design of functions is controlled by two mechanisms: typing and heuristics. Typing rules control combination of operators guaranteeing that their inputs and outputs have the same data types. For example, an FFT will receive an input an acoustic signal and will transform into a spectral output, or vice versa. A Mean receives any sequence of information and transforms it into a scalar. Thus, EDS can generate: *fft(HpFilter(x))*, but not *fft(mean(x)*.

Heuristics represent the specialized knowledge of signal processing professionals. It allows to bet on some functions without having to calculate their performance. It is also characteristic of the mechanism to prevent formations of unnecessary functions, such as *fft(fft(fft(fft(x))))* [7], [8].

3.1.2 Generic Operators and Pattern

Typing system allows the creation of *generic operators* (it is not generic feature), which are regular expressions that support one or more operators and form functions whose type of output is forced [7]. For example: the operator “*_a (x)” suggests a combination of several operators whose output type is a scalar “a”. In the case *Square (Mean (x))* is a valid function to satisfy the operator.

EDS implements three generic operators:

- "?_T" points to 1 operator whose output type is "T".
- "*_T" points to several operators whose output types are all "T".
- "!_T" points to several operators whose only final output is of the type "T"

It's possible to define patterns of functions as: “?_a (!_Va (Split (*_t:a (SIGNAL))))” which supports the following functions and so on [7]:

- *Sum_a (Square_Va (Mean_Va(Split_Vt:a (HpFilter_t:a(SIGNAL_t:a, 1000Hz), 100))))* or
- *Log10_a (Variance_a (NPeaks_Va(Split_Vt:a (Autocorrelation_t:a(SIGNAL_t:a), 100), 10)))*.

3.1.3 Mechanisms of Genetic Algorithm

Genetic algorithm seeks to "evolve" a population of individuals in order to find those most apt. For this it uses operations of recombination, mutation and selection. In the context of this work, given a random population of audio features (individuals), it is possible to improve the quality of these attributes by applying successive genetic operations.

Operations are shown through the examples that follow.

Starting from the expression *Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))* the system executes the following operations:

- Cloning – It changes parameters of some functions.

  E.g. Before: *Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))*

  After: *Sum (Square (Mean (Split (HpFilter (SIGNAL,* ***430Hz***, ***65ms***))))

- Mutation (head or tail) – It changes part of head or tail of the expression.

  E.g. Before: *Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))*

  After: ***Max (Max*** *(Split (HpFilter (SIGNAL, 430Hz), 65ms))))*

- Deletion – It removes any function from the expression.

  E.g. Before: *Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))*

  After: *Sum (Mean (Split (HpFilter (SIGNAL, 500Hz), 50ms)))*.

- Addition – It adds any function to the expression head.

  E.g. Before: *Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))*

  After: **Log** (Sum (Square (Mean (Split (HpFilter (SIGNAL, 500Hz), 50ms)))))

- Replacement – It swaps one function for any other with equivalent type.

  E.g. Before: *Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))*

  After: *Sum (Square (Mean (Split (****LpFilter*** *(SIGNAL, 500Hz), 50ms))))*

- Cross-overs – It recombinates two individuals to generate a new one.

  E.g. A possible result of the cross between *Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))* and the expression *Mean (**Autocorrelation (SIGNAL)**)*, can be the two individuals shown below: 1º - *Sum (Square (Mean (Split (**Autocorrelation (SIGNAL)**, 50ms))))* and 2º -*Mean (HpFilter(SIGNAL, 500Hz))*

It is not scope of this work to explain the operators of the examples showed. Many of them can be found in [24].

In addition to the operators shown, EDS defines algorithm meta-parameters (number of generations, population size and etc.). Finally, to determine when a feature in question is better than another, the system defines the fitness metric. It uses Fisher Discriminant Ratio [33] or some classifier model that, according to its accuracy, an fitness is attributed to input feature.

3.1.4 Global Algorithm

Implementation of EDS is organized in two parts: Learning of new attributes by genetic algorithm and the selection of relevant features resulting from this process. Figure 3 illustrates this, while figure 4 presents pseudocode of the algorithm implemented by EDS.

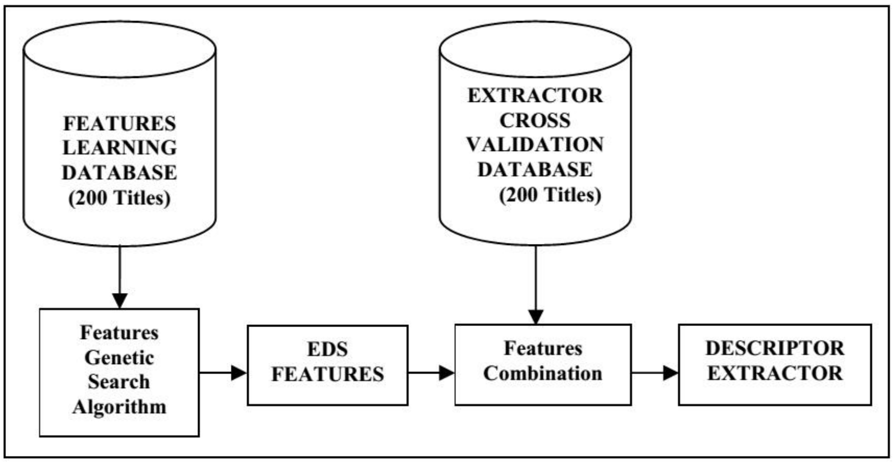
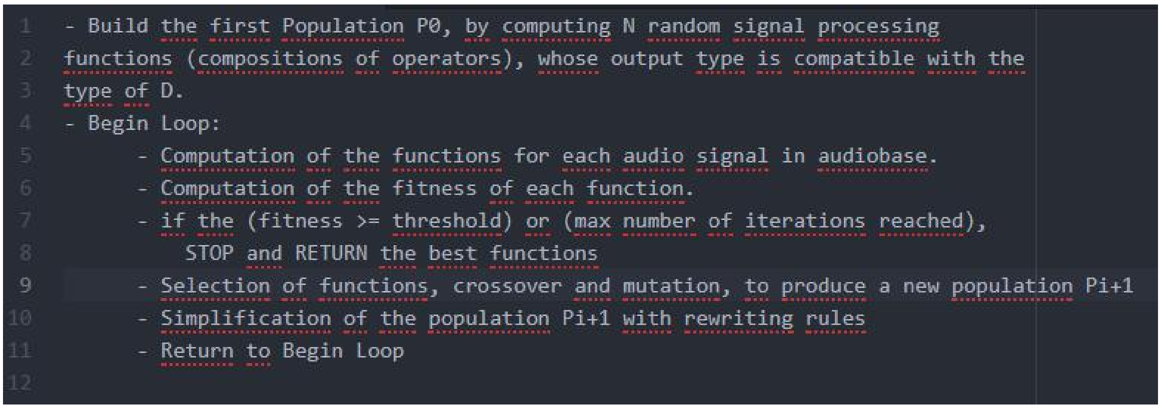

FEATURES LEARNING DATABASE (200 Titles)
EXTRACTOR CROSS VALIDATION DATABASE (200 Titles)
Features Genetic Search Algorithm
EDS FEATURES
Features Combination
DESCRIPTOR EXTRACTOR

Figure 3. Global architecture of the Extractor Discovery System [7].



```
- Build the first Population P0, by computing N random signal processing
functions (compositions of operators), whose output type is compatible with the
type of D.
- Begin Loop:
      - Computation of the functions for each audio signal in audiobase.
      - Computation of the fitness of each function.
      - if the (fitness >= threshold) or (max number of iterations reached),
        STOP and RETURN the best functions
      - Selection of functions, crossover and mutation, to produce a new population Pi+1
      - Simplification of the population Pi+1 with rewriting rules
      - Return to Begin Loop
```

Figure 4. Global algorithm of the EDS [7].

