Adaptive Discovery of Indexing Rules for Video

Marcin Detyniecki

CNRS, LIP6, Université Paris 6
8, rue du Capitaine Scott, 75015 Paris, France
Marcin.Detyniecki@lip6.fr

Abstract. This paper presents results, at an early stage of research work, of the use of fuzzy decision trees in a multimedia framework. We present the discovery of rules in three different indexing scenarios. These rules represent knowledge that can be interpreted as guidelines for the development of better indexing tools. We use a fuzzy decision tree algorithm to extract these rules (just) from color proportions of key-frames extracted from one video-news broadcast. Experimental results and comparisons with other data mining tools are presented.

1 Introduction

On the one hand, the growth of video data has caused a need to analyze and exploit it. Hints of this increase are the availability of video news on the web or the appearance in the market of video recorders equipped with hard drives. Due to the overwhelming quantity, it appears that the users tend to interact e.g. in order to find what he wants. But in order to respond to the user requests indexing is needed. Unfortunately, today's indexing is generally done manually. Added to this the growth of video data and the requirement of new applications for finer grain access, urges an automation of the indexing process.

On the other hand, in the recent years, fuzzy data mining introduces new methodologies to extract and discover fuzzy knowledge from either classical or fuzzy data repositories. It leads to the improvement of the knowledge of the domain from where the data is obtained. The advantage of using *fuzzy* algorithms is that it enables us not only to offer in a more comprehensive way the discovered knowledge, but also to be able to handle uncertain and/or fuzzy data (as well as traditional numerical or symbolic data) 1.

Thus, it appears natural and promising to link fuzzy data mining with multimedia data to obtain robust fuzzy multimedia mining. For instance, in this paper the mined rules are knowledge that can be used to improve the indexing process (i.e. helping the development of better indexing tools).

Besides the problem of the quantity, dealing with multi-media introduces a new difficulty related to the polymorphism of the data 2. In fact the information that can be extracted from, for instance, a video (or a website) are texts, images, sounds, temporal data, metadata, etc. A solution is to use flexible and automated data-mining

A. Nürnberger and M. Detyniecki (Eds.): AMR 2003, LNCS 3094, pp. 176–184, 2004.

tool, which will induce knowledge from all kinds of data. A particular instance of such tools is the fuzzy decision tree based algorithm 1.

In this paper we presents results, at an early stage of research work, of the use of fuzzy decision trees in a multimedia framework. Here we focus on the mining of color proportions of key-frames extracted from one single video-news. This simple approach allows us to clearly understand (interpret) the results and identify potential difficulties. However, the presented approach is designed in such a way that it can be directly applied on more complex data, as for instance cross-media indexes.

In Section II, we shortly provide some references for our data-mining tool: the fuzzy decision trees algorithm and software. In Section III we illustrate the knowledge discovery process. We expound the discovery of rules for three different indexing scenarios. These rules represent knowledge that can be interpreted as guidelines for the development of better indexing tools. Experimental results, comparisons with other data mining tools and limitation of the decision tree approach are also presented in this section. Finally we conclude with a short discussion about the obtained results.

2 Fuzzy Decision Trees

Knowledge Discovery from Data (KDD) was introduced at the beginning of the nineties 4. However, due to the complexity of multimedia data, Multimedia Data Mining (MDM) was recently proposed as a new topic of research 5.

In fact, in a multimedia framework, versatile data-mining tools are necessary. One case of such tools is the fuzzy decision tree learning algorithm, which provides rules that summarize and explain the data. We use the Salammbô software, which is able to handle typical numerical input (non fuzzy), and it constructs without human intervention a fuzzy decision tree 6.

Another advantage of using fuzzy decision trees resides in the fact that they represent in a natural and understandable way the knowledge: a fuzzy decision tree is equivalent to a set of fuzzy "*if...then*" rules.

These trees are built based on an entropy measure, which translates a certain degree of order. In other words we can automatically discover which features are the most important (discriminant) and what are the values to be consider for these features.

