Content-based discovery of multiple structures from episodes of recurrent TV programs based on grammatical inference

Bingqing Qu, Félicien Vallet, Jean Carrive, Guillaume Gravier

Content-based discovery of multiple structures from episodes of recurrent TV programs based on grammatical inference

Bingqing Qu[1,3], Félicien Vallet[3], Jean Carrive[3] and Guillaume Gravier[2]

[1] University of Rennes 1 – IRISA & Inria Rennes, France
[2] CNRS – IRISA & Inria Rennes, France
[3] French National Audiovisual Institute, France
{bqu,fvallet,jcarrive}@ina.fr, guillaume.gravier@irisa.fr

Abstract. TV program structuring is essential for program indexing and retrieval. Practically, various types of programs lead to a diversity of program structures. In addition, several episodes of a recurrent program might exhibit different structures. Previous work mostly relies on supervised approaches by adopting prior knowledge about program structures. In this paper, we address the problem of unsupervised program structuring with minimal prior knowledge about the programs. We propose an approach to identify multiple structures and infer structural grammars for recurrent TV programs of different types. It involves three sub-problems: i) we determine the structural elements contained in programs with minimal knowledge about which type of elements may be present; ii) we identify multiple structures for the programs if any and model the structures of programs; iii) we generate the structural grammar for each corresponding structure. Finally, we conduct use cases on real recurrent programs of three different types to demonstrate the effectiveness of proposed approach.

1 Introduction

With the sharp increase of multimedia contents that are broadcasted on television, there is an emerging need for efficient audiovisual information management, including browsing, indexing and retrieval. All of these tasks can benefit from TV program structuring, the goal of which is to temporally segment programs into their basic constitutive elements. Practically, different programs have their own rule and methodology for edition, which leads to various structures. Most approaches for program structuring assume that the structure of the program is known as prior knowledge, for example in the case of sports [1, 2] and news [3, 4] which have received tremendous attention. To skirt the supervised issue, recent work exploit the properties of recurrence of some TV programs to infer the structure with minimal prior knowledge, such as [5] and [6] both focusing on recurrent TV programs.

In this paper, we follow this last path and attempt to discover the structure of a program via the analysis of a collection of episodes of the recurrent program. Recurrent programs designate programs with multiple episodes periodically broadcasted (e.g., daily, weekly): Such programs include TV news, magazines and entertainments and are frequently seen on all television channels. Let's take TV news, a typical recurrent program, as an example to further understand recurrent TV programs: TV news usually start with a brief outline of the reports (*outline*), followed by an alternation of anchorperson's announcement of the upcoming topics and news report (*news content*). Most news programs end with interview segments (*interview*), sports or program trailers (*trailer*). We designate such constitutive elements of the program as *structural elements* which refer to a video segment with a particular syntactic meaning. Most episodes of recurrent TV programs follow the same editorial structure: Structural elements repeat across episodes and appear in almost the same order with similar duration. In summary, recurrent TV programs usually have two structural properties:

i. *element repetitiveness* and ii. *temporal stability*. The stable temporal structure of recurrent TV programs allows generating a common model to present the overall structure of the program, as proposed in previous work [6, 7] where a grammar of the program is inferred. However, across episodes, the organization of a program may partly vary according to different factors. Considering the TV news example above: The days when there are invited people, the right episodes usually end with the *interview*, while the days when a new film is on, the episodes usually end with the *trailer*. As a result two different structure models could be found for the same program, i.e., *multiple structures* may exist for the program, a fact that was disregarded in previous attempts to infer a grammar from a collection of episodes.

With this idea in mind, we address the problem of grammatical inference to discover the structure of recurrent TV programs allowing the discovery of multiple structures. To avoid any confusion, we propose to use the term *program* to refer to a recurrent TV program and *episode* to refer to an exemplar of the program. A recurrent program usually has quite a few episodes, which obviously leads to very hard workload if all episodes are processed. Since recurrent programs have stable structure across episodes, we adopt a collection of episodes from the program to discover the structures and generate the corresponding *structural grammars*, i.e., a graphical representation of the structural model representing the structure of the program, including structural elements, their temporal organizations, relative duration as well as presence probabilities. In practice, the inferred grammar can be utilized to process additional episodes from the same program, which is the application scenario of this paper, or to acess and browse particular events within the collection of episodes.

Our work is elaborated in an unsupervised way with the assumption that we have minimal prior knowledge about the program genre, the existence of multiple structures and the types of structural elements which may be present. Considering the absence of prior knowledge, we design a three-stage method to determine the structural elements and model the structure which is evidentiated by one or several grammars, depending on the number of structures found. Based on a collection of episodes, firstly we adopt a large number of audiovisual detectors to detect general events, which refer to the audiovisual segments with basic features or information of video content, such as, monochrome image sequences, silence segments, etc. The structural elements are then discovered based on the repetitiveness property using density filtering. Secondly, in order to identify the existence of multiple structures, we adopt a hierarchical clustering method to classify the episodes into different clusters, where the number of clusters is automatically determined. For each cluster, we statistically model the structure of the episodes using a categorical distribution matrix. Finally, a graphical representation of the grammars embedded in the distribution matrix is generated by segmenting the structure of the program into coherent states.

