MARKOV CHAINS FUSION FOR VIDEO SCENES GENERATION

A. Piacenza, F. Guerrini, N. Adami, R. Leonardi

Department of Information Engineering, University of Brescia, Italy

ABSTRACT

In this paper we address the general issue of merging Markov chains used to model two instances of a given process with some properties in common. In particular, in this work we apply this scenario to a multimedia application that generates new video scenes mixing the original segments of a given movie. To perform the latter process, it is first necessary to describe the structure of the scenes in some way, which in our case is done through Markov chains. The video scenes are then recombined by fusing their corresponding models using the general method described here. We analyze and validate the proposed methodology only for this specific application, however the solution presented here could be used in a very diverse array of applications where Markov chains are routinely used, ranging from queuing modeling to financial decision processes.

Index Terms— Markov Chains, Video Modeling, Interactive Storytelling.

1. INTRODUCTION

Markov chains (MCs) are variously applied in many different fields, in particular in real world problems involving random processes. For example, in [1] five applications are analyzed in detail: biology, queueing models, resource management models, Markov decision processes and Monte Carlo simulations. Other applications of Markov chains include marketing forecasts [2], Google's PageRank [3] and video modeling [4, 5, 6].

Now let's focus our attention on a given application in which Markov chains represent the modeling tool of choice, such as ours, and let's suppose two models pertaining to two different entities have been obtained. If they share some properties such as the presence of common states as is the case in this work, it could be interesting to exploit this fact to obtain a new unique model that describes both entities simultaneously and that still inherits the structure of the two original models in some way.

Here, the proposed MC fusion method has been implemented as a part of the movietelling application developed during the IRIS NoE [7] that constructs new filmic variants of a baseline movie. We developed it because no existing MC processing technique satisfies our requirements to the best of our knowledge. In [7], an earlier technique to generate new narrative scenes was proposed. The objective of the MC fusion is to suggest possible video scenes to the narrative generation module, and successively an author assesses whether the scenes are of a good enough quality to be included in the narrative domain model. We evaluate the generative power of the new method using Michael Radford's screen adaptation of Shakespeare's *The Merchant of Venice* [8], as we did in [7]. In the present movietelling system, the audio portion is discarded and substituted by appropriate subtitles describing the narrative. However, we expect the fusion process to be able to easily incorporate audio features should they will be integrated in the system.

The paper is organized as follows: in Section 2 our specific framework is introduced by describing how Markov chains can model video scenes. Then, in Sections 3 and 4 we describe the general method that allows to fuse two Markov chains. Section 5 shows some experimental results, including both subjective and objective evaluations, and Section 6 draws the conclusions.

2. VIDEO MODELING

In [9, 10], a video is modeled as a *Scene Transition Graph* (STG) starting from its shot segmentation [11]. In those works, the nodes of a STG represent clusters of visually similar shots and the edges identify the transitions between consecutive shots. By removing the cut-edges from the graph, an STG can be decomposed in a number of well connected subgraphs, which represent the *Logical Story Units* (LSUs) of the video and, if the latter is a movie, the LSU concept is related with its scenes [12]. Each LSU can be equivalently modeled by a Markov chain whose states represent the visual clusters and the transition probabilities is computed by counting the temporal transitions between consecutive shots.

In our application, rather than considering the shot visual similarity, we are more interested in their semantic content. Hence the LSUs are only used for segmentation purposes and after that they must be re-clustered using some kind of semantic description of the shots. For this purpose, we define a set of semantic tags with which each shot is described: *characters* (the list of the main characters present in the shot and

This work has been funded (in part) by the European Commission under grant agreement IRIS (FP7-ICT-231824).

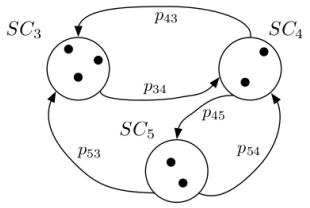


Fig. 1. Example of a Markov chain generated through semantic clustering of a movie scene. The states of the MC are the semantic clusters and the points inside them represent the shots belonging to the semantic cluster.

their mood: positive, neutral or negative); *camera field* (long, medium or close-up); and *environment*, which is a description of the time of the day (day or night), the location (indoor or outdoor) and crowd presence (crowded or not).

The reorganization of the clusters belonging to a LSU, depending on the semantic content of the shots, leads to the generation of a new graph that we call *Semantic Story Unit* (SSU), because of its strong connection with the LSU concept. In the same way an LSU is associated to a Markov chain, we can still associate a MC to the SSU concept; obviously the number of the states and the relative transition probabilities of the MC associated to an LSU and the corresponding SSU may be different. An example of a Markov chain associated to an SSU is shown in Fig. 1.

