Multimedia Genre Characterisation with Fuzzy Embedding Classifiers

Alberto Messina*
Università degli Studi di Torino
Corso Svizzera, 185 I-10149
Turin, Italy
messina@di.unito.it

Maurizio Montagnuolo†
Università degli Studi di Torino
Corso Svizzera, 185 I-10149
Turin, Italy
montagnuolo@di.unito.it

ABSTRACT

Multimedia classification is a key issue in modern data management, where the number of available items is dramatically growing and there is an increasing demand for access to distributed multimedia data. Selection by genre is a simple and effective mechanism for most of the users interested in these applications. In this paper, we present a feature extraction architecture and a novel learning algorithm for multimedia genre characterisation. We show how genre classification can be regarded as a sub-case of this general task, for which we give a complete solution. Our extracted features were designed to offer a reduced semantic gap, trying to take into account structural and cognitive content descriptors, rather than low-level features. Our learning algorithm is based on fuzzy set theory, and makes use of fuzzy C-means (FCM) algorithm as the kernel to learn concepts configurations from data. We tested our learning framework on a test database of over 100 hours of TV broadcast programmes belonging to 7 different common genres. Experimental evaluations showed the effectiveness of our approach. Additionally, we compared our technology with neural networks applied on the same task, in terms of training accuracy. We also compared the generalisation performances of our technique with neural networks and support vector machines.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval models; I.2.6 [**Artificial Intelligence**]: Learning—*Concept learning,Knowledge acquisition*; I.5.3 [**Pattern Recognition**]: Clustering

General Terms

Multimedia Classification, Fuzzy Set Theory, Clustering

Keywords

Genre classification, Fuzzy c-means, Concept mining, Content Analysis, Concept characterisation

1. INTRODUCTION

The increasing demand for wide-access multimedia applications makes access metadata a crucial aspect, alone capable of determining the success or the decline of a system. Genre annotation is the simplest way to give users the ability to select data in large collections, although it is not a straightforward task to be performed automatically. A classical discipline is video genre recognition, which aims at classifying objects into classes according to their stylistic properties, such as newscasts, cartoons, or commercials, by exploiting the idea that objects belonging to the same class share stylistic aspects reflecting the author's intentions in producing those objects [4]. However, often objects can hardly be assigned to a single genre, and at the same time a certain genre is semantically overloaded and imprecise. For these reasons genre classification should be regarded as a particular case of the more general multimedia genre characterisation task, i.e. the ability of associating sets of relevant genres to multimedia items, all together with their degree of relevance for the item. Moreover, it is requested that automatic classifiers should be able to find out relations among genres dynamically. Our approach innovates in this direction, providing a complete framework usable for genre characterisation tasks, based on mid-level content features extraction, and using fuzziness to model the intrinsic uncertainty underlying human judgements about genres. Our extracted features offer a reduced semantic gap and are oriented to capture structural and cognitive properties of the content rather than low-level features. These abstraction properties give our feature set an higher generalisation capability w.r.t. traditional low-level features. We applied our research in the TV area, which is a reference domain for these problems, and a challenging test arena for any new automatic classification technology. Several pattern recognition tools and artificial intelligence techniques were used to address the problem so far. In [14], a C4.5 Decision Tree classifier trained on a 10-dimensional feature vector is employed. In [1] authors use Hidden Markov Models (HMMs), while in [16] automated video genre classification using MPEG-7 descriptors is investigated. In [13], camera motion is used to

*Eng. Messina is a PhD student at Computer Science Department of Turin University, also working as R&D coordinator for RAI CRIT (www.crit.rai.it), where the research has been conducted.

†Eng. Montagnuolo is a PhD student at Computer Science Department of Turin University, sponsored by EuriX s.r.l. (www.eurixgroup.com)

Ambi-sys'08 February 11-14, 2008, Quebec, Canada.

discern among sports video genres. In [9], audio analysis is used to distinguish among talk-shows and news. Hierarchical SVMs are used in [17] to distinguish among 3 sets of genres, including generic video, sports and movies. In [15], the authors use a hybrid ensemble of elementary classifiers (HMMs and SVMs) to discern among 6 genres and using 20 seconds-long clips and a total training set of 3 hours. Finally, fuzzy classifications of video sequences was proposed in [2, 3, 8, 18].

1.1 Innovative contributions of this work

Despite the number of existing efforts, the problem of comprehensively discerning multimedia genres has not been satisfactorily solved yet. This depends on two concomitant factors: (i) the difficulty to analytically define the *concept of genre*, which is a typical subjective, time- and data- dependent concept; and (ii) the intrinsic multiformity of multimedia objects, which can simultaneously belong to several genres. Consequently, most of the existing approaches either attempted to discern only few genres from a simple taxonomy [1, 4, 14, 16], or to distinguish a single well-defined genre from all the others [5], or only focussed on one very specific domain [13]. Additionally, most of them use crisp classifiers, assigning an absolute class label to each multimedia object. However, in real world scenarios, a single object can belong to none, one or more genres at the same time, and boundaries between genres are not necessarily sharp. How to classify e.g. a compound programme containing interviews, highlights of football matches and commercial insertions? Or how to analytically characterise the "entertainment" genre, to which several kinds of different programme formats are typically associated? To overcome these limitations we need to go up one step of abstraction, i.e. to redefine genre classification as an instance of the more general genre characterisation task. We use fuzzy classifiers, which capture additional information about the certainty degree of the classification decision, to model both genre fuzziness and multimedia objects multiformity w.r.t. genres.