The rest of the paper is organized as follows. Section 2 reviews the existing techniques for TV program structuring. Section 3 formally states the problem and the overall method. Section 4 describes the method to identify the structural elements. Section 5 explains the multiple structure identification and structure modeling. Section 6 introduces the structural grammar generation. Experimental evaluations are reported in Sect. 7, followed by a conclusion in Sect. 8.

2 Related work

A number of TV program structuring techniques has been proposed in literature. Existing approaches can be classified in two categories, according to whether prior knowledge of the program structure is used or not.

In the first category, [1] and [2] target sports programs by integrating domain knowledge about the sports and editing rules to modle the structure with a hidden Markov model. In the news domain, [3] and [4] automatically annotate the news content to identify typical video shots. These studies target the entire

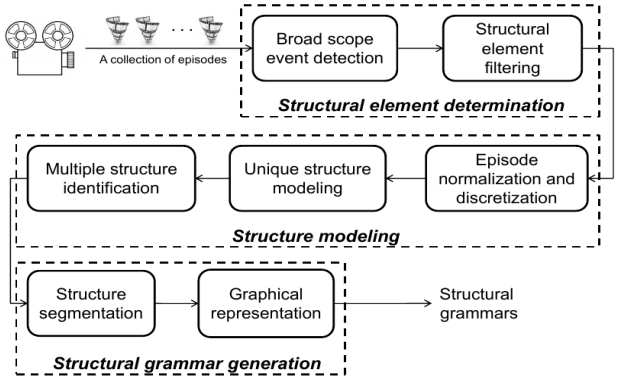


Fig. 1. General architecture of the three-stage approach for the grammatical inference of program structuring.

structure of a specific type of programs. Alternate work focuses on detecting typical structural elements. In [8] and [9] anchorpersons are detected in newscasts. In [10] and [11] goal events are identified for soccer videos. Most of these approaches are however supervised since prior knowledge of program structure is required.

In the second category, the problem of program structuring is addressed without prior knowledge of the structure and the structure of the program is discovered by generic solutions. For example, the event repetitiveness is leveraged by considering visual recurrence in [5] to detect separators, where a *separator* refers to certain short sequences which may appear before or after these structural elements to signal their starts or ends. In [12], recurrent segments exhibiting audiovisual consistency are discovered and filtered according to their relevance to the video structure, while [13] adopts a frequent pattern approach for anchorperson detection and other related purposes. These approaches focus on detecting a typical structural element by considering their inherent properties across diverse types of programs. Recent work also addresses the entire structure of programs in an unsupervised manner. In [6] and [7], taking advantage of the property of recurrence, the underlying structure of recurrent programs is explored by grammatical inference using multiple sequence alignment technique. The approaches of this category try to shift from supervised to unsupervised techniques for program structuring, incorporating minimal prior knowledge.

Taking the last path, our work also makes use of the property of recurrence to structure TV programs. Extending previous work on the subject, we utilize a statistical modeling method instead of multiple sequence alignment to model the structure, allowing us to further identify the existence of multiple structures within a collection of episodes. While multiple sequence alignment failed at dealing with the existence of multiple structures, in particular due to the representation adopted, the statistical modeling approach taken here allows for clustering to detect the existence of various structures which are combined in a probabilistic grammar in the end.

3 Overall method

In this paper, we exploit grammatical inference to discover multiple structures by analyzing contents from a collection of episodes of a recurrent TV program, in the absence of prior knowledge about the program genre, the existence of multiple structures and the types of structural elements which may be present. Facing these challenges, a three-stage approach is proposed as depicted in Fig. 1, discovering the structural elements, modeling the structures and generating the structural grammars.

Let $EP = \{ep_1, ep_2, .., ep_N\}$ be a collection of N episodes from a recurrent TV program. Firstly, structural elements, i.e., the elements relevant to the structure of the program, are discovered by leveraging a broad scope of audiovisual detectors and analyzing the temporal distribution of these general events. We denote $E = \{E_1, E_2, ..., E_d\}$ as d detected structural elements. In the second stage, the length of episodes in the collection is normalized to T, and each episode is segmented into T uniform time intervals. Relying on a hierarchical clustering method, the episodes are classified into M different clusters $C = \{C_1, C_2, ..., C_M\}$, corresponding to M different structures for the program. For each cluster, a statistical model corresponding to one of the structure of the program is obtained, yielding a collection of structural models denoted as $S = \{S_1, S_2, ..., S_M\}$. In the last stage, the structural grammar is obtained by segmenting each structure into coherent structural states and adopting symbolic representation for the purpose of concise visualization of the structural models. Hence, M corresponding grammars are generated for the program, denoted as $G = \{G_1, G_2, ..., G_M\}$.