3.1.5 Solution requirements met by EDS

According to the satisfaction criteria for the expected solution (Section 2.2) EDS improves the classification results (*correctness*); it serves for different classification problems (*reusability and scalability*); it can dispense specialized knowledge in execution (*economy*). However, it fails in the *adequacy* criteria, since it does not allow the development of solutions with satisfaction of constraints.

For AAC problems that need to satisfy constraints a solution is to use multi-objective genetic algorithms.It evolves the accuracy of the classification at the same time that it fits the constraint to be satisfied. For example, in a voice identification problem to access control of some system. It is tolerable that rarely access control fails to identify the voice of a registered person and denies permission (false negative), but it is intolerable that the system allows access to any unregistered person (false positive) due to a failure in their voice identification process. Thus, beside to improve the classification accuracy it is also necessary to reduce the occurrence of false positives (or false negatives, depending on the nature of the problem). A simple genetic algorithm does not make it possible to improve these two aspects of the same problem. But multi-objective strategy allows to seek both aspects during the process.

In addition to the satisfaction criteria, another gap left by EDS is that features are evaluated in isolation. Those with low fitness do not survive throughout the iterations, but there is a possibility that features lead to better classification results when are combined with others. In order to avoid wasting features with low fitness, EDS can save the generated attributes list by the genetic algorithm and apply an attribute selection technique. However, this approach does not allow low fitness features to interfere in evolution of another feature during algorithm iterations. In this way it is necessary to guarantee their survival, which suggests the individual fitness is not the only aspect to optimize by search.

Furthermore, we understand that it is possible to simplify the procedure by eliminating the Features Combination step (Figure 3).

In short Extractor Discovery System can be improved because of the following motivators:

- It do not use heuristics to satisfy false positive or negative constraints;
- It do not preserve, during genetic programming, audio attributes with low fitness;
- It is possible to simplify the process by eliminating one step.

We present a solution for automatic audio classification, using analytical attributes resulting from a multi-objective search, in order to satisfy the points listed.

4 Proposed solution

4.1 Elements

Although EDS have good results, we consider possible to improve it, either in the classification accuracy or in the constraints satisfaction. Fundamental difference between mono-objective and multi-objective optimizations in this work is that the former considers only the isolated audio feature and the latter considers a feature set. This is important because it determines how a population of individuals evolves over the generations of the algorithm. In mono-objective approach, the features with low fitness do not survive throughout the iterations, but it could be possible that these features achieve to better results when combined with others. A multi-objective approach does not allow features to be discarded because of their low individual fitness. By surviving, they can contribute to better collective outcomes.

EDS sets each individual as an audio feature. In our proposal, we represent the individual as a feature set. Figure 5 shows the isolated fitness attributed to each expression in a similar manner to EDS and collective fitness assigned to the function set. The *n-th* expression has a fitness of 0.32 (32% is its accuracy) which can easily be surpassed by others. But when combined with the others it will contribute to a collective accuracy of 0.88 (88%) and can be better than anyone of all other isolated functions. The additional fitness mesure can be relevant for the evolution of the features, since the bad feature would not persist in isolation, but because it is part of a set it survives and contributes to improve of set.

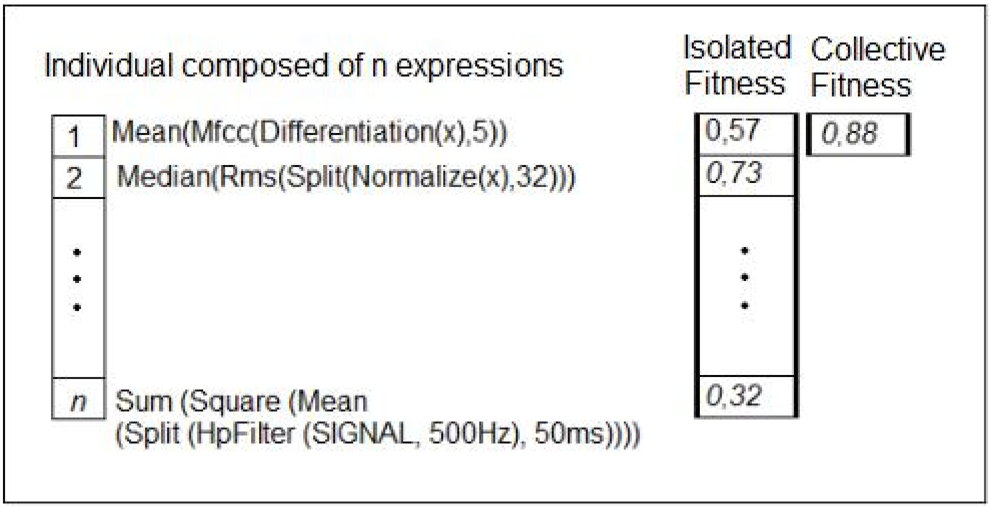

Individual composed of n expressions
Isolated Fitness
Collective Fitness
1
Mean(Mfcc(Differentiation(x),5))
2
Median(Rms(Split(Normalize(x),32)))
n
Sum (Square (Mean (Split (HpFilter (SIGNAL, 500Hz), 50ms))))
0,57
0,73
0,32
0,88

Figure 5. Ilustrative example of individual multi-objective solution and its fitness measures (between 0-1).