In literature it is common the attempt at classifying *clips* of content instead of semantically coherent units. Though this approach has the evident advantage of making the systems able to detect genres analysing only few minutes of content, we believe that it may produce less useful results when the techniques are applied in contexts where objects are typically classified and accessed as wholes (e.g. digital libraries). Besides, complete programmes are often very complex and shirk from being fully characterised only by a short clip. By classifying a clip, we would be able to declare the *local* genre of a programme, without being able to say anything about the *global* content genre, which is the most important aspect in digital libraries search and retrieval systems. Furthermore, clip selection may be prone to a bias effect introduced by whom is selecting the clip boundaries. Complete programmes are editorially controlled by producers and publishers, therefore our potentially biasing mediation is not present. In our view, an object to be characterised must be *semantically consistent*, i.e. either it represents a closed action in time or a beginning and an end can be easily recognisable by a user in its spatio-temporal flow. In the television domain, the semantic consistency can be restricted to be bound to a very simple concept: *finished TV programmes* are semantically consistent multimedia objects. We will henceforth call finished TV programmes *Programme Units* (PUs). Another related issue is that a multimedia object may be composed of *editorial parts*, i.e. modal-temporal segments of the object representing a semantic consistent part *from the perspective of its author* [12]. Editorial parts of an object are actually semantically consistent units. Therefore, they can be considered characterisable units as well. Taking into consideration the relationship between wholes and parts, we can exploit contextualisation information to better characterise parts. This process is impossible if starting from random clip selection. In our framework, we introduced the potentiality to solve these aspects, by wiring in it a native capability of managing such structured objects, expressed in our characterisation method (see Section 3.1.4).

A final drawback of existing works is the relatively modest dimension of the test set (usually from few minutes to some hours, with the exception of [17], where the authors use a database of 60 hours). Furthermore, databases are typically random collections of clips without editorial cohesion. The risk is to produce highly data-coupled classification models and poor generalisation capabilities in real scenarios. We addressed this problem by preparing a test set counting $\approx$ 110 hours of programmes taken in a controlled period of time and from an affine set of delivery channels.

Summing up, we present in this paper many innovative components in the task of multimedia genre characterisation: (i) A novel learning algorithm, called *fuzzy embedding classifier* (FEC) based on FCM [6] and on the novel concept of class projection functions; (ii) A promising set of extracted features w.r.t. the genre classification task; (iii) A modular feature extraction architecture;

2. THE CONCEPT OF FUZZY EMBEDDING

Our learning algorithm is a general purpose machine learning method, and as such it could be used for any automatic data conceptual characterisation task. It is based on the idea that fuzzy clustering is able to provide a rich characterisation of how data items adhere to available concepts through the association of a degree of membership to each of them. We believe that this has the advantage of making the classification operation relying on more than one data configuration evidence at the same time, thus shifting from classification to *characterisation*. Clusters are seen as aggregations of data around a central latent *fuzzy concept prototype*, constituted by the contributions provided by individual concepts through the annotated items, and which is represented by the fuzzy cluster prototype (e.g. the centroid). Figure 1 shows a simple example with two clusters of centroids c_1 and c_2 in a 2-dimensional feature space (f_1, f_2) and two concepts, ω_1 and ω_2. $P_\omega(m)$ is the precision at membership degree m. The item x is associated both to ω_1 and ω_2 with a degree given by the *characterisation function* f.

We call the data transformation procedure from the native attribute space to the fuzzy space *fuzzy embedding*. Consequently, our characterisation method is called *fuzzy embedding classifier* (FEC). To achieve conceptual characterisation, instead of forcing the data configuration to resemble to a sharply separable problem, by e.g. applying a kernel trick, we operate on the arrival set, i.e. the concept labels associated to each item, giving a fuzzy combination of learnt concepts for each item. In our framework sharp classification is still possible selecting the most relevant concept, but it can be seen as a special case in a more general context. We

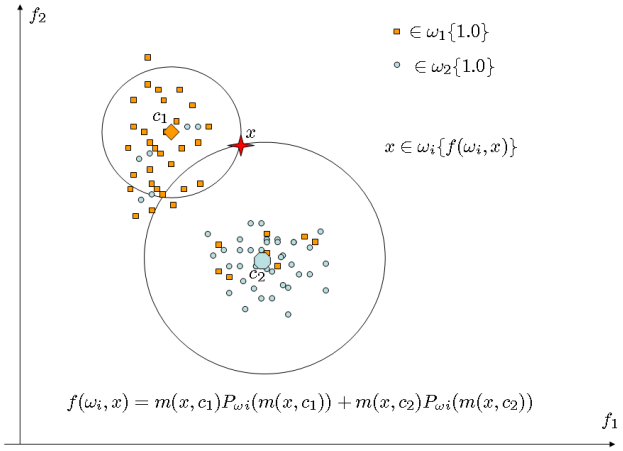

Figure 1: Illustration of the FEC method.

believe that this approach has several advantages: a) it allows to deal with fuzzy annotated data; b) it is able to detect fuzzy concepts in data collections; c) it allows for innovative elaborations as multimodal characterisation and fuzzy concept mining. In addition, notice that classical data mining techniques, e.g. PCA, are still applicable in our framework.

2.1 Feature space definition

2.1.1 Multimodal surrogates

In order to build an automatic learning algorithm we need a numerical surrogate for each PU, and a sound foundation for the feature extraction architecture suitable for the application of our method. Being based on FCM, we need to ensure that the numerical vectors extracted are in a closed set w.r.t. the operations made during FCM clustering, namely linear summation (centroids update step), and distance calculation (membership update step). To achieve this, we need that the feature space have some specific properties, defined in the following section.

2.1.2 FEC-valid feature space

Let S_C be the set of native properties, or attributes, extractable from the PU. We call S_C feature space, and attribute space each individual property in S_C.