In the following sections, we detail each stage of the proposed approach for program structuring of recurrent TV programs.

4 Structural element determination

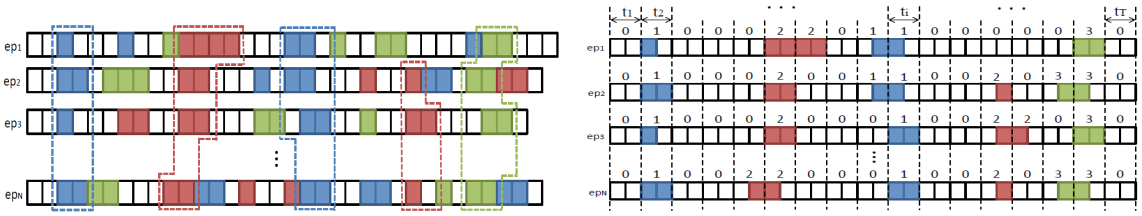
Discovering the elements composing the program is an essential task for program structuring. However, given the assumption of having no prior knowledge about the types of structural elements which may be present in the program, we decide to mine possible elements for each program from all existing ones, giving the priority to the basic and common elements. With this proposition, we apply a number of audiovisual detectors to the programs and filter out the valid elements by means of the property of element repetitiveness. This method for structural element determination is successfully used in [6] and [7].

4.1 Broad scope event detection

In order to discover structural elements generic enough for various types of programs, a large number of event detectors should be adopted to firstly detect general purpose events which may potentially be relevant to the structure of programs. Practically, considering a trade-off between the type of programs, the complexity at run time and implementation issues, nine key detectors are applied. Among them, seven are visual detectors: shot detector, dissolve detector, monochrome image detector, text region detector, motion activity detector, person clustering, as well as shot reverse shot detector. These visual detectors are the common ones for discovering the basic visual features of TV programs. Besides, two audio detectors, i.e., speech/music/silence detector and audio recurrence detector, aim at detecting generic audio features for TV program structuring.

4.2 Structural element filtering

A considerable amount of general events is detected in the previous stage, but they are not all relevant to the structure of recurrent programs. For instance, a short sequence of black frames could be a separator inserted between two successive parts of the program, however it could be also found in a night scene. In order to identify the valid structural elements, we adopted two strategies, i.e., *role recognition* and *density filtering*. On the one hand, based on person clustering, role recognition aims at identifying the persons who have an important role in the program, such as anchors, conductors, etc. The presence of these role is always a strong cue with respect to the structure. Role recognition is realized by comparing the time related features defined for characterizing each cluster, and identifying the person who has the most significant features as



(a) Episodes with structural elements (b) Episodes after normalization and discretization

Fig. 2. An example of structural element determination and episode pre-processing

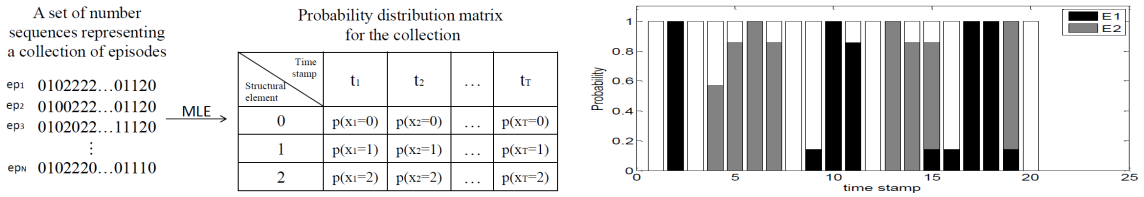
a dominant one (i.e., anchors or conductors) for each episode. On the other hand, density filtering analyzes the temporal distribution of general events, and filters out the ones which repeat with relative temporal stability across episodes. By leveraging the property of repetitiveness of structural element across episodes, the valid elements are selected based on the temporal density analysis with a Gaussian kernel function. In Fig. 2(a), we illustrate a collection of episodes, depicting the events detected (colored rectangles) as well as the structural elements resulting from density filtering (enclosed in dashed boxes).

5 Program structure modeling

After obtaining the structural elements composing each program along with their respective positions in each episode, we now study the task of representing and modeling the structure. As mentioned earlier, recurrent TV programs have stable temporal structures across episodes, which allows yielding a common structure for the program. However, multiple structures may exist for one program. Taking the example given in the Sec. 1, there are two structures in the news program. Sharing the same structural elements at the beginning, one ends with *interview* when people are invited, while the other ends with *trailer* when a new film is on. This motivates us to utilize a clustering technique to identify multiple structures. The proposed structure modeling method was successfully used in [14] to model periodic behaviors for moving objects. Before identifying and modeling the program structures, pre-processing is done in order to obtain a suitable representation yielding more concise and informative structures.