New evolutionary operations were devised to match with the new form of individuals representation. For example, crossing functions between an individual and another or, remove or add functions. Thus, we adapted the mono-objective operations as follows:

From an individual with three genes (functions): *1 - Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))*; *2 - (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))* *and 3 - Mean (Autocorrelation (SIGNAL))*, the system performs the following operations:

- Mutation (head or tail) – It changes entirely part of the head or tail of an individual, swapping by any other expression. E.g.:
  - *Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))*
  - ***Median(FFT (SIGNAL, 500Hz), 50ms))***
  - ***RMS (Normalize (SIGNAL))***
- Deletion – Removes any expression from the set. E.g.:
  - *Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))*
  - *Mean (Autocorrelation (SIGNAL))*
- Addition – Adds any expression to the set. E.g.:
  - *Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))*
  - *Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms)))*
  - *Mean (Autocorrelation (SIGNAL))*
  - ***RMS (Normalize(SIGNAL))***
- Replacement – Swaps an expression for any other. E.g.:
  - ***RMS (Normalize(SIGNAL))***
  - *Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))*
  - *Mean (Autocorrelation (SIGNAL))*
- *Cross-overs* – Recombinates two individuals to generate a new one. E.g.: Crossing with the individual of two genes: *A - RMS (Normalize(SIGNAL))* e *B- Median(FFT (HpFilter(SIGNAL, 500Hz), 50ms))))*, can result in two new children:

  Child 1:

  A - ***RMS (Normalize(SIGNAL))***

  2 - Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))

  3 - Mean (Autocorrelation (SIGNAL))

Child 2:

1 - Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))

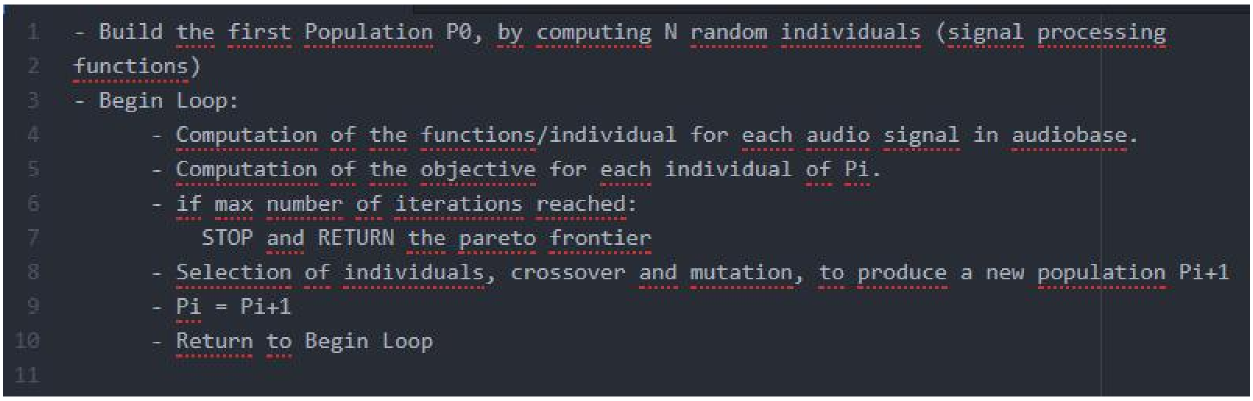
B - ***Median(FFT (HpFilter(SIGNAL, 500Hz), 50ms))))***

In addition, the mutation operations of the EDS approach could still be used as a particular type of mutation in the new algorithm since they act on a single expression.

We also conceive a restriction. It was found that throughout the generations the individuals grew tedious to be bigger, by the action of the recombinações, making the solution slow and costly. We try to circumvent this tendency by allowing a limit on the size of individuals, which are penalized when they exceed this limit. In this way, the method tends to evolve by keeping the size of individuals up to a certain threshold chosen by the user.

In what concerns the measurement of aptitude, we use the same mechanism of evaluation of individual of the previous implementation, the instance of a classifier to verify the results of fitness. Two or more goals can be defined for the problem. One of them being necessarily "increase hit rate". And the others being any of: "increase the fitness of the best feature of a individual", "increase the fitness of the 'worse' feature of the individual", "increase the distance between the fitness of the best and worst feature of the individual", "Decrease the distance between the fitness of the best and the worst feature of the individual", or restrictions such as "decrease in the number of false positives or negatives of the individual", among other objectives that can be designed specifically for each problem instance of AAC.

As these are objectives that may be in conflict, the result will be a Pareto frontier [34] (figure 6). What matters is the individual (set of audio features) that best fits the nature of the problem, that is, it will be up to the professional to decide, starting of the results achieved, which solution of the border is most appropriate for their problem.



```
- Build the first Population P0, by computing N random individuals (signal processing
functions)
- Begin Loop:
      - Computation of the functions/individual for each audio signal in audiobase.
      - Computation of the objective for each individual of Pi.
      - if max number of iterations reached:
          STOP and RETURN the pareto frontier
      - Selection of individuals, crossover and mutation, to produce a new population Pi+1
      - Pi = Pi+1
      - Return to Begin Loop
```

Figure 6. Global algorithm multi-objeticve.

In a problem of satisfying constraints, in the reduction of false positives or negatives, the result must be the Pareto frontier, being at the discretion of the developer to define which of the non-dominated individuals should be used as a solution to their problem. But in case the problem has no restrictions, what interests us is only the accuracy. Therefore, if the best value is reached by the first objective (collective fitness) of an individual then it should be used in the classifier model, but if the best value is reached by the 2nd goal (isolated fitness) of some gene (audio feature), then only this feature should be used.

4.2 Prototype

With the design of the aspects presented previously, the proposed solution covers what was not covered by the EDS, however it remains to know how effective an implementation can be that uses our approach. For this, a computational prototype was developed which we call ExpertMIR.

4.2.1 Architecture and flow

Initially ExpertMIR was developed with an EDS-like system, seeking to implement the features of the Extractor Discovery System in order to better study the behavior of such an approach, this was necessary because Sony's solution was closed, making it impossible to try. In a second moment, it was proposed an evolution of this system, using multi-objective algorithms in order to achieve some improvement.

Thus, the solution was constructed to operate in two perspectives: in the mono-objective and multi-objective optimization of audio features. Both operating on the same analytic space, having in common the operators and patterns used to generate individuals in genetic algorithms.

ExpertMIR can explore features that follow the following standards:

- " !_a (SIGNAL)" – any function whose final output is a scalar;
- " Mean(!_t:a (SIGNAL))" – the average of any function whose output is in the time domain.
- " Mean(!_f:a (SIGNAL))" – the average of any function whose output is in the frequency domain.

Also the same technological framework was used for both: the extractor system, the classifier algorithm among others. Figure 7 shows how the system modules are designed to interact.

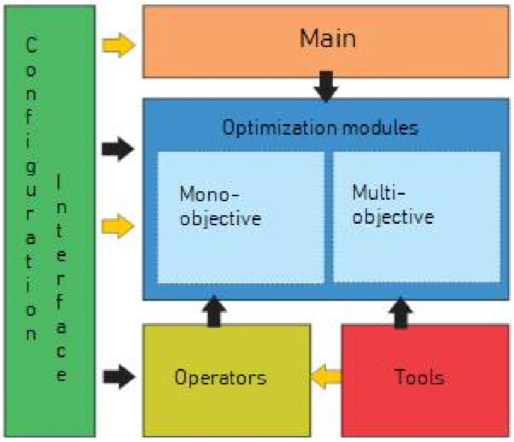


Figure 7. Architecture of the prototype's modules.

Configuration interface: Set of classes that allow the non-expert user to configure a new problem (classes, audio base, constraints or goals and so on), define the perspective in which the system will operate, have files and reports to record the result of the process.

Main: It gathers all possibilities of tool execution, controlling what can be done and how it should be done. Here the problems are instantiated and the calls of the evolutionary algorithms are made.