Definition 1. Attribute space dimensionality. We define a dimensionality mapping $\mathsf{D} : S_C \rightarrow N$ associating a dimensionality $D^s \in N$ to each $s \in S_C$

Definition 2. Attribute value spaces. We define a mapping function $\mathsf{M}^S : S_C \rightarrow \{\mathcal{P}(R^n), n \in N\}$, associating to each attribute space $s \in S_C$ its value space $X^s \in \mathcal{P}(R^{\mathsf{D}(s)})$. Vectors in X^s are said s-vectors and represent all the possible real values for attribute s.

For the application of FCM, the following operations need to be defined in S_C:

1. *Vector addition*, defined by the function $\oplus : X^s \times X^s \rightarrow X^s$, which associates the vector $x \oplus y \in X^s$ to the couple of s-vectors (x, y);
2. *Vector multiplication*, defined by the function $\odot : R \times X^s \rightarrow X^s$, which associates the vector $a \odot x \in X^s$ to the vector x;
3. *Linear summability*: an attribute space s is linearly summable if $\forall a_i \geq 0 \in R$, $\sum_{i=0}^{N} a_i = 1$ $\forall x_i \in X^s$, $i = 1...N$, $\sum_{i=1}^{N} a_i \odot x_i \in X^s$;
4. *Vector distance metric*, defined by the function $\xi : X^s \times X^s \rightarrow R$, which associates to the couple of s-vectors (x, y) a value $d \in R$.

If s_1, s_2 are two summable and scalable attribute spaces, then (s_1, s_2) is summable and scalable. Furthermore, a vector distance metric is definable in (s_1, s_2) by combining the distance functions of s_1 and s_2, e.g. by making an averaging operation. Another simple property is that if s is summable and scalable, then s is linearly summable.

We have now reached a first result: an attribute space s is *FEC-valid* if it is linearly summable and it is possible to define a vector distance metric between the couple of vectors $(s_1, s_2) \in s$.

In our framework, we represent each PU through its numerical surrogate:

Definition 3. PU representation. The PU, p, is represented by its *feature vector* $\overline{P} = (\mathbf{x}^{s_1 T}, \mathbf{x}^{s_2 T}, ..., \mathbf{x}^{s_{|S_C|} T})^T$, $s_i \in S_C$, $\mathbf{x}^{s_i} \in X^s$ which contains the ordered set of s-vectors natively extracted from the numerical processing of the audiovisual material associated with p.

3. LEARNING ALGORITHM

3.1 Learning method

Our learning method aims at producing models for multidimensional data characterisation. Its key feature is the use of fuzzy clustering, i.e. FCM clustering algorithm, as a processing kernel capable of adaptively learning multidimensional characterisation patterns of objects. The learning algorithm can be trained using fuzzy annotated learning sets, and produces characterisation models able to associate to each test element a fuzzy vector expressing its membership degree to the learnt concepts. The method allows for a flexible management of feature extraction, resolving feature dimensionality mismatch and cross-feature normalisation issues. In fact, each feature is treated separately and coherently with its value space and metrics, and combinations are made in the fuzzy signature spaces. FCM operations are performed using the distance metrics and linear combination functions defined for each attribute space, thus allowing a good level of scalability.

3.1.1 Description of the method

Let Ω be a set of concepts, L the learning set, and $\Sigma \subseteq S_C$ the set of considered attribute spaces. The learning algorithm works in 4 steps:

1. *Run FCM* on the learning set L, using fuzziness parameter f, and restricting PUs' surrogates to attribute space s. This means using X^s as the feature value space for clustering. Let C_s be the set of fuzzy clusters found by FCM on L using X^s as feature value space, and N_{C_s} its cardinality.

2. *Construction of a fuzzy signature* for each PU $p \in L$. The signature is a vector $\Phi_p^s = (\phi_{p,1}^s, \ldots, \phi_{p,N_{C_s}}^s)$, whose element $\phi_{p,i}^s$ is the membership degree of the PU p to the cluster i found by applying step 1. The construction of the fuzzy signature is a space transformation $\tau : X^s \rightarrow R^{N_{C_s}}$ from the value space of s to a new multidimensional space. This transformation is called *fuzzy embedding operation.*

3. Construction and memorisation of a set of *class projection functions* $F^s = \{F_{ij}^s(\phi), i \in C_s, j \in \Omega\}$ on the basis of PUs' signatures. These functions are the core elements for the characterisation: they tell how well the cluster j explains the concept i.

4. *Memorisation of the fuzzy cluster centroids vector* $\Gamma^s = (\gamma_1^s, ..., \gamma_{N_{C_s}}^s)$, $\gamma_i^s \in X^s$.

A learning model based on the attribute space s is then a tuple $M_s = \langle \Gamma^s, F^s, f^s \rangle$ produced by a single instantiation of the above steps on the learning set where each PU's surrogate is restricted to its attribute space s.

Definition 4. Data fuzzy embedding $M_\Sigma = \{M_s, s \in \Sigma\}$ is the data fuzzy embedding of L w.r.t. Σ, denoted also $\mathbf{E}_L^\Sigma$.

3.1.2 Data-driven parameter configuration for FCM

FCM requires resolving some configuration issues. First, the number C of initial clusters. We select it by doing a statistical analysis on L. Let $s \in \Sigma$ be the attribute space to be elaborated, we calculate the first and second order statistics of the s-vectors in each element of the fuzzy partition of L induced by the concept ω_i, as follows:

$$\mu_i^s = \frac{\sum_{k=0}^{|L|} \mathbf{x}_k^s M_{k,i}}{\sum_{k=0}^{|L|} M_{k,i}} \qquad \sigma_i^s = \sqrt{\frac{\sum_{k=0}^{|L|} \xi^s(\mathbf{x}_k^s, \mu_i^s)^2 M_{k,i}}{\sum_{k=0}^{|L|} M_{k,i}}} \quad (1)$$