5.1 Episode normalization and discretization

In practice, the episodes from the same recurrent program do not have exactly the same length, or the same structural elements neither on type nor on quantity. It brings great difficulties to realize the clustering technique. In order to facilitate the task of clustering, we pre-process the episodes by normalization and discretization to translate the episodes into digital number sequences. We segment each episode into T uniform intervals after normalizing the length of the episodes to T. Let t_i $(t_1 \leq t_i \leq t_T)$ denote the i-th relative time stamp in T. As illustrated in Fig. 2(b), given a set of relative time stamp $TS = \{t_1, t_2, ..., t_T\}$, we mark each time stamp t_i with a number k $(1 \leq k \leq d)$, each number representing a structural element E_k from the d detected structural elements $E = \{E_1, E_2, ..., E_d\}$. We mention that the number "0" denotes the absence of any structural element. If more than one element are detected in an interval, we take the one who has longer duration in the time stamp as the valid element. In the extreme case where two elements are found in the same time stamp with the same duration, we consider the type priority to choose the valid one, because some types of elements have a stronger clue for program structures than others, such as separator or anchor. Hence each episode can be represented by a number sequence, while a collection of episodes produces a set of number sequences.



(a) Generation of probability distribution matrix (b) Illustration of probability distribution matrix

Fig. 3. Structure modeling for unique structure with two structural elements.

5.2 Unique structure modeling

While modeling the structure, we assume that a program is homogeneously structured, i.e., *unique structure*, before moving to the step of multiple structure identification.

Considering a collection of episodes $EP = \{ep_1, ep_2, ..., ep_N\}$ with d discovered structural elements $E = \{E_1, E_2, ..., E_d\}$, we have a set of number sequences corresponding to the episodes. We use ep_j to denote the j-th episode, and e_i^j to denote the structural element in the i-th relative time stamp in the episode. Thus $e_i^j = k$ $(0 \leq k \leq d)$ means that the structural element is k at t_i in the j-th episode. For example, in news program, two structural elements, $E = \{E_1, E_2\}$, are determined, where E_1 represents *separators*, E_2 represents *outlines*. Therefore, $e_2^5 = 2$ means at t_2 in the 5th episode, the valid structural element is the *outline*, while $e_{10}^3 = 0$ means at t_{10} in the 3rd episode, no determined element is found.

Naturally, we may use the categorical distribution to model the element probability for TV programs. Let x_i be the categorical random variable representing the type of structural elements at time stamp t_i. We propose $\mathbf{P} = [\mathbf{P}_1, \mathbf{P}_2, ..., \mathbf{P}_T]$ to be a categorical distribution matrix, where each column $\mathbf{P}_i = [p(x_i = 0), p(x_i = 1), ..., p(x_i = d)]^T$ is an independent probability vector satisfying $\sum_{k=0}^{d} p(x_i = k) = 1$. The structure of the collection $EP = \{ep_1, ep_2, ..., ep_N\}$ modeled by some distribution matrix $\mathbf{P}$ yields

$$P(EP \mid \mathbf{P}) = \prod_{ep_j \in EP} \prod_{i=1}^{T} p(x_i = e_i^j) \tag{1}$$

considering all episodes and time intervals independent. From the estimation point of view, the best maximum likelighood model is given by

$$p(x_i = k) = \frac{\sum_{ep_j \in EP} 1_{e_i^j = k}}{|EP|} \tag{2}$$

where $\sum 1_{e_i^j = k}$ indicates the number of occurrences while $e_i^j = k$. In other words, $p(x_i = k)$ is the relative frequency of structural element E_k at t_i across all episodes from EP.

As seen in Fig. 3(a), a simple example with two discovered structural elements is given. A set of number sequences issued from a collection of episodes is modeled with a probability distribution matrix. In Fig. 3(b), an illustration of the probability matrix represents the time ordered structural elements and their relative probability at each time stamp.

5.3 Multiple structure identification

The model representation clearly assumes that the structure is unique across episodes, which is not always the case in practice. Recurrent programs may have different structures according to the editorial rules.

Given a collection of episodes from the same program, we aim at identifying multiple structures using a clustering technique. Supposing that there are M underlying structures, i.e., $S = \{S_1, S_2, ..., S_M\}$, each of which has its own structural elements and probability distribution matrix $\mathbf{P} = \{\mathbf{P}^1, \mathbf{P}^2, ..., \mathbf{P}^M\}$, the episodes $EP = \{ep_1, ep_2, ..., ep_N\}$ should be partitioned into M clusters, i.e., $C = \{C_1, C_2, ..., C_M\}$, so each cluster represents one structure. There are many clustering techniques to classify the episodes into M clusters, however the number of underlying structures (i.e., M) is unknown beforehand. So we propose a hierarchical agglomerative clustering method to group the episodes while at the same time determining the optimal number of structures.