Mono and multi-objective optimization modules: They are the main modules of the system, responsible for executing feature learning, they are never executed at the same time. Because of the complexity and processing cost of genetic algorithms, it is important that each approach can be performed separately. In addition, not overloading the computational resource also allows better evaluation of the performance of each algorithm.

Operator Package: Here, each mathematical, signal processing and music-specific operator is implemented, which is the set of operators that determine the analytical space of expressions that the genetic searches will follow. This module of the system is scalable, it can, whenever it is necessary to add as many operators as you like, as well as activate and deactivate an operator.

Tools Package: In order to execute the genetic algorithms well according to our interest and necessity, it was necessary to develop a package of support to the algorithms, in this package it is possible to resort to several inherent tools of the process, like: algorithm classifier (used in the calculation of the fitness of individuals) (extracts the values from the converted audio samples), features validator (responsible for checking if the feature found in the search is being correctly constructed) and so on.

To understand the various transformations of data throughout a system run, the execution flow diagram is shown in figure 8.

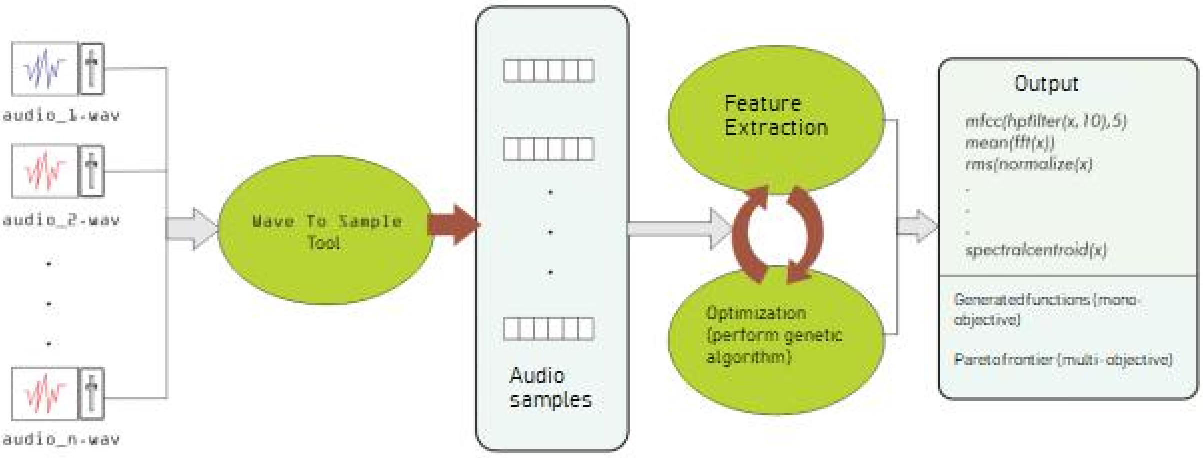


Figure 8. Flow of data manipulated in the execution of the prototype.

When looking for new audio features the solution requires an annotated audio base, since the learning is supervisioned. Files in the WAVE format representing instances of the classes of a given problem are preprocessed and represented as system objects. This is necessary to make the process faster, as, throughout the genetic interactions of learning, the information contained in the samples will be frequently requested.

The second moment is the most important step of the process and occurs with two modules constantly exchanging information. The optimization module, each generation of a population, will require the feature extractor to calculate the value of the features generated for each sample of the base. This information should be used in the individual's fitness calculation, which uses a classifier instance to perform cross-validation in the annotated database. This allows to find out how much the features found are adequate for the solution of the problem.

Finally, the result of the process is the set of features learned by the mono-objective algorithm, or the set of non-dominated solutions (Pareto frontier) of the multi-objective algorithm.

4.2.2 Technologies used

ExpertMIR was developed in Java 8 EE through the Eclipse IDE. The choice was due to the good options of complementary tools in this language and that could be used by ExpertMIR. These tools are shown below:

JMIR is a suite of Java open source applications for MIR searches. It was proposed by Cory McKay [35] in order to offer wide support the most varied applications of MIR. The jMIR allows the manipulation of the acoustic signal in digital signal format (Wave, MP3 and etc.) as well as in symbolic format (MIDI) and serves the most diverse applications as: audio data mining in the Web, audio classification, and so on. It is a complete tool in terms of MIR and reference in the.

Precisely it is the automatic audio sorting feature makes us interested in this tool. An AAC problem is solved by using two of its modules: jAudio and ACE. Each can be understood for the following purposes:

- *jAudio Feature Extractor* [36]: Module responsible for extracting audio features. It contains a set of audio and signal features (FFT, MFCC, normalization, compactness, histogram, etc.) and the possibility of saving its values for each sample of the problem in jMIR, ACE XML standard format or until even in Weka's own ARFF format [37]. Among the other features of the tool are, for example, the possibility of audio recording, execution, adjustment of sampling rate, fragmentation and normalization of the signal. It was used in this work to assist in the extraction of features because it has several of them already implemented.
- ACE (*Autonomous Classification Engine*) [2]: It is the jMIR Machine Learning module. Responsible for applying classification algorithms to the values extracted by jAudio and associating them with categories. In addition, it implements seven types of classifiers: k-NN, Naive Bayesian, Decision Tree C4.5, Multilayer Perceptron, Support Vector Machine, Adaboost and Bagging with C4.5. And also three

dimensionality reduction techniques: PCA, exhaustive search and genetic search. It was necessary in this work to instantiate the classifier used in the calculation of fitness.

In ExpertMIR, jMIR 2.4 was used, which had many audio features coded in its extractor module and also classification algorithms.

JMetal [38] is a Java framework for the development of multi-objective applications with metaheuristics. It also supports mono-objective heuristics and was useful in the specific activities from evolutionary algorithms. It was up to us to design the individual representation, the definition of the operators and the method of evaluation of individuals. JMetal still has versatility in the alternation of its algorithms (NSGA II, SPEA, etc.), which can easily be replaced and tested for when one wants to infer the impact of each one on the quality of the features generation. The version used in our implementation was 4.5.

5 Evaluation

This chapter aims to present and discuss the results obtained with the experimentation of the proposed solution. It will be describes how the tests were performed and the validation of the hypothesis that "it is possible to improve the performance of a set of audio features used in AAC activities through the multi-objective optimization of the feature set" (Section 1).

5.1 Methods

The problem of AAC to classify very similar audios needs a set of features in such a way that these audios can be differentiated in the best way. As pointed out in our state of the art, mono-objective optimization techniques have been promising in generating analytical features. However multi-objective techniques, which have not been proposed in the literature, suggest an improvement in the efficiency of feature learning, besides allowing better adaptation to problems with constraints.

This raises our research question: Is it possible to improve the performance of an audio featureset used in AAC tasks through the multi-objective optimization of this set?

The performance of the features was determined based on two aspects: 1 - the accuracy obtained through the set these; 2 - the measure of sensitivity or specificity when minimizing false negative or positive, respectively.

To answer this question, we organized an experiment. The criterion for choosing that problem was for two fundamental reasons: to be related to a real situation difficult to solve due to the great similarity of the classes of the problem and the availability of the audio bases.