where $i = 1...N_\omega$ $\mathbf{x}_k^s \in X^s$ is the s-vector representing the value of attribute s for the PU k. k is associated to concept ω_i with degree $M_{k,i}$, and $\xi^s()$ is the distance defined in X^s. C is initialised using the following saturating function:

$$C_i = \rho(\sigma_i^s, \mu_i^s) = \lceil [1 + (C_{MAX} - 1)(1 - e^{-\frac{(\sigma_i^s)^2}{\|\mu_i^s\|^2 + \beta}})] \rceil \quad (2)$$

$$C = \sum_{k=1}^{N_\omega} C_k, \quad (3)$$

where the maximum number of generated clusters is limited by C_{MAX}. Equations 1 and 2 have the effect of weighting the probability distributions $p(\mathbf{x^s}|\omega_\mathbf{i})$ in L so that more sparse distributions give more contribution to the sum of C. Once that each C_i is calculated, a corresponding number of centroids to initialise FCM is generated randomly in the region

$$X^s \cap \left\{ a \frac{\xi^s(\mathbf{x}_k^s, \mu_i)^2}{(\sigma_i^s)^2} \leq 1 \right\} \quad (4)$$

where D^s is the dimensionality of s.

Another issue related to the use of FCM is cluster validation. Our goal is optimising the FEC characterisation performance. Therefore, reduction of model complexity (i.e. the number of clusters) is the most important requirement for us. We chose three main methods to prune clusters resulting from a FCM process: a) *Pruning on coo-position.* Two cluster centroids are merged if their mutual distance is lower than a defined threshold ϵ_d; b) *Pruning on absolute membership.* A cluster is rejected if the average membership over the data set is lower than a defined threshold ϵ_m. c) *Pruning on membership variance.* A cluster is rejected if the membership variance over the data set is lower than a defined threshold ϵ_v

3.1.3 Learning projection functions

Projection functions F^s are the core of our system. They are a compact way to store what the method learns from the data configuration and have a straightforward use in the characterisation operation. We define $F_{ij}^s(\phi)$ as the fuzzy classification precision for concept j given by cluster i at membership degree ϕ. This is calculated by counting the fuzzy fraction of elements in the learning set belonging to concept j falling in a region of membership to i greater than ϕ. More formally:

$$F_{ij}^s(\phi) = \frac{\sum_{p \in \mathfrak{F}_i(\phi)} M_{p,j}}{|\mathfrak{F}_i(\phi)|} \quad (5)$$

where $\mathfrak{F}_i(\phi)$ is the set of PUs in L whose membership degree to cluster i is greater than ϕ, $M_{p,j}$ is the degree of membership of PU p to the concept j. In the construction of these functions we follow a data-driven adaptive mechanism: it is made by sampling the membership interval $[0, 1]$ in a number of sub-intervals that linearly depends on N_{C_s}. This is justified by the fact that as the number of found clusters increases, the average distance among clusters centroids decreases, therefore a greater discriminant in the range of variability of ϕ is needed. Once the calculation at all sampled points is made, we store the found functions in tabular form. As shown in Equation 5, our system can be natively trained with fuzzy data, i.e. with learning sets where PUs have a degree of membership to each available concept instead of having a crisp association to only one of them. Crisply classified data are treated by our system as a particular case.

3.1.4 Objects Characterisation

In our framework, conceptual characterisation is performed by exploiting the information stored for each model M_s. Let T denote the test set. For a test element $t \in T$, its s-vector is calculated. Then, a fuzzy embedding $e_t^s = (e_1^s, e_2^s, ..., e_{N_{C_s}}^s)$ is built by calculating the membership of t to the clusters contained in C_s, which are stored in Γ^s. This is done by applying the membership update step of FCM using the distance defined for s and the fuzzifier f^s. Finally the element membership ϑ_i^s to concept i is calculated as follows:

$$\vartheta_i^s = \sum_{k=0}^{N_{C_s}} \int_0^1 F_{ki}^s(\phi) N_i(\phi) \delta(\phi - e_k^s) d\phi \quad (6)$$

$$\Theta^s = (\vartheta_1^s, \vartheta_2^s, ..., \vartheta_{N_\omega}^s)^T \quad (7)$$

where $\delta(x)$ is the Dirac distribution and $N(\phi)$ is a normalisation function. In our experiments we used $N(\phi) = \phi$ to enhance higher membership values and depress lower ones. Therefore, each test element is associated to the *fuzzy characterisation vector* $\Theta^s = (\vartheta_1^s, ..., \vartheta_{N_\omega}^s)$ expressing the membership degree of the element to the concepts $(\omega_1, \omega_2, ..., \omega_{N_\omega})$. We can give F_{ki}^s the physical interpretation of a function accounting for the *fuzzy mass* distribution of concept i around cluster k, and $N_i(\phi)$ that of accounting for the law of variation with ϕ of fuzzy mass attraction towards the centroid of

k. A natural extension of our approach can be exploited in those cases where the membership of an element can be expressed by a *function of* ϕ rather than by a single value. An example of this is characterisation of multimedia collections, where each collection item contributes with its membership embedding to the total membership function of the collection. Here, the $\delta(\phi)$ would be substituted with a generic function of the parameter ϕ built up by a functional composition of the elementary membership embedding.

The fuzzy characterisation vector Θ^s can be used to perform a crisp classification by selecting the concept ω resulting with the maximum membership in the corresponding element of the vector:

$$\omega = \arg \max_{i=1...N_\omega} \vartheta_i^s \quad (8)$$

3.2 Building the characterisation model

We adopt a two-step generation/selection strategy to build the characterisation model, followed by the generation of an ensemble classifier to enhance the overall accuracy.