In order to adopt the clustering technique, a distance measure between two clusters, i.e., two structures, needs to be defined. Since the structure is represented by a probability distribution matrix, the distance between two structures is determined by their probability distribution matrices. Several methods are available for measuring the distance between two probability distribution matrices. Here we propose to utilize the Jensen-Shannon divergence (*JS divergence*), because it measures the similarity between two probability distributions with a symmetric and bounded value, required for hierarchical clustering techniques. The JS divergence, based on the Kullback-Leibler divergence, between two probability distribution matrices $\mathbf{P}$ and $\mathbf{Q}$ is defined as:

$$D_{JS}(\mathbf{P} \parallel \mathbf{Q}) = \frac{1}{2} D_{KL}(\mathbf{P} \parallel \mathbf{M}) + \frac{1}{2} D_{KL}(\mathbf{Q} \parallel \mathbf{M}) \tag{3}$$

where $\mathbf{M} = \frac{1}{2}(\mathbf{P} + \mathbf{Q})$, and $D_{KL}()$ denotes the Kullback-Leibler divergence. A small $D_{JS}(\mathbf{P} \parallel \mathbf{Q})$ means that two distribution matrices $\mathbf{P}$ and $\mathbf{Q}$ are similar, and conversely. We define the distance between two structures as

$$dist(S_1, S_2) = dist(C_1, C_2) = D_{JS}(\mathbf{P}^1 \parallel \mathbf{P}^2) \tag{4}$$

During each iteration of the hierarchical clustering, two clusters with minimal distance are merged, the new cluster inheriting the episodes owned by the original clusters C_1 and C_2. The probability distribution matrix corresponding to the newly merged cluster is computed as

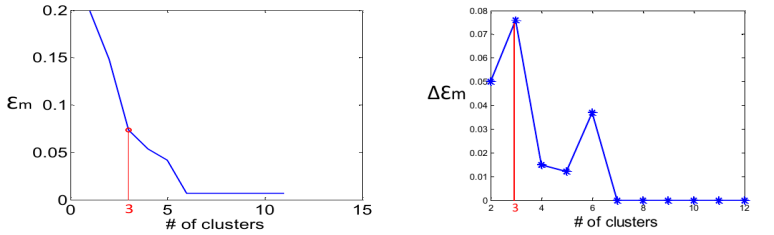
$$\mathbf{P}^{new} = \frac{\mid C_1 \mid}{\mid C_1 \mid + \mid C_2 \mid} \cdot \mathbf{P}^1 + \frac{\mid C_2 \mid}{\mid C_1 \mid + \mid C_2 \mid} \cdot \mathbf{P}^2 \tag{5}$$

With such distance measure and cluster merge method, we now identify the multiple structure of programs. However without the number of underlying structures, M is unknown. We propose to determine the optimal cluster number while exhibiting hierarchical clustering by monitoring the quality of clusters using an *impurity factor*. At each iteration of clustering, two clusters with minimal distance are merged and the number of clusters turns from m to $m-1$. At each iteration, an impurity factor is computed as

$$\varepsilon_m = \frac{1}{m} \sum_{i=1}^{m} IM(C_i) \tag{6}$$

where $IM(C_i)$ is the impurity [14] of each cluster, defined as

$$IM(C) = \frac{\sum_{ep_j \in EP} \sum_{i=1}^{T} 1_{e_i^j \neq 0} \cdot (1 - p(x_i = e_i^j))}{\sum_{ep_j \in EP} \sum_{i=1}^{T} 1_{e_i^j \neq 0}} \tag{7}$$



(a) Impurity factor (b) Increase of impurity factor from m to $m-1$

Fig. 4. Determination of cluster number by impurity factor

During the process of clustering, we monitor the change of ε_m. When the number of clusters turns from m to $m-1$, if the impurity factor ε_m increases significantly, it indicates that m might be the correct structure number. Because that the significant increase of ε_m signifies that the newly merged cluster may contain two different structures. Therefore, the optimal structure number M is determined when the impurity factor has maximal increase while the cluster number turns from N, i.e., the total episode number of the collection, to 1. We illustrate a real example of news program in Fig. 4, the impurity factor suddenly increases at $m = 3$. It indicates that there are three structures for the program.

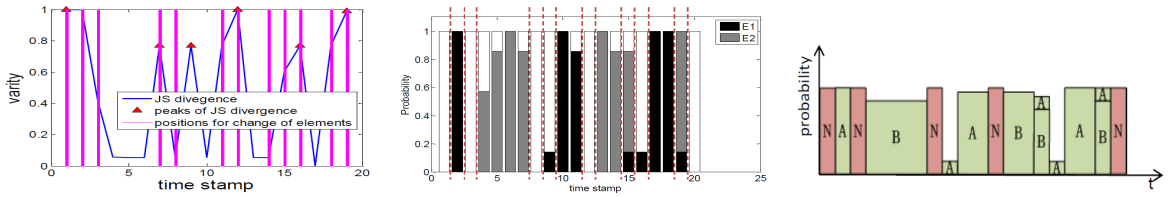
Finally, we are able to identify multiple structures for a recurrent TV program by a hierarchical clustering method with an automatically determined structure number. These discovered structures mean to be common models to represent the structures of the recurrent program. However, verifying the clustering results obtained above, we observe the case that in certain clusters there exists just one episode. These phenomena can be explained by the fact that the isolated structure may result from an episode badly processed in earlier stages or some special made episodes, such as episode on Christmas or other festivals. So we treat the episode as an isolated structure for which no grammar can be made to make sure that the discovered structure is common enough to represent the program. We can observe this case in the example in Fig. 4(a), so practically the number of structures for the news program is deemed as two instead of three.