It is hoped to show that the use of multi-objective genetic algorithms has a strong indication of effectiveness among other learning techniques, which does not mean that the solution of this work must always overcome others, but it arouse the interest of the scientific community for exploration possibilities would already be an excellent contribution.

The techniques in comparison were three: the search in the analytical space with an evolutionary algorithm mono-objective (algorithm EDS-Like), the search in the analytical space with multi-objective evolutionary algorithm (ExpertMIR) and the Feature Learning with PCA (jMIR). Although we are particularly interested in evolutionary strategies, we find it important to compare with some alternative outside this group in order to situate the solution among the options of the area. The choice of PCA was mainly due to its implementation being present in the technological support framework of this work, jMIR.

To ensure consistency in the comparison of approaches, a few points had to be equated. First, the quantity and type of operators used by both. A total of 9 operators were used among the mathematical and signal processing groups listed below:

- MFCC
- FFT of binary frequencies
- Normalization
- *Power Spectrum*
- *Magnitude Spectrum*

- RMS
- *Zero Crossing*
- *Spectral Centroid*
- *Spectral Roll-off*

These operators are the generic features from which will result in the analytical features of each learning process. The description of these agents can be found throughout many others in [24].

Second, it was necessary to set the maximum size of a feature for the processes that involve genetic algorithms. We consider it appropriate to adopt the size employed by [8], ten operations.

Finally, for the evaluation of the techniques it was necessary to use the same type of classification algorithm, K-NN with k = 1 and the same validation technique, cross-validation, which according [39] it is the best method of data sampling to be used in the evaluation of a classification model.

Having defined these aspects, we can assume a fair comparison between the three solution models for the problems.

Table 1. Methods used incorporating the approaches analyzed. GA: Genetic Algorithm.

Tag	Method
PCA	PCA with 1-NN classification algorithm
MO	GA mono-objective with 1-NN classification algorithm
MT1	GA multi-objective with 1-NN classification alg. (Obj2 = reduces false negative)
MT2	GA multi-objective with 1-NN classification alg. (Obj 2 = increase the accuracy of the best feature)

Table 1 shows the algorithms used to compare the techniques. We can observe that MT1 is the evolutionary algorithm whose second objective is to reduce the incidence of false negatives, which means to improve the sensitivity of the solution. MT2 is the algorithm whose second aim is to increase the accuracy of the most fit feature of a multi-objective individual. Both methods implement the solution proposed in this work, however, these different choices regarding the second objective were to see if there is difference in choosing as a secondary objective the decrease of false negatives (or positives) when the nature of the problem so requires. For this, it is necessary to compare with another method that operates with any other secondary objective (in this case MT2 seeking to optimize the accuracy of the most apt feature).

According to the Central Limit Theorem, where the distribution of sample means tends to a normal distribution as sample size n increases. Because of the size of the databases involved, we performed 10 times the first experiment (n = 10) and 30 times the second (n = 30), obtaining four samples with data about the accuracy and sensitivity of the methods and performed a series of tests through the R Studio tool [40].

For the methods with evolutionary algorithms (MO, MT1 and MT2), p = 15 and e = 1500, where p and e are, respectively, the size of the population and the number of evaluations to be made. The recombination and mutation rates were respectively 90% and 5%. MO implements a simple genetic algorithm, while MT1 and MT2 perform the NSGAII.

5.2 Experiment I: Monitoring of Environments and Safety (MES)

A security system for monitoring environments is able to recognize alert situations such as firearm shots, vehicle collision, shattered glass, shouting and so on. The audiobase used in this experiment is a closed real base, used for commercial purposes and has up to 18,000 examples labeled of the most varied types of sound for alert and safety.

Such a solution requires a wealth of knowledge covering geolocation, sound and image capture, coding, data transmission, image processing, compression, storage, and so on. Among them the classification of sounds is a fundamental activity.

For this work, the problem addressed had a balanced subset of 1,900 instances of this already fragmented base. The situation chosen is to distinguish sounds between two classes: 1 - Firearm shooting; 2 - Burst of fireworks. The strong similarity between these classes makes the problem rather difficult to solve, which suggests the use of feature optimization to improve the results of the techniques used to solve the problem. A similar question was addressed by [41] and [42] with the difference that they seek to distinguish between gun sound and any other sound. In our case, we approach a more specific problem by choosing fireworks as the second class of the problem, by the similarity between them, which is so large that classification is not easy even for the human ear. [41] and [42] go through a traditional path, selecting the most relevant generic features of a set.

In this experiment, 950 firing instances of various types of firearms (revolvers, pistols, machine guns, rifles, rifles, etc.) of various calibers were collected as opposed to 950 instances of fireworks bursts. Each with a sampling rate of 48 kHz and an average duration of 0.085 seconds.

5.2.1 Results

As regards the accuracy of the solutions, the T-test was performed for each of the samples and it was found that MT1 presented the best mean (90.67%) and it was also responsible for the highest accuracy found (92, 68%). The results obtained are illustrated in the graph of figure 9, although MT1 has the best results, MT2 closely resembles it (average accuracy = 90.57%), better accuracy = 91.74%). And although the maximum obtained in MO (91.05%) approaches the multi-objective algorithms, it is perceived that its average is just below (85.83%).

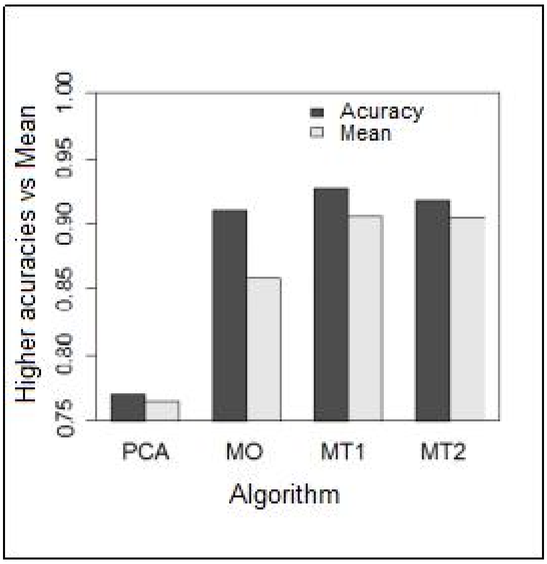


Figure 9. MES. – Maximum recorded accuracy x accuracy mean of methods.

Table 2. MES. – T-tests for each method + maximum accuracy recorded.