In Figure 1 we can see an example of the output of the Salammbô software for mining based on colors for the detection of inlays (see section III.A). The rules are self-explanatory even though the decisions "less than" and "greater than" are fuzzy (see Figure 3).

3 Discovering Indexing Rules

In order to show the potential of using fuzzy trees (and more generally any data mining tool), we restrain our research to a well-known feature: the color.

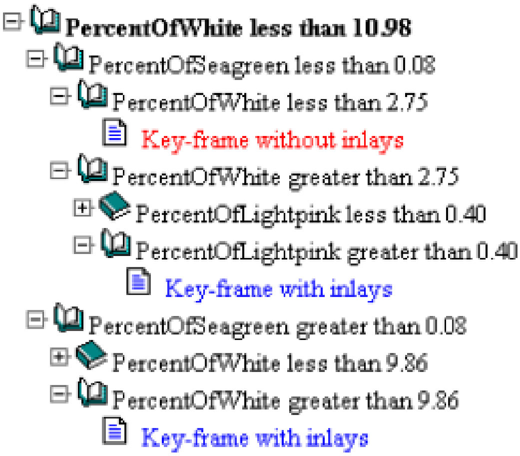

Fig. 1. Example of rule extracted by Salammbô

We start from a set of key-frames extracted (per shot) from a single news broadcast (video) 7. In a second step, the set of colors of each key-frame is vectorized and "projected" to a given reference-palette (for instance a palette of 64 or 256 colors obtained by sampling equally the RGB space). We obtain like this a common basis to compare the key-frames. Then for each key-frame a histogram of frequencies (of the colors) is computed. This provides us with a vector in a reference space (defined by the colors of the reference palette) for each key-frame (see Figure 2).

So, from each key-frame we obtain a vector, which is then considered as a training example. Thus, from a set of classified key-frames (examples), a training set can be composed. Finally we built the fuzzy decision tree using the Salammbô software.

In this paper we consider three different mining problems. All related to the extraction of knowledge associated to the extraction of the general structure of the video news (macro-segmentation).

3.1 Discovering the Presence of Inlays

Inlays that appear on the TV screen are very often hints for the structure of the video news. They also appear either when a new person is presented or when a report ends. They usually consist in a square or a rectangle that frames some text (e.g. name of the journalist, name of the place, etc).

We have conducted several mining experiments in order to determine if colors are relevant (discriminant) for the detection of inlays. We composed a training set with 176 vectorized key-frames of one single video-news, to each vector was assigned one class (type) of the key-frame: either *with* or *without inlays*.

A first experiment was conducted with the whole training set and based on a reference palette of 64 colors. The resulting fuzzy decision tree is not very deep and has a root node on which the presence of the white color is requested (white is the major

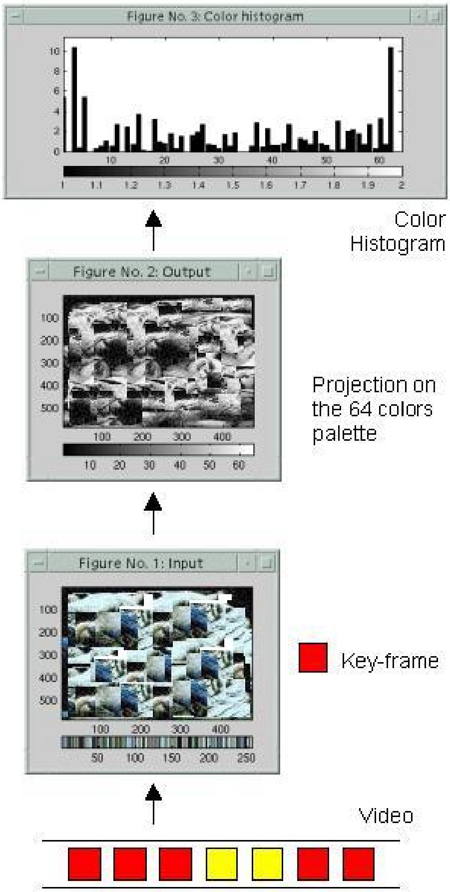

Fig. 2. Color histogram extraction. (not actual key-frames)

background color of inlay key-frames in the training video-news broadcast). We notice that the accuracy, recall and precision of this "root-rule" are extremely high (for details refer to 3). This result points out that only a few numbers of colors is needed to discriminate the presence of inlay key-frames. And more generally, we confirm the empirical observation that the use of colors can be used for discriminating the appearance of inlays in a single video.