6 Structural grammar generation

Having discovered all structures for a recurrent program, we further generate structural grammars for the more concise visualization of structural models. Since every discovered structure for the program will be considered separately, the rest of this section will focus on one specific structure.

6.1 Structure segmentation

In the stage of structure modeling, the episodes are segmented into uniform time stamps. Therefore the structure represented by a probability distribution matrix is based on the time stamps, as shown in Fig. 3(b). This illustration provides a global understanding of the structure, however it has very limited abstraction capabilities and is not concise enough owning to some time stamps repeating with the same structural elements and similar probability distribution. In order to better visualize structures, we propose to separate the time stamps into coherent states, each of which has the same structural elements with similar probability distribution. To achieve this goal, we verify between two successive time stamps the variations of two indicators, i.e., the JS divergence and the composition of structural elements. If one of the two indicators is changed, we consider that there is a rupture of state. More concretely, as illustrated in Fig. 5(a), the



(a) State change positions (b) Structure segmentation (c) Structural grammar

Fig. 5. Illustration of structural grammar inference

positions where the JS divergence has a peak or where the combination of structural elements are changed are deemed to be a rupture for coherent states. The peaks of JS divergence imply that in the corresponding positions the distribution vectors at two successive time stamps are far too different. Practically, while segmenting the time stamps, we ignore the elements who has small presence probability if necessary to clarify the structures. In the Fig. 5(b), the structure is divided into different coherent states according to the Fig. 5(a).

6.2 Graphical representation

In order to generate a concise grammar with enough structural information, we leverage a graphical representation for grammars, adopting symbolic representation to present structural elements based on prior knowledge of common TV programs. For instance, a structural element corresponding to a sequence of white frames is a separator, denoted as S, while a long duration shot containing the dominant person at the beginning of a program is the conductor's opening, denoted as A. The grammar is designed by considering the structural elements, their relative duration and probabilities. As illustrated in Fig. 5(c), two structural elements, i.e., A and B, are determined, while the segments corresponding to no determined structural element are denoted as N. A rectangle filled with a symbol is used to illustrate each state for the program: The height of rectangles indicates the relative probability of each valid element while the width is proportional to its duration. In short, for a grammar, the elements with their relative duration and presence probabilities are introduced in time order.

7 Experimental results

We conduct some use cases on real TV programs of different types, viz., game, news and talk show, in order to show the effectiveness of our approach for recurrent program structure discovery. Considering the applicable purpose of our work, which is to facilitate structuring the additional episodes from the same program using the inferred grammars, we firstly generate grammars for the programs and then perform two quantitative evaluations, i.e., prediction accuracy and element matching recall.

Table 1. Description of the datasets for evaluation

Dataset	Date	Episodes	Type	Average duration
NEWS	2007	24	TV news	37.9 m
GAME	1991-1992	24	Game	31.9 m
TALK	2014	24	Talk show	71.3 m