Since each SSU is mapped to a narrative scene, the purpose of fusing two SSUs using the method outlined in what follows is generating a new movie scene constructed by mixing the content of two original scenes, which would be a useful feature for our movietelling system.

3. MARKOV CHAINS FUSION METHOD

This section presents the technique that allows to fuse two separate Markov chains; our assumption is that the two chains have to share some common properties, so we impose that they possess at least one matching state to be eligible for the fusion process. Our aim is to obtain a new MC that is constructed starting from both of them and still inherit in the best possible way the properties of the two original structures, in this case the transition matrices. Moreover, we assume that two semantic clusters (the states in our model) match if they represent the same semantic concept, i.e. they share the same values for the semantic tags set.

3.1. Working Example

We will refer to Fig. 2(a) as a working example. Starting from $MC1$ and $MC2$ (top half), with transition matrices P and Q respectively, the new fused Markov chain with transition matrix R is built. Observe that $MC1$ has 4 states and $MC2$ has 3 states and there are 2 matching pairs of states (highlighted in Fig. 2(a)), hence the fused Markov chain with transition matrix R has 5 states. The transition matrices into play are:

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \end{bmatrix}, \quad Q = \begin{bmatrix} q_{55} & q_{56} & q_{57} \\ q_{65} & q_{66} & q_{67} \\ q_{75} & q_{76} & q_{77} \end{bmatrix}$$

$$R = \begin{bmatrix} r_{AA} & r_{AB} & r_{A1} & r_{A2} & r_{A7} \\ r_{BA} & r_{BB} & r_{B1} & r_{B2} & r_{B7} \\ r_{1A} & r_{1B} & r_{11} & r_{12} & r_{17} \\ r_{2A} & r_{2B} & r_{21} & r_{22} & r_{27} \\ r_{7A} & r_{7B} & r_{71} & r_{72} & r_{77} \end{bmatrix}$$

To maintain the original structure as represented by the transition matrices, all the transitions between the states not involved in the fusion process should be preserved (in this case, between S_1 and S_2) as much as possible in the matrix R.

Starting from these assumptions, we have to express each coefficient r_{ij} of the matrix R using the coefficients of the constituent matrices P and Q. The most practical solution is therefore to extend the transition matrices P and Q to take into account all the states pertaining to both of them, rearranging the rows to reflect the same row order of R. Finally, the individual matching states that are to be fused in the final fused states are identified (here, S_3 and S_5 with S_A; S_4 and S_6 with S_B). Hence, the new transition matrices are:

$$P^* = \begin{bmatrix} p_{AA} & p_{AB} & p_{A1} & p_{A2} & 0 \\ p_{BA} & p_{BB} & p_{B1} & p_{B2} & 0 \\ p_{1A} & p_{1B} & p_{11} & p_{12} & 0 \\ p_{2A} & p_{2B} & p_{21} & p_{22} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, Q^* = \begin{bmatrix} q_{AA} & q_{AB} & 0 & 0 & q_{A7} \\ q_{BA} & q_{BB} & 0 & 0 & q_{B7} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ q_{7A} & q_{7B} & 0 & 0 & q_{77} \end{bmatrix}$$