This rule confirms the intuition and seems to be trivial. But if we look closely at the fuzzy sets built by the system, we notice that the proportion of the main color of the inlay has to by inside a fuzzy range. In other words the system not only tell us what the rule is, but also that a certain range has to be respected. In fact, it was clear that the presence of a large proportion of one single color is a hint of an inlay, but too much of that single color means that it there is actually not an inlay. In our case the percentage of white had to be greater than 2.75% and less (fuzzy membership) 12.03% (see Figure 3).

Fig. 3. Membership function describing low and large percentage of white color

Another interesting observation is that other colors (other than white) are used to determine the presence of the inlays. By studying this carefully, we found out that this is due to a bad projection (or detection) of the colors to the reference palette. This kind of problem is common in any visual indexing system. Here, using the fuzzy decision tree, we not only discover potential problems, but also the system provides a solution: to use the "wrong projected" colors.

Notice that the fuzziness we are dealing with in these experiments is only related to the proportions of colors. But it is clear that further research should focus on the uncertainty related to the projection of the colors.

In order to compare the Salammbô algorithm with regard to other learning algorithms, we conducted a second experiment (see Table 1). For the other algorithms we used the free software Weka 8. We remark that here, the recall and precision values of the model are as important as the accuracy of the model: it is important to perfectly recognize at least one kind of key-frames (with or without inlays in this case).

We can observe that our fuzzy decision tree method is not only among the methods with the highest accuracy, but also, presents high recall and precision rates for the inlays recognition. Moreover, it appears that the construction of fuzzy decision trees by Salammbô is also among the lowest time consuming methods, an important property for multimedia applications. This, in addition to the understandability of the fuzzy decision tree model, Salammbô presents better accuracy and quickness ratio than any other of the tested methods.

Table 1. Results for inlays recognition (64-colors palette)

Algorithm	**Accur. (%)**	**With inlays**		**Without inlays**		**Bld. Time (s)**
		Recall	**Precision**	**Recall**	**Precision**	
Salammbô (FDT)	81.3	0.88	0.78	0.75	0.86	1
Naive Bayes	54.6	0.39	0.57	0.71	0.53	0.4
Voted Perceptron	71.6	0.71	0.72	0.73	0.71	0.6
Weka J48 (C4.5)	78.4	0.75	0.81	0.82	0.77	1
Decision table	82.9	0.87	0.81	0.8	0.85	3.3
Neural Network	80.1	0.84	0.78	0.76	0.83	322

3.2 Discovering Errors in the Shot Detection

Fundamental information for structuring a video is the shot detection. A shot is a sequence filmed by one camera without any cuts and can be considered as the basis for a macro-segmentation. Even though it is claimed that this information is, at today's state of the art, easy to extract, we observed that the shot detection tools produce in general a lot of false positives.

Thus, we have conducted two experiments to see if with a very naïve approach, we are able to discover when two successive key-frames are part of the same shot. And this only based on colors proportions. In reality, our mining problem is a bit more complex than just a shot detection, since we used as training base and test base, only key-frames from errors of a shot detection tool. We expect like this, not only to test the fuzzy decision trees, but also to mine knowledge, useful for the improvement of the shot detection tools.

In order to test the trees, in the first experiment, we did not use any a priori knowledge, while in the second we injected some knowledge: the correspondence between colors.

The first training set was made out 92 learning examples separated in two groups (46 for each class). The first group of training examples was composed by two successive key-frames from two successive shots (class "*different shot*"). Then each key-frame was vectorized in a 64 colors palette. Finally, the training vector was built by merging the two vectors. We obtained a single training example with 128 features. Notice that there is no information about the relationship between colors (for instance color *i* and color *i*+64). The second group (class "*the same shots*") was also composed by two successive key-frames, but this time they belonged to the same shots, even if the shot detection tool considered (wrongly) as from different ones.