3.2.1 Model generation and selection

For each attribute space $s \in \Sigma$ we generate an explorative set of models $\mathcal{M}^s = \{\check{M}_s\}$. Each model in $\mathcal{M}^s$ is generated by running the model generator described in the previous sections on the learning set L. This is done by running FCM with different values of fuzzy parameter f and C_{MAX}. Fuzzifier f is varied from f_{min} to f_{max} with σ_f step; C_{MAX} is varied from C^m_{MAX} to C^M_{MAX} with a step of σ_c. In addition, a single step of cluster pruning based on coo-position is made after each clustering. After the pruning step, FCM is re-run over the data. The total number of explorative models generated is thus $|\mathcal{M}^s| = \frac{f_{max}-f_{min}}{\sigma_f} \frac{C^M_{MAX}-C^m_{MAX}}{\sigma_c}$. Each model in $\mathcal{M}^s$ is therefore indexed by the couple (f, C_{MAX}) used to generate it. The total set of runs is made making the external loop on f and the internal loop on C_{MAX}. The threshold ϵ_d for the pruning step is varied adaptively along the generation process. Starting from a fixed value $\epsilon^0_d = \alpha\overline{\|\mathbf{x^s}\|}$, corresponding to each external loop on f with $C_{MAX} = C^m_{MAX}$, its value is lowered by a fixed percentage r if the ratio between the number of found clusters and the number of data items is lower than a parameter Min_C, indicating over-aggregation, and raised by the same percentage if the same ratio is higher than another parameter Max_C. This way, the effect of the pruning is kept in a controllable range. After testing each $\check{M}_s \in \mathcal{M}^s$ against L and T, we associate to it a performance vector $\Pi = (\pi^{tr}, \pi^{ts}, c^{tr})$, where π^{tr} is the training classification accuracy, calculated as the ratio between the number of correctly characterised elements in L and the cardinality of L, $|L|$; π^{ts} is the test classification accuracy, calculated as the ratio between the number of correctly characterised elements in T and the cardinality of T, $|T|$; c^{tr} is the training complexity, calculated as the ratio between the number of clusters generated (after pruning) N_{C_s} and the number of learning items in L. Notice that in the general case, a test element t can be judged correctly characterised if the vector space distance between the calculated fuzzy characterisation vector and the ground-truth vector is less than a predefined threshold. Once that $\mathcal{M}^s$ is generated, we need to select the best performing model among its elements. In general this can be done according to several criteria, also taking into account the classifier complexity. In our experiments, we chose to select the model having the best test performance for each $s \in \Sigma$, i.e. the model $\check{M}^{msp}_s$ associated to the parameter couple $(f, C_{MAX})^{msp}$ for which the maximum test precision π^{ts}_{max} was reached.

Table 1: The television programme database.

	T.S.	Co.	Mu.	Ca.	Fo.	Ne.	W.F.
Hours	44.2	3.5	3.9	18.8	17.6	21.2	2
# PU	60	67	60	59	22	63	65

3.2.2 Ensemble characterisation

After the model selection phase, we have one model for each attribute space $s \in \Sigma$. We then construct an ensemble classifier according to either: (i) *k-best performers selection.* We select the k models corresponding to the k best performers attribute spaces w.r.t. the model selection criterion adopted; or (ii) *Feature semantics-driven selection.* We select models corresponding to affine attribute spaces. Once the selection of k models is made, the ensemble classifier associates to each PU in T the *ensemble characterisation vector* $\Theta^{\Sigma_e}_e = \frac{1}{k}\sum_{s_i \in \Sigma_e} \Theta^{s_i}$, where $\Sigma_e \subseteq \Sigma \subseteq S$ is the set of the k selected attribute spaces.

4. EXPERIMENTAL EVALUATION

The main task of the experimental evaluation consisted in testing the ability of our technology to perform a crisp genre classification task on a real-life set of PUs. This was done to assess the position of our technology w.r.t. state-of-the-art in crisp classification, thus demonstrating the added value brought by its fuzzy infrastructure, which other methods do not provide. All test PUs were previously annotated crisply with their genre label: Commercials, Cartoons, News, Talkshows, Weather Forecasts, Music, and Football.

4.1 Extracted features and task settings

The learning database collects $\approx$ 100 hours of TV programmes from the daily programming of national and regional broadcasters (see Table 1). Programmes were selected to ensure genre representativeness without loosing in generality. We constructed the set by balancing the number of PUs per genre, respecting the genre frequency observed in the average broadcast schedule of a closed period of time. Programmes were selected randomly in a wider period of time, to avoid bias effects due to local reuse of material in the short term broadcast (e.g. newscasts and weather forecasts).

Effective multimedia classification requires a multimodal approach, considering aural, visual, and textual information, which means extracting properties from all available media tracks. We implemented a set of processors able to extract the following set of properties:

- Average Face Number (AFN). The average number of faces in a PU (face-per-frame rate).
- Face Number Distribution (FAD). The distribution of faces in a PU. This distribution is expressed by a normalised (w.r.t. the total number of frames) 11-bin histogram. The i^{th} $(i = 0, ..., 9)$ bin counts the fraction of frames containing i faces. The last bin counts the fraction of frames containing more than 10 faces.
- Face Covering Percentage (FCP). The covering percentage of faces in a PU, calculated as the ratio be-

tween images area containing faces and the total images area of the PU.