	Higher accuracy (%)	Mean (%)	Confidence interval (%)	Significance (%)	p-value
PCA	77	76,46	76,34– 76,6	5	2.2e-16
MO	91,05	85,83	83,67 – 87,99	5	1.34e-16
MT1	92,68	90,67	89,51 – 91,83	5	2.2e-16
MT2	91,74	90,57	89,99 – 91,15	5	2.2e-16

The boxplot (figure 10) allows us to more clearly visualize the empirical distribution of data. We can observe where the most relevant part of the sample data is concentrated and how similar the methods are studied. We observed that MT1 has slightly greater variability than MT2, and also slightly different interval, mean and median, and these solutions are similar. In the parameters chosen, the variability shown in the graph closely approximates the confidence intervals of the tests, and it is possible to be guided by it to understand the test. It is also possible, observing the interval of the methods box, to understand behaviors such as the possibility of a given being outside the mean and how far it can distance itself from it. For example, the MO test indicates that in

95% of cases its accuracy will be between 83.67 and 87.99 (fourth and fifth column of table 1), but the range of MO in the boxplot is even larger and this explains the appearance of 91.05% (its best recorded accuracy) as a predictable event. Values outside this range would be discrepant and in this case it would not be guaranteed to obtain them by repeating the experiment.

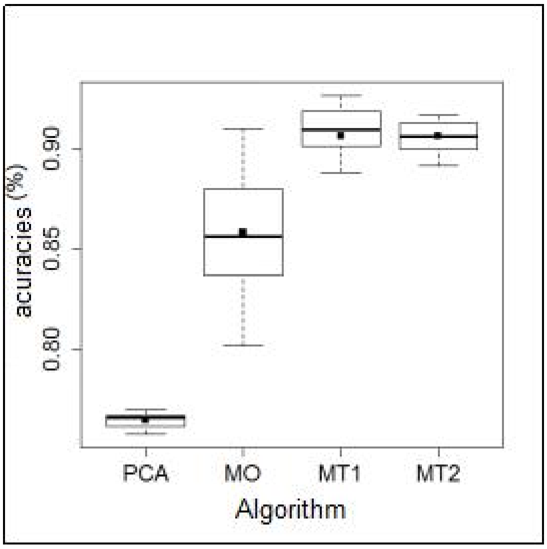


Figure 10. MES. – Boxplot of the accuracy of the methods.

The unilateral binomial test at right [43] was applied to know if, in fact, MT1 is better than MT2. We obtained the following values:

- Threshold of 50%
- Significance α = 0,05 (5%)
- p-value = 0.05469

The test reveals that possibly in more than half of the cases MT1 exceeds MT2 and still gives the probability of 80% chance of MT1 being better. Figure 11 provides these ordered data so that you can visualize the behavior indicated in the binomial test.

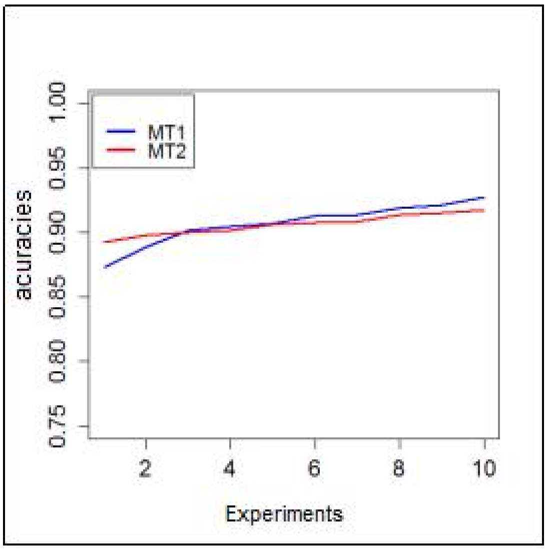


Figure 11. MES. – Accuracy of MT1 and MT2 methods.

For the database, sensitivity describes the probability that a legitimate firearm shot will be classified as a firearm shot.

The T-test was performed for each sample and it was found that the difference between MT1 and MT2 was more pronounced, showing the best means of sensitivity between the methods, 89.46% and 87.83, respectively. The other methods do not have as interesting an outcome as those achieved by multi-objective optimization (figure 12). Table 3 provides the data of the illustration.

Table 3. MES. – T-tests for each method + maximum sensitivity recorded.

	Higher sensitivity (%)	Mean (%)	Confidence interval (%)	Significance (%)	p-value
PCA	76,63	72,55	75,58 – 76	5	2.2e-16
MO	89,79	83,57	81,15 – 85,99	5	4.684e-14
MT1	92	89,46	87,83 – 91,1	5	7.575e-16
MT2	91,37	87,83	86,42 – 89,24	5	2.316e-16

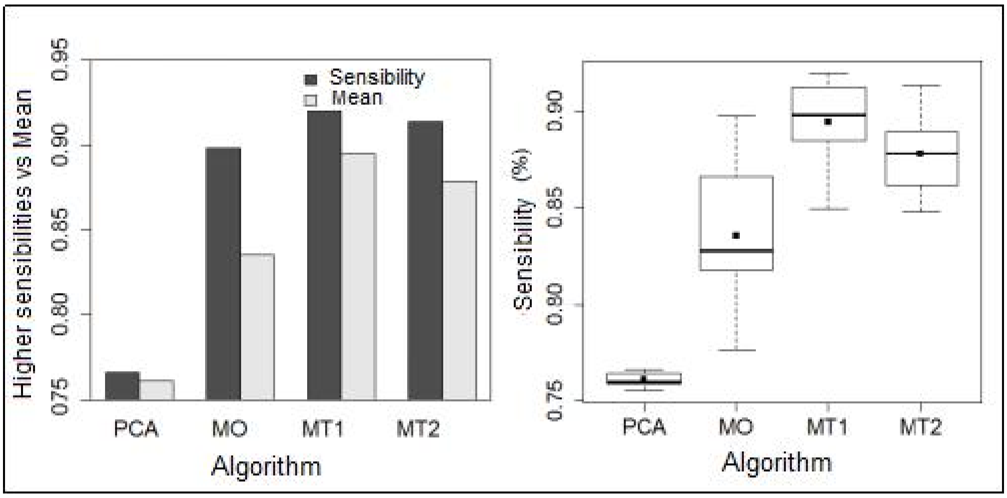


Figure 12. MES. – Sensitivity tests obtained.

Performing the binomial test for the times MT1 exceeds MT2, the following result is obtained:

- Threshold of 50%
- Significance α = 0,05 (5%)
- p-value = 0,0009766

The test is conclusive as to the hypothesis of MT1 to overcome MT2, the same test presents a probability of success of 100%. In figure 13 you can see their behavior in reducing false negatives. The lowest values of MT1 and MT2 are 4% and 4.3% respectively and represent the percentage of false negatives in the confusion matrix of the best instance of each sample.

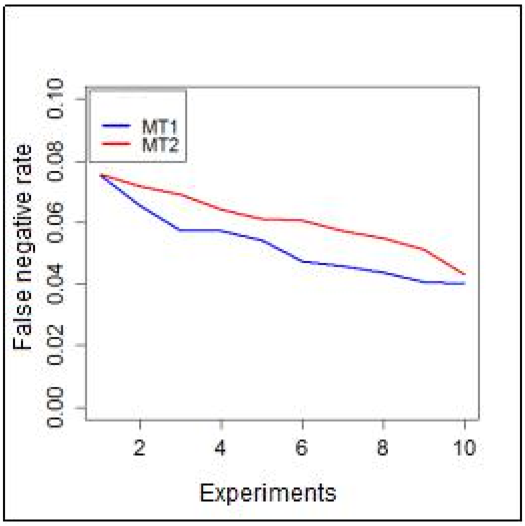


Figure 13. MES. – False Negative Rate of methods MT1 and MT2. The smaller the better.