Table 2. Quantitative evaluations

Evaluations	NEWS	GAME	TALK
PA	100%	87%	68%
MR	0.83	0.70	0.49

7.1 Dataset description

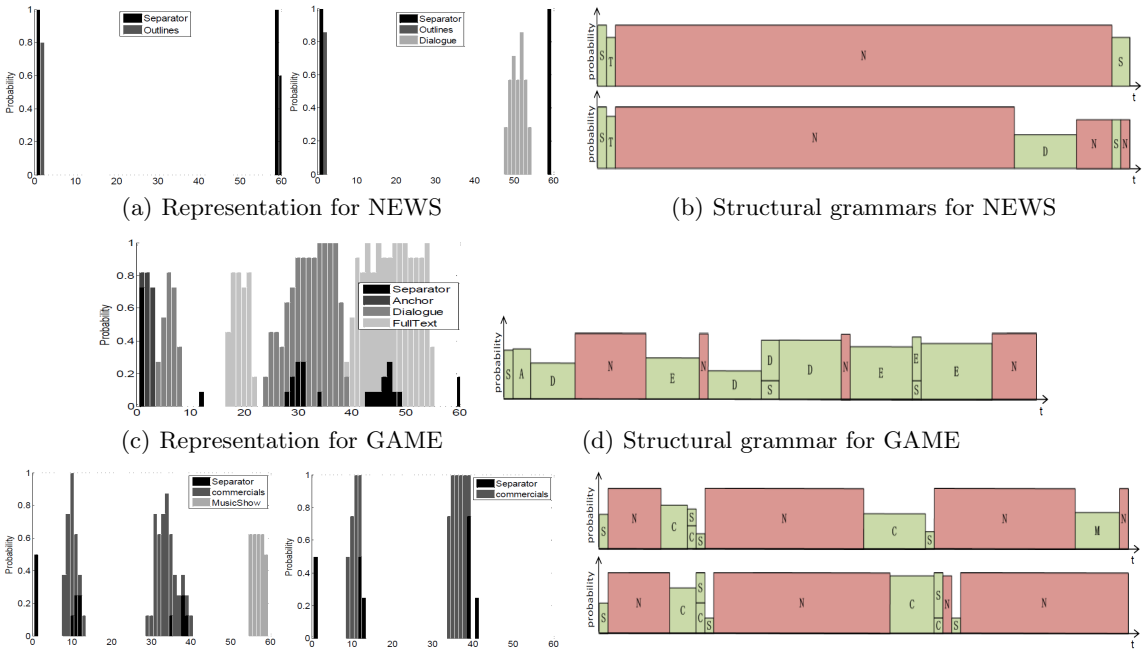
Three different programs are utilized for inference and evaluation, as given in Tab.1. We adopt 24 episodes of each program, with 12 episodes for grammar inference (*inference set*) and 12 episodes for quantitative evaluation (*test set*). *20h News* (NEWS), a daily news show of 2007, follows a standard pattern for such programs. *Que le meilleur gagne* (GAME) is a game show whose episodes were taken over two years (1991 and 1992). It has four parts divided by separators. The program, hosted by a conductor, mainly contains interview scenes and question/answer scenes with full text segments. *Le grand journal* (TALK) hosted by a conductor, including news reports, talks, weathers and musical performances. The episodes are taken from the first months of 2014.

7.2 Use cases

Grammar inference For a recurrent program, we firstly use inference set to infer the structural grammars. Figure 6 shows the inferred grammars for three different programs. Figure 6(a), 6(c) and 6(e) illustrate the structures presented by distribution matrices. NEWS has two different structures, i.e., with or without interview segment, depending on whether there are invited people or not. The game show, having unique structure, is composed of separators, anchor's opening, interview segment and full screen text scene. For TALK, two structures are identified: the structure ending with/without musical performance. The different structures for the same program have evident distinction, i.e., the presence of certain structural elements, while slight difference for the rest parts results from the episodes from which the structures are inferred. Figure 6(b), 6(d) and 6(f) are the corresponding grammars for the three programs. While inferring the grammars, we ignore the elements whose presence probability is less 0.2 to clarify the grammars. Each rectangle represents a coherent state with its structural element in symbol (S:*separator*, T:*outline*, A:*anchor*, C:*commercials*, D: *dialog*, E:*full text* and M:*music show*). The red rectangle with symbol N refers to the segment no determined, which will be further studied.

Quantitative evaluation Two quantitative evaluations are performed to further verify the effectiveness of our approach: prediction accuracy and element matching recall. These experiments aim at structuring the episodes from the *test set* using the grammars inferred by the *inference set*. Prediction accuracy computes the accuracy of structural element prediction in the test set, whose structural elements are annotated with their corresponding types. In other words, we want to verify, in the test set, how many corresponding structural elements can be found in the segments predicted by grammars. In the case of multiple grammars, the best score is chosen. As illustrated in 7(a), supposing a program has M grammars and n episodes in the test set, the prediction accuracy is $PA = max(P_1, P_2, ..., P_M)$ with $P_i = n_e/(n * N_e)$, where N_e refers to the number of rectangles in the grammar which contain the determined structural elements (segments in green), and n_e refers to the segments in the test episodes which contain the corresponding elements. So the proportion between the number of correctly predicted elements in the test set and the number of inferred elements in the inference set is considered as the prediction accuracy for grammars.

Element matching recall evaluates the quality of element boundary. As in Fig. 7(b), taking a predicted segment, denoted as B, we compare it with the corresponding element in ground truth denoted as T, the matching recall is $MR = \frac{T \cap B}{T}$. If $MR = 0$, it indicates that no corresponding element is found in the predicted segments. The matching recall for the whole test set is the average recall of all segments. To be mentioned, the matching recall of each episode is computed based on the grammar which has higher prediction accuracy. In Tab .2, we list the results of quantitative evaluations for the three programs. NEWS has a prefect prediction accuracy with a matching recall equal to 0.83. This result owes to its structure