- Face Position Distribution. (FPD). The distribution of position of faces in a PU. This distribution is expressed by a normalised (w.r.t. the number of total faces) 9-bin histogram. Each bin i represents the fraction of faces in the i^{th} position in the frame[1].
- Audio Segmentation Analysis (ASA). The percentage of *speech, silence, noise* and *music* within the programme's audio track.
- Background Audio Analysis (BAA). The percentage of *silence, noise* and *music* within the spoken parts in a PU. All values are normalised with respect to the total number of samples labelled as speech content.
- Average Shot Length (ASL). The average video shot length in seconds in a PU.
- Average Speech Rate (ASR). The average number of words per second in a PU.
- Video Shot Clusters Duration (CLD). The normalised duration of video shots taking part in a video shot cluster with cardinality $>= 2$.
- Video Shot Clusters Saturation (CLS). The ratio between the number of visual shots aggregated in shot clusters counting at least two elements and the total number of shots of a PU.
- Shot Length Distribution (SLD). The shot length distribution of a PU, expressed by a 65-bin normalised histogram. Bins ranges are uniform and the 65^{th} bin counts shots whose length is greater than 20 seconds.

Some of these are well-known and used in literature (e.g. ASL, SLD, ASR), some others are new (e.g. FPD, CLD, FCP). The feature space is thus defined by $\Sigma = \{ASA, BAA, ASR, ASL, CLD, CLS, SLD, AFN, FAD, FCP, FPD\}$.

Due to normalisation constraints, all multidimensional attribute spaces are linearly summable. In our experiments, distances are calculated using Euclidean distance between s-vectors. We used K-fold cross validation with $K = 6$, i.e. we partitioned our database D in 6 parts $l_1, l_2, ..., l_6$, $\cup_{i=1}^{6} l_i = D$, $l_i \cap l_j = \emptyset \ \forall i \neq j$ and made 6 distinct learning-testing rounds, each selecting a part l_i as the test set T and the remaining part $L' = \cup_{j \neq i} l_j$ as training set. The item set subdivision was done taking care of respecting the observed genre frequencies in the overall set. We selected the Max Test Precision criterion for model selection and the k-best performers for construction of the ensemble. We used $(1.2, 3.0)$ as the range for fuzziness f, $(10, 100)$ as C_{MAX} range in the model generation phase, and $a = 1$ in Equation 4. We set $\sigma_f = 0.1$, and $sigma_c = 10$, $\alpha = 0.05$, $r = 0.1$, $Min_C = 0.1$ and $Max_C = 0.4$. Finally we used $\beta = 1$ in Equation 2.

4.2 Analysis of the experimental results

Tables 2, 3, and 4 report the experimental results in the main task. Table 2 shows performance of the different surrogate attribute spaces in the six experiments, ordered by average accuracy. Table 3 shows performance of the ensemble classifier with aggregation index in the range 2 to 6. Aggregation index equal to i means that the first i surrogate attribute spaces of Table 2 were taken to make the ensemble. Finally, Table 4 illustrates the classification performance of the ensemble classifier for the different genres, at aggregation index equal to 4, at which the best average performance was obtained. In some cases the average accuracy per class is close to 100%. The behaviour of the accuracy observed with the increase of the aggregation index shows that selected features are very effective in the classification task, because of their intrinsic capability of capturing independent aspects contributing to genre characterisation.

Despite that comparisons with other works is in general an hard task, due to the different data sets used and to the different set of reference genres, we find it interesting to confront the obtained accuracy with some previous works [10, 11, 14, 16, 17] dealing with video genre classification (see Table 5). Apart from the obtained accuracy, some other important issues distinguish our work from existing efforts: the consistently bigger size of the database ([17] use half the hours we use), its statistical well-formedness w.r.t. a real-life application scenario, the validation scheme and the number of recognised genres.

[1]Positions are *top-left*, *top-right*, *bottom-left*, *bottom-right*, *left*, *right*, *top*, *bottom*, *centre*.

Table 2: Performances of the surrogate attribute spaces in the test run.

att. sp.	[experiment]						[avg]
ASA	0.72	0.74	0.64	0.72	0.76	0.75	**0.72**
SLD	0.69	0.77	0.66	0.69	0.77	0.64	**0.71**
FPD	0.57	0.97	0.63	0.58	0.61	0.69	**0.68**
FAD	0.59	0.68	0.72	0.72	0.64	0.63	**0.66**
AFN	0.56	0.68	0.63	0.63	0.63	0.64	**0.63**
ASR	0.56	0.53	0.57	0.60	0.53	0.72	**0.59**
FCP	0.50	0.62	0.60	0.62	0.58	0.53	**0.57**
BAA	0.54	0.55	0.46	0.56	0.54	0.62	**0.54**
CLD	0.51	0.47	0.51	0.59	0.54	0.52	**0.52**
ASL	0.51	0.52	0.48	0.50	0.49	0.55	**0.51**
CLS	0.43	0.55	0.50	0.52	0.45	0.58	**0.51**

Table 3: Performances of the ensemble classifier.

Aggr. index	2	3	4	5	6
avg. acc.	0.82	0.90	0.90	0.84	0.85

4.3 Fuzzy genres mining

Out system is able to perform fuzzy genre mining. The

Table 4: Performances of the ensemble classifier per class (aggregation index=4).

Exp.	[genre]						
	com.	news	w.f.	cart.	mus.	talk-s.	foot.
1	0.91	0.91	0.91	0.90	1.00	0.60	0.75
2	1.00	1.00	1.00	1.00	1.00	1.00	1.00
3	0.82	0.64	1.00	0.90	0.70	0.90	0.25
4	0.83	0.91	1.00	0.90	1.00	1.00	1.00
5	1.00	1.00	1.00	0.80	0.80	0.90	0.75
6	1.00	0.90	0.91	0.80	0.40	0.89	0.67
avg	**0.93**	**0.89**	**0.97**	**0.88**	**0.82**	**0.88**	**0.74**

Table 5: Comparison with previous works.

Work	Avg. acc.	Work	Avg. acc.
Roach [11]	0.82	Truong [14]	0.83
Xu [16]	0.87	This work	0.90
Yuan [17]	0.87	-	-

Table 6: Fuzzy genres combinations.

