Plenary Lecture:

Softcomputing Methodologies Applied to Audio-Based Information Retrieval

Mario Malcangi

Università degli Studi di Milano
DICo - Dipartimento di Informatica e Comunicazione
Via Comelico 39 – 20135 Milano - Italy
Laboratorio DSP&RTS (Digital Signal Processing & Real-Time Systems)
Laboratorio LIM (Laboratorio di Informatica Musicale)

malcangi@dico.unimi.it
Premises

• **Intuitive** and **efficient** access to multimedia information is **becoming a strategic option**, given the increasing availability of such information in large archives and on the web.

• **Audio information** is a **powerful medium** to communicate naturally with such systems, while accessing multimedia information semantically.
• Integrating multiple signal-processing algorithms and soft computing is a new approach toward the development of an audio-based front end for multimedia retrieval.

• Algorithm-based features extraction, artificial neural networks used as pattern matchers, and fuzzy-logic used like classifiers, lead to the development of a content-based, audio-access system capable to retrieve information in a multimedia domain.
Premises (cont.)

- **Audio is an information medium** capable of embedding much more information than we tend to imagine.

- A generic **audio source** (e.g. phone ring, bell sound, people talking, etc.) **embeds information** that is typically overlooked but can easily be used as a key for media retrieval.

- For example, searching a film for a crowd segment is simpler and more effective if we **search for audio** rather than scene or title.
• Current search-engine implementations are very smart at retrieving text-based information (e.g. web pages, documents, files in which text information is available)

• Current search-engines are wanting in their ability to locate multimedia information, especially audio and audio-related information.
• **Two main requirements arise** in building a multimedia search engine:
  
  – **client-side ability to extract features from audio** and to transform text into audio features
  
  – **server-side ability to match and find those features** amidst vast, distributed multimedia information.
Premises (cont.)

- **Human audio perception is fuzzy**, as are audio features: an exact match between query and target is often impossible.

- **Artificial Neural-network feature classifiers** have proven optimal in automatically indexing digital audio collections.

- **Fuzzy logic** has also been applied to audio classification tasks. Such classification is complementary to Artificial Neural-network based pattern matching.
System framework

• The system consists of:
  – an audio-feature extractor (AFE)
  – an artificial neural network-based classifier (ANN)
  – a fuzzy-logic inference engine (FLE)
System framework (cont.)

- Audio features are fed to the ANN-based classifier that identifies the class the audio belongs to.
System framework (cont.)

- The fuzzy-logic inference engine generates a smart query to access an audio repository in search mode.
System framework (cont.)

• The audio-feature extractor consists of a set of digital signal-processing algorithms applied to raw audio data (low-level (physical), time-domain features and frequency domain features).
System framework (cont.)

- The **ANN-based classifier maps** the multidimensional space of audio features onto two-dimensional space to cluster information about features.
- **Clustered audio features** represent the data for the fuzzy-logic inference engine to classify.
System framework (cont.)

- The **fuzzy-logic inference engine classifies** clustered data at the ANN output layer by applying a set of fuzzy rules and membership functions.
System framework (cont.)

- A **similarity query** is then generated fuzzily.
Audio-feature Extraction

- **Time-domain audio features** are calculated according the following general formula:

\[
Q(n) = \sum_{m=0}^{N-1} T[s(m)]w(n - m)
\]

- \(s(n)\) is the audio signal
- \(Q(n)\) is a short-time sampled calculation of a feature
- \(T\) is the transformation function applied to signal \(s(n)\)
- \(w(n)\) is the windowing function for short-time feature calculation
- window size is 20 ms (N samples for a given sampling rate).
Audio-feature Extraction (cont.)

- Root mean square (RMS)

\[
RMS(n) = \sqrt{\frac{1}{N} \sum_{m=0}^{N-1} s^2(m)}
\]
Audio-feature Extraction (cont.)

- Zero-crossing rate (ZCR)

\[
ZCR(n) = \sum_{m=0}^{N-1} 0.5|\text{sign}(s(m)) - \text{sign}(s(m-1))|w(n-m)
\]
Audio-feature Extraction (cont.)

- An additional computed feature is the silent frame rate (SFR)

  \[ \text{SFR} = \frac{\text{silent frames}}{\text{total frames}} \]

- Fuzzy-logic calculation of the silent frame rate (SFR) audio feature.
Audio-feature Extraction (cont.)

- **Frequency-domain audio features** are calculated according to short-time Fourier analysis formula:

\[
S_n(e^{j\omega}) = \sum_{m=0}^{N-1} s(m)e^{j\omega m}w(n-m)
\]

\(S_n(e^{j\omega})\) is a short-time computation of audio-signal energy \(s(m)\) in a limited bandwidth related to the chosen frequency.
Audio-feature Extraction (cont.)