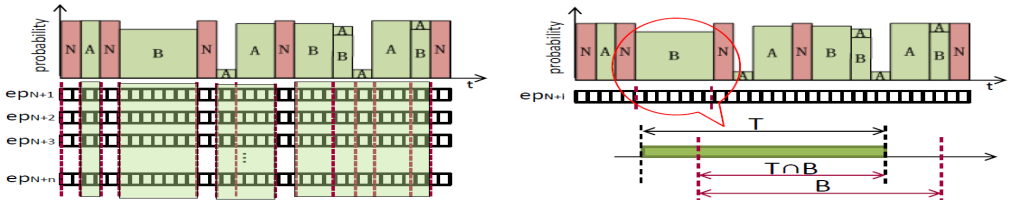


(a) Representation for NEWS (b) Structural grammars for NEWS

(c) Representation for GAME (d) Structural grammar for GAME

(e) Representation for TALK (f) Structural grammars for TALK

Fig. 6. Structures presented by distribution matrix and structural grammars for real data sets



(a) Prediction accuracy (b) Element matching recall

Fig. 7. Illustration for quantitative evaluations

stability and to the fact that less structural elements are determined. Comparing with NEWS, GAME and TALK have more structural elements determined which leads to lower prediction accuracy. Even though GAME and TALK have similar prediction accuracy, TALK has a lower element matching recall. This can be explained by two reasons. Firstly, the episodes from TALK are taken from the year 2014, the recent programs vary more in structure comparing with earlier ones, such as GAME in 1991 and 1992. Secondly, TALK has a longer duration, i.e., 71.3 minutes in average, therefore, in the uniform segmentation step, with the same number of time stamps, the longer a programs is, the coarser structural information it exhibits. If we increase the number of time stamps, a finer structure will obtained, however at the same time some redundant structural information may also comes out. So it is not wise to utilize a larger number of time stamps for a program with such duration.

8 Conclusion

In this paper, we address the problem of unsupervised structure discovery for recurrent TV programs using grammatical inference. Taking the properties of recurrent TV programs, we are able to discover the structural elements composing the programs and identify the existence of multiple structure for the programs. We proposed a three-stage approach, including structural element determination, structure modeling and grammar generation. The use cases on different types of TV programs demonstrate the effectiveness of our approach in practice. Results reported here mostly hint that structure discovery in recurrent collections using grammatical inference is viable and deserves further attention. The proposed approach can be extended in a number of directions to ameliorate TV program structuring. Obviously, more structural element could be determined to complete program structures. Elements determination beyond the repetitiveness may provide hierarchical program structures. At the same time, uniform time stamps in stage of episode discretization may no long suited, because increased structural elements requires improving structure modeling techniques. Furthermore, enriching grammars with more structural information is also an important tasks.

References

1. Xie, L., Xu, P., Chang, S.F., Divakaran, A., Sun, H.: Structure analysis of soccer video with domain knowledge and hidden Markov models. Pattern Recognition Letters (2004)
2. Kijak, E., Gravier, G., Oisel, L., Gros, P.: Audiovisual integration for tennis broadcast structuring. Multimedia Tools and Applications (2006)
3. Bertini, M., Del Bimbo, A., Pala, P.: Content-based indexing and retrieval of TV news. Pattern Recognition Letters (2001)
4. Eickeler, S., Muller, S.: Content-based video indexing of TV broadcast news using hidden Markov models. In: IEEE International Conference on Acoustics, Speech, and Signal Processing. (1999)
5. Abduraman, A.E., Berrani, S.A., Merialdo, B.: An unsupervised approach for recurrent TV program structuring. In: European Interactive TV Conference. (2011)
6. Qu, B., Vallet, F., Carrive, J., Gravier, G.: Content-based inference of hierarchical structural grammar for recurrent TV programs using multiple sequence alignment. In: IEEE International Conference on Multimedia and Expo. (2014)
7. Qu, B., Vallet, F., Carrive, J., Gravier, G.: Using grammar induction to discover the structure of recurrent TV programs. In: International Conferences on Advances in Multimedia. (2014)
8. Hanjalic, A., Lagendijk, R., Biemond, J.: Template-based detection of anchorperson shots in news programs. In: International Conference on Image Processing, IEEE (1998)
9. Gao, X., Tang, X.: Unsupervised video-shot segmentation and model-free anchorperson detection for news video story parsing. IEEE Transactions on Circuits and Systems for Video Technology (2002)
10. Chen, S.C., Shyu, M.L., Chen, M., Zhang, C.: A decision tree-based multimodal data mining framework for soccer goal detection. In: IEEE International Conference on Multimedia and Expo,. (2004)
11. Ancona, N., Cicirelli, C., Branca, A., Distante, A.: Goal detection in football by using support vector machines for classification. In: International Joint Conference on Neural Networks, IEEE (2001)
12. Ben, M., Gravier, G.: Unsupervised mining of audiovisually consistent segments in videos with application to structure analysis. In: IEEE International Conference on Multimedia and Expo. (2011)
13. Jacobs, A.: Using self-similarity matrices for structure mining on news video. In: Advances in Artificial Intelligence. Springer (2006)
14. Li, Z., Ding, B., Han, J., Kays, R., Nye, P.: Mining periodic behaviors for moving objects. In: ACM SIGKDD international conference on Knowledge discovery and data mining. (2010)