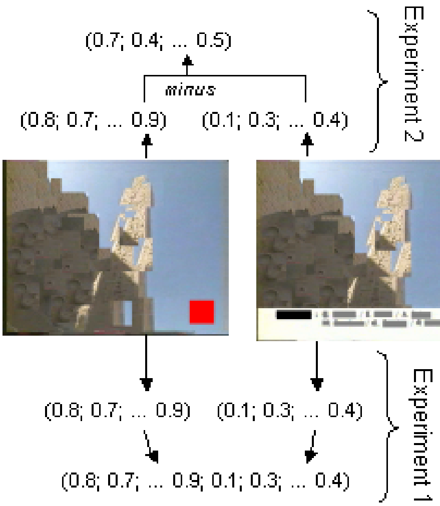

Fig. 4. Construction of training vectors for the shot error detection experiments

By observing the great resemblance of the key-frames for each error, we expected to find rules translating the similarity. But, the built fuzzy decision tree highlights that the most discriminant approach is to detect the increase of the proportion of a particular color: white (here the dominant color of inlays). For this experiment, the mean accuracy with cross-validation of the fuzzy decision trees was 78.26%.

This rule suggests that the most common shot detection error is the appearance of inlays. We also learned that after a classical shot detection (usually based on similarities), the accuracy of the shot detection can be enhanced using the color dissimilarity between successive key-frames.

Based on these results, we set up a second experiment, where we forced in the idea of looking at the differences. This time the training vectors were built in a different way: Instead of merging the two 64 features into a 128 color vector, we computed the difference color by color (for instance color i minus color i+64) to obtain a 64 color difference vector. This time with the exact same data set, we obtained as mean accuracy with cross-validation 86.9% (an improvement of 11% with respect to the first experiment). The rules points out again that the reason of the errors is the appearance of the inlays, but this time the model is more accurate. This reveals one limitation of the decision trees, the incapacity to combine different attributes.

3.3 Discovering Host, Diagram, Correspondent

Another crucial hint about the structure of the news is to detect the appearance of the host. Thus an experiment was conducted in order to recognize the presence or non-presence of the host (anchor) in a key-frame.

As in previous experiments, a training set of 50 learning examples was constructed. A first group of 25 training examples was composed by key-frames, where the host appears. A second group was composed by key-frames (randomly chosen) without host. Each key-frame was vectorized in a 256 colors palette, which was the best size for this experiment.

Based on this example dataset and on the size of the face of the host in the positive examples, we expected to obtain a rule related to the skin color (sort of simple face recognition system). But the most effective fuzzy rule to recognize if we are in presence of the host (or not) is based on a color of the host's background (presently, one specific blue color). This rule points out that the best way to know if we have a host is to look if the scene takes place in the studio. In fact, with this rule we can differentiate the host from a journalist or a guest not in the studio. In Figure 5 we see the region where the color is present (dashed area).

The accuracy of the fuzzy decision tree here is 88%.

3.4 Discussion about the Extracted Knowledge

For these three different problems, the extracted knowledge is in the form of three unexpected seminal rules. Note that we did not introduce any a priori knowledge.

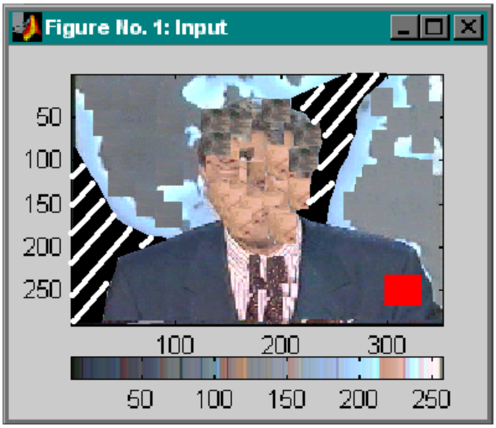

Fig. 5. Localization (dashed area) of discriminating color in host presence detection

In order to recognize the presence of inlays, the system suggests detecting a large proportion of a single color, putting forward that all inlays are large forms of similar colors (inside one video).