Fuzzy genre	Score	Fuzzy genre	Score
{news, talk-s.}	0.44	{comm., cart.}	0.32
{comm., music}	0.31	{cart., music}	0.31
{comm., cart., music}	0.31	-	-

underlying nature of the fuzzy embeddings implies that a certain PU can be characterised in terms of its adherence to a fuzzy combination of basic genres. In fact, each element of the characterisation vector Θ^s represents the membership of a PU to each different genre among those learnt from the analysis of L. We define this result PU *genre characterisation*. Genre classification is then a particular case of genre characterisation, in which the top-score genre is selected. By exploiting this features, we have the possibility to explore the degree of *affinity* that a particular genre share with other genres. To demonstrate this, we considered the value of the ensemble characterisation vectors $\Theta_e^{\Sigma_e}$ on collections of items corresponding to the 7 genres. This was done by calculating a set of *collective* characterisation vectors $\Theta_{e,i}^{\Sigma_e}$, where i ranges in the set of genres, obtained by averaging the $\Theta_e^{\Sigma_e}$ vectors on the subset of the learning set L of the PUs labelled with genre i. For the construction of the elementary ensemble characterisation vectors $\Theta_e^{\Sigma_e}$ we used an aggregation index equal to 4 (i.e. ensembling ASA, SLD, FPD and FAD). We then built a genre affinity matrix A, averaging the collective characterisation vectors on the six distinct experiments of our cross validation schema. The row element A_i is thus the $\Theta_{e,i}^{\Sigma_e}$ vector, normalised w.r.t. its maximum element value, i.e. $A_i = \frac{\Theta_{e,i}^{\Sigma_e}}{\max_k \vartheta_{e,k}^{\Sigma_e}}^T$. The affinity matrix A is showed in Table 7. We observe that the obtained matrix is composed by an empirically symmetrical block (in which $(a_{ij} - a_{ji})^2 < 0.05$), ranging from Commercials to Talk-Shows. We can define fuzzy genres by making all the possible aggregations of elementary genres. The number of possible genre aggregations with cardinality greater than 2 equal to $\sum_{k=2}^{N_\omega} \binom{N_\omega}{k}$. Then, we can associate a *strength* S_k to each fuzzy genre g_k generated, as follows:

$$S_k = \min_{i,j} a_{ij}, \qquad a_{ij} \in A^{g_k} \tag{9}$$

where A^{g_k} is matrix A restricted to columns and rows corresponding to the genres aggregated in g_k. Table 6 shows the strength scores of the mined fuzzy genres having strength score greater than 0.3. Results match with our intuitive expectations: from the aural (ASA), video structural (SLD) and facial presence (FPD, FAD) points of view, the discovered top-score combinations contain concepts that are affine in reality. This result allows us to state that the genres e.g. of News and Talk-Shows are *fuzzy* genres, because their mutual affinity, learnt from the given examples, is sufficiently strong. Following this method, we can characterise entire collections, as our test database, providing the mined fuzzy genres, and this can be done by analysing set of items crisply associated to genres and evaluating the strength of their aggregations.

Table 7: Genre affinity matrix.

	com.	news	w.f.	cart.	mus.	t.-s.	foot.
com.	1.00	0.19	0.20	**0.32**	**0.31**	0.11	0.07
news	0.21	1.00	0.19	0.20	0.21	**0.46**	0.05
w.f.	0.19	0.18	1.00	0.20	0.13	0.23	0.04
cart.	0.**45**	0.21	0.29	1.00	**0.31**	0.10	0.17
mus.	**0.46**	0.27	0.18	**0.34**	1.00	0.19	0.10
t.-s.	0.15	**0.44**	0.21	0.09	0.18	1.00	0.03
foot.	**0.36**	**0.40**	**0.29**	**0.51**	**0.38**	**0.43**	1.00

Table 8: Maximum training precision for fuzzy embedding classifier

att. sp.	[experiment]						[avg]
ASA	0.78	0.87	0.82	0.84	0.86	0.86	**0.84**
SLD	0.92	0.92	0.92	0.89	0.91	0.93	**0.91**
FPD	0.96	0.94	0.96	0.95	0.97	0.95	**0.95**
FAD	0.86	0.85	0.85	0.85	0.85	0.84	**0.85**
AFN	0.84	0.84	0.84	0.85	0.85	0.83	**0.84**

5. COMPARISON VS. OTHER METHODS

We compared the learning capability of our technology with artificial neural networks (ANN). We trained sigmoid-based ANNs for the classification task using the iRPROP technique [7]. The classification of an item was done by selecting the genre with maximum output value from the output layer of the network. Tables 8 and 9 report the maximum training performances obtained. Our fuzzy embedding classifier outperforms the neural network one in AFN and FAD attribute spaces, is consistently better in SLD, while essentially the same performance is reached for ASA and FPD.

We compared the generalisation performance of our technology with that of ANNs and that of radial basis kernel Support Vector Machines. To make our comparisons, we used the same cross-validation architecture and the same set of attribute spaces for training and testing FECs, ANNs and SVMs.

Table 10 reports test accuracy for the various features reached using FEC, ANN and SVM. Our approach gives better results across the entire feature set if compared with ANNs. Comparison with SVMs is more balanced. Some features give better performances with SVMs (e.g. ASA,

Table 9: Maximum training precision for neural network classifier.

att. sp.	[experiment]						[avg]
ASA	0.89	0.75	0.87	0.82	0.76	0.87	**0.83**
SLD	0.82	0.86	0.82	0.77	0.78	0.80	**0.81**
FPD	0.86	0.91	0.94	0.88	0.88	0.96	**0.91**
FAD	0.57	0.66	0.60	0.49	0.56	0.49	**0.56**
AFN	0.53	0.55	0.52	0.52	0.50	0.54	**0.53**

Table 10: Generalisation performance comparison.