5.2.2 Analysis

We return to our research question: "Is it possible to improve the performance of an audio featureset used in AAC tasks through the multi-objective optimization of this set?".

We have the following hypotheses that relate to the general objective explained in the introduction of this work (analyze the power of evolutionary multi-objective algorithms in the design of audio analytical features for problems of automatic audio classification) and that it concerns the measures of performance (accuracy and sensitivity for the said problem) of the research question:

Hypothesis 1: Solution accuracy.

- H0: The use of the multi-objective optimization model does not improve the accuracy of the classification.
- H1: The use of the multi-objective optimization model improves the accuracy of the classification.

Hypothesis 2: Solution sensitivity.

- H0: The use of the multi-objective optimization model does not improve the classification sensitivity.
- H1: The use of the multi-objective optimization model improves the classification sensitivity.

We used the MT1 method that obtained the best performance among the multi-objective methods and compared it with MO, since this was clearly the best method not to use the multi-objective technique. table 4 presents the results of each hypothesis test..

Table 4. Results of binomial tests (MT1 and MO) for each hypothesis.

	Threshold	Significance	Confidence interval	p-value
Hipótese 1	50 %	5 %	60,58 % - 100 %	0,01074
Hipótese 2	50 %	5 %	60,58 % - 100 %	0,01074

Since p-value is less than the level of significance of both problems, null hypotheses are discarded. This means that statically the multi-objective optimization method with false negative reduction (MT1) is superior to mono-target optimization in more than half of the cases. According to the same test, this effectiveness is statistically superior to 60.58% of the cases for both hypotheses and is probably 90%.

5.3 Consolidation of Results

When analyzing the methods in terms of accuracy and sensitivity, it was obtained important indications that the multi-objective solution has a significant weight in the effectiveness of the solution of AAC problems. MT1 (a multi-objective genetic algorithm in which the second objective is to reduce false negatives combined with 1-NN) was the method that obtained the best performance in each criterion of the experiment performed.

The problem addressed in the experiment is real, which shows the applicability of the method. The particularly interesting indication of how promising it can be is the approximation with the original commercial solution, discussed in [25]. In a problem of two classes: 1 - firing of firearm; 2 - another noise, the solution already marketed obtained a reduction of error to 3.57% [25] already the proposal in this work even reduced the error to 7.32% (equivalent to the accuracy of 92.68% of the method MT1 in table 2), but, there is a difference in the database as to the quantity and type of sounds, since the original problem is distinguishing firearm shots from any other sound, while the problem addressed here is to distinguish between gun shots of fire and fireworks, which are more similar classes and therefore the distinction is more difficult. It is possible that the proposed solution offers more interesting results on the same basis as the original problem. An opportunity for future study.

In addition to the statistical conclusions, some aspects can be highlighted regarding each method. Feature learning with PCA and the simple genetic algorithm are the fastest, reaching the results in minutes, whereas the multi-objective approach usually gives results in hours. However, taking into account that it is an automated process the result is more interesting than manual solutions like ad-hoc. In the end, eliminating the selection step at the end of the EDS implementation (Section 3.1.4), while simplifying the process does not leave the solution more agile.

The solution proposed in this work was the one implemented by the two MT1 and MT2 methods and is statistically superior to the others. Of course, this is only said about the ability to optimize audio features. There was no concern in optimizing the classifier algorithm, which was fixed.

Referring to the expected solution of session 2 it can be seen that the proposed solution corresponds to the expectation of the research, so to speak:

- Correctness: The solution influenced the improvement of the classification.
- Adequacy: The solution took into account the particularities of the problems, seeking the satisfaction of constraints.
- Reusability and scalability: The technique can be employed in several types of problem.
- Economy of knowledge: It is not necessary to know the nature of the audio features, it is enough only the handling of the technologies of classification.

6 Conclusions

The audio classification area is huge and diverse, where feature learning is a very important aspect. Despite the significant advances achieved with Deep Learning, this work showed that there is still room for improvement in other ways.

The main purpose of the solution is to improve the performance of the audio featureset in the classification tasks. Inspired by the Extractor Discovery System, it was dared to extrapolate the space of generic features, which are usually those employed in the extraction and selection phases, seeking to build more dynamic solutions that are not easily thought of by a specialist.

These solutions are intrinsically linked to the classification algorithm used, since no search or attempt was made to improve the classifier by adjusting its parameters. Even so, the classification produced interesting results. It is concluded that the analyzed feature optimize the classifier for that used data base. Although one type of classifier is not suitable for a given Automatic Audio Classification problem, it is possible to improve its performance through the development of analytical features. It does not mean that another type of classifier configured with other parameters can not overcome it, but that within its specifications it will be optimized. Therefore, ExpertMIR may be able to optimize all sorts of audio classifiers.

The need for tools that assist the community in the development of audio features can be met by implementations of the approach proposed by this work.

Taking into account the difficulty of process and cost in the design of analytical features and, aiming at better adapting the nature of some problems of AAC, we tried to develop an alternative that suited the needs of the area. With the consolidated results of the experiment it was possible to state the usefulness of the proposal. ExpertMIR is an open and useful solution for the optimization of acoustic features, especially when you can not count on specialists, you do not have the resources and time, or even when a solution is difficult to be designed analytically.

It is important to expand the comparison with other approaches and to do tests with other bases in order to identify any limitations of the model or to evolve it. Thus, our materials and methods are available so that others people can carry out further studies and complementary research.

Acknowledgements

This research was supported by FACEPE (Foundation of Support to Science and Technology of Pernambuco) with financial.

References

[1] BENGIO, Yoshua; COURVILLE, Aaron; VINCENT, Pascal. Representation learning: A review and new perspectives. **IEEE transactions on pattern analysis and machine intelligence**, v. 35, n. 8, p. 1798-1828, 2013.

[2] MCKAY, Cory et al. ACE: A Framework for Optimizing Music Classification. In: **ISMIR**. 2005. p. 42-49.

[3] YASLAN, Yusuf; CATALTEPE, Zehra. Audio music genre classification using different classifiers and feature selection methods. In: **18th International Conference on Pattern Recognition (ICPR'06)**. IEEE, 2006. p. 573-576.

[4] BURKA, Zak. Perceptual audio classification using principal component analysis. 2010.

[5] ERONEN, Antti. Musical instrument recognition using ICA-based transform of features and discriminatively trained HMMs. In: **Signal Processing and Its Applications, 2003. Proceedings. Seventh International Symposium on**. IEEE, 2003. p. 133-136.

[6] HUMPHREY, Eric J.; BELLO, Juan Pablo; LECUN, Yann. Moving Beyond Feature Design: Deep Architectures and Automatic Feature Learning in Music Informatics. In: **ISMIR**. 2012. p. 403-408.