In order to detect the presence of a host, we should focus on the background, which corresponds to determine if the scene was taken in the studios. Even if it seem a paradox not to look at the host to detect his presence, this rules is more effective than looking at the characteristics of the host himself. It is clear that this rule cannot be directly used for another news broadcast with a different studio environment. But what is important here is that now we know (knowledge) that inside one single broadcast in order to effectively detect the anchor we should focus on the studio environment (background).

In order to ameliorate the shot detection (which is naturally based on similarity) it is recommended to look at the differences between key-frames. More precisely, the rule suggests that an inlay has been detected, implying that this is actually the cause of the errors.

We would like to remind here that this rules are intended as guidelines for the development of better indexing tools (i.e. useful knowledge), and not directly applicable for indexing. The extracted knowledge depends always on the used dataset. Here, we worked on one single video file and therefore the rules could be only used as indexing rules for the same type or similar video news program. But if we wanted to use the extracted rules directly on any video news, then we should train our fuzzy decision trees on a representative dataset containing all type of video news. The risk of such approach is that the variety of formats will melt the interesting knowledge and we will not able to find any interesting rule. Our results tell us what are the discriminating features once we are working on one single type of video journal and not how to know everything in any case. In order to get more general, what should be done instead is to check if the same type of rules is also discriminating for other types of format of video journals. We are currently working on this problem.

4 Conclusion

In this paper, we presented an example of multimedia knowledge discovery applied to the structuring of news in a video format. We used fuzzy decision trees, because of their simplicity and because of the understandability of the extracted rules. We focused on the mining of the visual aspects and in particular the color feature of the key-frames. The extracted rules provided hints for better indexing in a structuring perspective. In fact, these rules deal with appearance of important information on the screen (inlays), the presence of the host, or correcting possible mistakes in the shot detection.

This is a first step in the direction of a complete multimedia mining of a video (i.e. cross-media mining). But still in this only one media approach shows it potentiality. The next step is to continue the mining of visual contents as for instance the texture and also the mining of structural content (length of a shot, type of transition). Future work will consider the other medias (sound, text, etc.) in order to exploit the interaction. We will in particular compare a per-media mining and then fusion with an altogether mining.

Acknowledgments

I would like to thank Christophe Marsala for his comments and for providing the Salammbô software. This work was partially founded by the European project KLIMT: Knowledge InterMedation Technology (ITEA 00008).

References

1. C. Marsala and B. Bouchon-Meunier, An Adaptable System to Construct Fuzzy Decision Trees in Proceedings of NAFIPS'99 (New-York, June 1999), pp. 223-227.
2. M. Detyniecki and C. Marsala, Fuzzy inductive learning for multimedia mining in Proceedings of EUSFLAT'01 (Leicester, UK, Sept. 200), pp. 390-393.
3. M. Detyniecki and C. Marsala, Fuzzy Multimedia Mining Applied to Video News in Proceedings of the IPMU'02 Conference (Annecy, France, July 2002), pp. 1001-1008.
4. U. M. Fayyad, G. Piatetsky-Shapiro and P. Smyth, From Data Mining to Knowledge Discovery in Databases in AI Magazine, 17:3, pp. 37-54, 1996.
5. First Workshop of Multimedia Data Mining. http://www.cs.ualberta.ca/~zaiane/mdm_kdd2000/
6. B. Bouchon-Meunier, C. Marsala and M. Ramdani, Learning from Imperfect Data in Fuzzy Information Engineering: a Guided Tour of Applications, D. Dubois, H. Prade and R. R. Yager eds, chapter 8, pp. 139-148, 1997.
7. R. Ruiloba and P. Joly, Framework for evaluation of video-to-shots segmentation algorithms in Video Data special issue of Networking and Information Systems, Vol. 3, pp. 46-57, 2000.
8. Weka 3 - Machine Learning Software in Java. http://www.cs.waikato.ac.nz/~ml/weka/