	ASA	SLD	FPD	FAD	AFN	ASR
FEC	0.72	0.71	0.68	0.66	0.63	0.59
ANN	0.68	0.54	0.53	0.55	0.57	0.53
SVM	0.75	0.82	0.58	0.67	0.60	0.57
	FCP	BAA	CLD	ASL	CLS	**Avg**
FEC	0.57	0.54	0.52	0.51	0.51	**0.60**
ANN	0.55	0.45	0.47	0.46	0.40	**0.52**
SVM	0.57	0.51	0.51	0.51	0.47	**0.60**

SLD), while the opposite holds for other features (e.g. FPD, AFN). The average test performance (last column of Table 10) make us conclude that in the examined task, FECs are definitely a better option than ANNs and a significant alternative to SVMs.

6. CONCLUSIONS AND FUTURE WORK

In this paper we introduced a new approach at multimedia genre characterisation. We demonstrated the effectiveness of fuzzy embedding classifiers in the genre classification of TV programmes, seen as a particular case of the more general multimedia genre characterisation task, for which our framework provides a complete solution. Reached accuracy in the crisp classification task shows an outstanding score if compared with relevant state of the art in the specific task, as well as with other established automatic classification techniques (ANNs and SVMs). Besides, our technology is natively able to perform other tasks like fuzzy genre mining and can be naturally extended to cover the more general conceptual characterisation task, i.e. associating set of concepts (e.g. LSCOM Lexicon Definition and Annotations) to multimedia objects.

In the future we want to investigate in the following directions: a) *Optimised model generation.* Instead of scanning the fuzziness parameter f and C_{MAX} in a fixed rectangle, optimal paths in the (f, C_{MAX}) search space could be found by e.g. applying genetic algorithms adapted to the fuzzy embedding architecture; b) *Improvement of the fuzzy cluster pruning strategy.* In the present work we perform a single pruning step after each FCM run, based on co-position. Other pruning methods (minimum membership, minimum variance) could be applied iteratively till a stability is reached; c) *Class projection functions pruning.* Filtering off functions having negligible values in the whole ϕ interval, i.e. those not contributing to the computation of Θ^s, would substantially lower the complexity, leaving characterisation accuracy unchanged; d) *Class projection functions approximations.* Class projection functions are well approximated by low-degree polynomials. This would optimise the requested space for model memorisation; e) *Characterisation performance optimisation.* We run our tests using a fixed expression for the normalisation function $N_i(\phi)$ in Equation 6. We will develop an optimisation algorithm for a generic polynomial expression of $N_i(\phi)$, by minimising the quadratic characterisation error in the learning set;

7. REFERENCES

[1] N. Dimitrova, L. Agnihotri, and G. Wei. Video classification based on hmm using text and faces. In *Proc. European Signal Processing Conference*, 2000.

[2] A. Doulamis, Y. Avrithis, N. Doulamis, and S. Kollias. Interactive content-based retrieval in video databases using fuzzy classification and relevance feedback. In *Proc. IEEE ICMSC '99*, 1999.

[3] A. M. Ferman and A. M. Tekalp. A fuzzy framework for unsupervised video content characterization and shot classification. *SPIE Journal of Electronic Imaging*, 10(4):917–929, 2001.

[4] S. Fischer, R. Lienhart, and W. Effelsberg. Automatic recognition of film genres. In *Proc. ACM Multimedia 1995*, 1995.

[5] R. Glasberg, C. Tas, and T. Sikora. Recognizing commercials in real-time using three visual descriptors and a decision-tree. In *Proc. of IEEE ICME 2006*, pages 1481–1484, 2006.

[6] F. Höppner, F. Klawonn, R. Kruse, and T. Runkler. *Fuzzy Clustering Analysis.* Wiley, 1999.

[7] C. Igel and M. Hausken. Improving the rprop learning algorithm. In *Proc. of the 2nd Int. ICSC Symposium on Neural Computation*, 2000.

[8] R. Jadon, S. Chaudhury, and K. Biswas. Generic video classification:an evolutionary learning based fuzzy theoretic approach. In *ICVGIP-2002*, 2002.

[9] R. S. Jasinschi and J. Louie. Automatic tv program genre classification based on audio patterns. In *Proc. of. 27th Euromicro Conference*, pages 370–375, 2001.

[10] W. Kraaj and J. Arlandis. Feature extraction task: Overview. In *TRECVID Workshop 2004*, 2004.

[11] M. Roach. *Video Genre Classification.* PhD thesis, University of Wales Swansea, 2002.

[12] C. G. Snoek and M. Worring. Multimodal video indexing: A review of the state-of-the-art. *Multimedia Tools and Applications*, 25(1):5–35, 2005.

[13] S. Takagi, S. Hattori, K. Yokoyama, A. Kodate, and H. Tominaga. Sports video categorizing method using camera motion parameters. In *VCIP 2003*, 2003.

[14] B. T. Truong and C. Dorai. Automatic genre identification for content-based videocategorization. In *Proc. ICPR'00*, 2000.

[15] S. Vakkalanka, C. K. Mohan, R. Kumaraswamy, and B. Yegnanarayana. Combining multiple evidence for video classification. In *Proc. of ICISIP 2005*, pages 187–192, 2005.

[16] L. Q. Xu and Y. Li. Video classification using spatial-temporal features and pca. In *Proc. of IEEE ICME 2003*, 2003.

[17] X. Yuan, W. Lai, T. Mei, X.-S. Hua, X.-Q. Wu, and S. Li. Automatic video genre categorization using hierarchical svm. In *Proc. of IEEE ICIP 2006*, 2006.

[18] Y. Zhiwen, Z. Xingshe, G. Jianhua, and Y. Zhiyi. Fuzzy clustering for tv program classification. In *Proc. of the ICIT 2004*, 2004.