- The calculated frequency-domain audio features are frequency centroid (FC) and cumulative band energy (CBE).

- FC is the balanced point of the spectrum, calculated as follows:

\[
\omega_c = \frac{\int_0^{\omega_0} \omega |S(\omega)|^2 d\omega}{\int_0^{\omega_0} |S(\omega)|^2 d\omega}
\]

- CBE is attained with the above formula repeating the calculation of energy at various frequency to cover four sub-bands:

\[
B_1 = [0, \frac{\omega_0}{8}], B_2 = [\frac{\omega_0}{8}, \frac{\omega_0}{4}], B_3 = [\frac{\omega_0}{4}, \frac{\omega_0}{2}], B_4 = [\frac{\omega_0}{2}, \omega_0]
\]
Sound-feature mapping

- The Kohonen feature-map (KFM) artificial neural network (ANN) was used to map multi-dimensional space onto two-dimensional space.
Sound-feature mapping

- A KFM can map n-dimensional input-vector space onto a neuron layer where neurons are organized according to similarities in input values.
- This ability has been successfully used to map speech sounds onto phonetic space for a high-performance implementation of speech recognition (phonetics-driven speech-to-text).
Sound-feature mapping (cont.)

• Euclidean distance was used to determine the winning node in the map:

\[ D_i = |X - W_i| = \sqrt{(x_1 - w_{i1})^2 + (x_2 - w_{i2})^2 + \ldots + (x_M - w_{iM})^2} \]

• When a node wins more than 1/N times (N is the number of Kohonen nodes), its distance is adjusted upward to attenuate its chance to win.
Sound-feature mapping (cont.)

- For nodes that win less than $1/N$ times, the distance is adjusted downward to make them more likely to win.
- The distance adjusting factor is: $Bi = g(1/N-Fi)$
- The adjusted distance $D^i$ is computed as: $D^i = Di - Bi$
Fuzzy-logic KFM categorization

• To categorize the KFM’s audio-feature mapping ability, an **upper layer is added to the Kohonen layer**. The upper layer consists of a fuzzy-logic engine (FLE) tuned to categorize sounds into types.
Fuzzy-logic KFM categorization

• Several important issues need to be resolved to set up the fuzzy rules and the membership function so that audio information can be classified in a hierarchical fashion and used for fast and effective search in the multimedia database:

![Fuzzy-logic KFM categorization diagram]
Fuzzy-logic KFM categorization (cont.)

- Crisp information from the KFM layer and from certain measured audio features for the given sound class to be categorized is fuzzified.

- Each membership function is derived by looking at the statistics for each feature and how it is clustered by the KFM.

- A membership function is then derived from the shape of the feature’s distribution, simply by superimposing membership shape on the distribution shape.
Fuzzy-logic KFM categorization (cont.)

• The rule model is: **IF (Condition 1) AND (Condition 2) THEN (Category)**

• *Condition 1* and *Condition 2* are fuzzy evaluations of one feature in the audio-measurement domain and one in the KFM-mapping domain.

• *Condition* uses a fuzzy measurement derived from the membership function in terms of qualitative grade scale (e.g., very low, low, medium, high, very high) to represent a fuzzy measurement of the feature (e.g. *RMS is medium, ZCR is low*, etc.).

• For each audio category a set of AND rules are generated.

• A singleton function is used to defuzzify each audio object, thus determining its degree of belonging to an audio category.
Fuzzy-logic KFM categorization (cont.)

• The fuzzy-logic engine needs to be tuned for best performance. Two options are available for the purpose: manual tuning or automatic tuning.

• Manual tuning relies on an audio expert, who chooses among different membership functions. The audio expert may also create rules for best categorizing audio, based on her or his knowledge. A graphic user interface (GUI) is helpful for this task.

• Automatic tuning uses only a triangular membership function to fit the audio-feature distribution shape and fixed format rules. Automatic tuning can also be assisted by a genetic-like process, so that a large number of rules are generated at tuning-time, but only those used most often are kept at run-time.
Thank you for your attention
(any question?)

Mario Malcangi

Università degli Studi di Milano
DICo - Dipartimento di Informatica e Comunicazione
Via Comelico 39 – 20135 Milano - Italy
Laboratorio DSP&RTS (Digital Signal Processing & Real-Time Systems)
Laboratorio LIM (Laboratorio di Informatica Musicale)

Please, address any further question to:

malcangi@dico.unimi.it