[7] PACHET, François; ZILS, Aymeric. Evolving automatically high-level music descriptors from acoustic signals. In: **International Symposium on Computer Music Modeling and Retrieval**. Springer Berlin Heidelberg, 2003. p. 42-53.

[8] PACHET, François; ROY, Pierre. Exploring billions of audio features. In: **2007 International Workshop on Content-Based Multimedia Indexing**. IEEE, 2007. p. 227-235.

[9] PACHET, François; ROY, Pierre. Analytical features: a knowledge-based approach to audio feature generation. **EURASIP Journal on Audio, Speech, and Music Processing**, v. 2009, n. 1, p. 1, 2009.

[10] ZHOU, Xinquan; LERCH, Alexander. Chord Detection Using Deep Learning. In: **Proceedings of the 16th ISMIR Conference**. 2015.

[11] POTAMITIS, Ilyas; GANCHEV, Todor. Generalized recognition of sound events: Approaches and applications. In: **Multimedia Services in Intelligent Environments**. Springer Berlin Heidelberg, 2008. p. 41-79.

[12] SOHN, Jongseo; KIM, Nam Soo; SUNG, Wonyong. A statistical model-based voice activity detection. **IEEE signal processing letters**, v. 6, n. 1, p. 1-3, 1999.

[13] VARILE, Giovanni Battista; ZAMPOLLI, Antonio. **Survey of the state of the art in human language technology**. Cambridge University Press, 1997.

[14] MUTHUSAMY, Yeshwant K.; BARNARD, Etienne; COLE, Ronald A. Reviewing automatic language identification. **IEEE Signal Processing Magazine**, v. 11, n. 4, p. 33-41, 1994.

[15] LIPPMANN, R. P. Review of neural networks for speech recognition. **Neural Computation**, MIT Press, v. 1, n. 1, p. 1–38, 2016/07/27 1989. Disponível em: http://dx.doi.org/10.1162/neco.1989.1.1.1.

[16] KWON, Oh-Wook et al. Emotion recognition by speech signals. In:**INTERSPEECH**. 2003.

[17] GUO, Y. B.; AMMULA, S. C. Real-time acoustic emission monitoring for surface damage in hard machining. **International Journal of Machine Tools and Manufacture**, v. 45, n. 14, p. 1622-1627, 2005.

[18] POTAMITIS, Ilyas; GANCHEV, Todor; FAKOTAKIS, Nikos. Automatic acoustic identification of insects inspired by the speaker recognition paradigm. In: **INTERSPEECH**. 2006.

[19] LEE, C.-H.; HAN, C.-C.; CHUANG, C.-C. Automatic classification of bird species from their sounds using two-dimensional cepstral coefficients. **Audio, Speech, and Language Processing, IEEE Transactions on**, v. 16, n. 8, p. 1541–1550, 2008. ISSN 1558-7916.

[20] SAHA, B.; PURKAIT, P.; MUKHERJEE, J.; MAJUMDAR, A.; MAJUMDAR, B.; SINGH, A. An embedded system for automatic classification of neonatal cry. In: **Point-of-Care Healthcare Technologies (PHT), 2013 IEEE**. [S.l.: s.n.], 2013. p. 248–251.

[21] GOLUB, S. Classifying recorded music**. MSc in Artificial Intelligence. Division of Informatics. University of Edinburgh**, 2000.

[22] EGGINK, Jana; BROWN, Guy J. A missing feature approach to instrument identification in polyphonic music. In: **Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on**. IEEE, 2003. p. V-553-6 vol. 5.

[23] KLAPURI, Anssi; DAVY, Manuel (Ed.). **Signal processing methods for music transcription**. Springer Science & Business Media, 2007.

[24] PEETERS, Geoffroy. A large set of audio features for sound description (similarity and classification) in the CUIDADO project. Tech. Rep., IRCAM, 2004.

[25] ARAÚJO, D. F. D. E. Busca como sistema de apoio à melhoria de classificadores automáticos de áudio. 2014.

[26] HUTTER, Frank; HOOS, Holger H.; LEYTON-BROWN, Kevin. Sequential model-based optimization for general algorithm configuration. In:**International Conference on Learning and Intelligent Optimization**. Springer Berlin Heidelberg, 2011. p. 507-523.

[27] WEST, Kristopher; COX, Stephen. Features and classifiers for the automatic classification of musical audio signals. In: **ISMIR**. 2004.

[28] AUCOUTURIER, Jean-Julien; DEFREVILLE, Boris; PACHET, François. The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music. **The Journal of the Acoustical Society of America**, v. 122, n. 2, p. 881-891, 2007.

[29] RITTHOF, O. et al. A hybrid approach to feature selection and generation using an evolutionary algorithm. In: **2002 UK workshop on computational intelligence**. 2002. p. 147-154.

[30] MIERSWA, Ingo; MORIK, Katharina. Automatic feature extraction for classifying audio data. **Machine learning**, v. 58, n. 2-3, p. 127-149, 2005.

[31] CABRAL, Giordano; PACHET, François; BRIOT, Jean-Pierre. Recognizing chords with EDS: Part one. In: **International Symposium on Computer Music Modeling and Retrieval**. Springer Berlin Heidelberg, 2005. p. 185-195.

[32] KINNEAR, Kenneth E. **Advances in genetic programming**. MIT press, 1994.

[33] FISHER, Ronald A. The use of multiple measurements in taxonomic problems. Annals of eugenics, v. 7, n. 2, p. 179-188, 1936.

[34] EIBEN, Agoston E.; SMITH, James E.Introduction to evolutionary computing. Heidelberg: springer, 2003.

[35] MCKAY, Cory. **Automatic music classification with jMIR**. 2010. Tese de Doutorado. McGill University.

[36] MCENNIS, Daniel; MCKAY, Cory; FUJINAGA, Ichiro; DEPALLE, Philippe. jAudio: A feature extraction library. In: **Proceedings of the International Conference on Music Information Retrieval**. 2005. p. 600-3.

[37] HOLMES, Geoffrey; DONKIN, Andrew; WITTEN, Ian H. Weka: A machine learning workbench. In: **Intelligent Information Systems, 1994. Proceedings of the 1994 Second Australian and New Zealand Conference on**. IEEE, 1994. p. 357-361.

[38] DURILLO, Juan J.; NEBRO, Antonio J. jMetal: A Java framework for multi-objective optimization. **Advances in Engineering Software**, v. 42, n. 10, p. 760-771, 2011.

[39] KOHAVI, Ron et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: **Ijcai**. 1995. p. 1137-1145.

[40] R Core Team (2015). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

[41] VALENZISE, Giuseppe et al. Scream and gunshot detection and localization for audio-surveillance systems. In: **Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on**. IEEE, 2007. p. 21-26.

[42] GEROSA, Luigi et al. Scream and gunshot detection in noisy environments. In: **Signal Processing Conference, 2007 15th European**. IEEE, 2007. p. 1216-1220.

[43] GAMERMAN, Dani; DOS SANTOS MIGON, Helio.**Inferência estatística: uma abordagem integrada**. Instituto de Matemática, Universidade Federal do Rio de Janeiro, 1993.