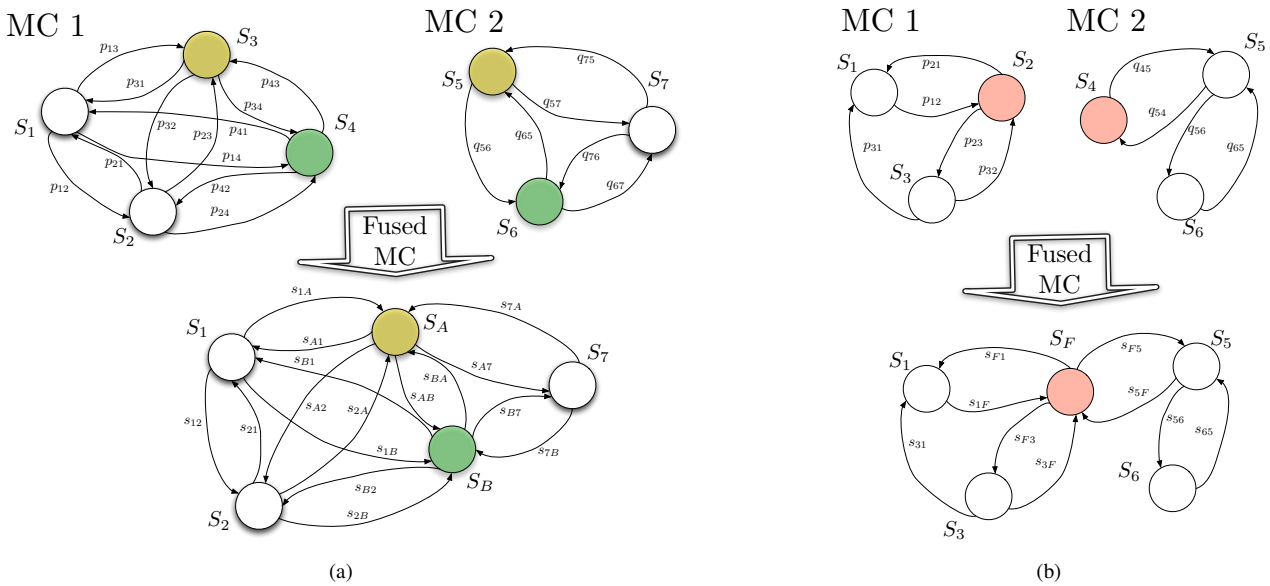
where S_A in P^* identifies with S_3 in P (S_B identifies with S_4) and S_A in Q^* identifies with S_5 in Q (S_B identifies with S_6). Now, in the transition matrix R, three different groups of rows can be identified:

$$R = \begin{bmatrix} R_F \\ R_P \\ R_Q \end{bmatrix} \tag{1}$$

In Eq. (1), R_F are the rows corresponding to the fused states (S_A and S_B), while R_P and R_Q are the rows corresponding respectively to the states in P and Q that are not involved in the fusion process (S_1 and S_2 for R_P and S_7 for R_Q). Now we express every i-th row of R as a linear combination of the i-th row of P^* and Q^*:

$$R_i = \alpha_i P_i^* + \beta_i Q_i^* \tag{2}$$

For R to make sense as a transition matrix, it is necessary that the parameters $\alpha_i + \beta_i = 1$ for all i and, by construction, α_i (resp. β_i) is equal to 1 when i corresponds to a row of R_P (resp. R_Q) and 0 when i corresponds to a row of R_Q (resp. R_P). The parameters relative to the fused states (let's call



(a) (b)

Fig. 2. Markov chains fusion process examples. (a) The fused MC has aggregated the matching states S_3 from $MC1$ and S_5 from $MC2$, and likewise S_4 and S_6, into the fused states S_A and S_B. (b) The case of a single pair of matching states. Auto-transitions are omitted for clarity.

them α_F and β_F), are the most important; in fact they represent how one chain is dominant w.r.t. the other in the fused MC. It can be observed that the transition matrix R as defined above is a "weighted average" of P and Q: for those states that have been fused, we average the corresponding transition probabilities, while those between the other are left unchanged. In this work, the parameters α_F and β_F need to be related to the cardinality of the involved semantic clusters. Let's define M_j as the *magnitude* of the j-th cluster:

$$M_j = \begin{cases} \#SC_j & \text{if } sh_L \notin SC_j \\ \#SC_j - 1 & \text{if } sh_L \in SC_j \end{cases} \quad (3)$$

where $\#SC_j$ is the cardinality of the j-th semantic cluster and sh_L is the last shot of the SSU. Eq. (3) takes into account the fact that there is no temporal transition from the last shot of the SSU. Considering a cluster with magnitude M_1 and one with magnitude M_2, we can express the parameters α_F and β_F as:

$$\alpha_F = \frac{M_1}{M_1 + M_2}; \quad \beta_F = \frac{M_2}{M_1 + M_2}$$

This leads to a fused SSU that is mostly composed by content from the SSU with higher cardinality and less content from the other.

3.2. General case

We fuse $MC1$ (with transition matrix P and extended matrix P^*) and $MC2$ (with transition matrix Q and extended matrix Q^*) with, respectively, N_1 and N_2 states; suppose that there are C pairs of matching states ($C < \min\{N_1, N_2\}$). By arranging the rows relative to the matching states in the top part of the matrix, we refer to each state of a matching pair as F_{ik} ($i = 1, ..., C$, $k = 1$ or 2), so F_{i1} is the state in $MC1$ that belongs to the i-th pair and F_{i2} is the matching state in $MC2$.

Now, P is an N_1xN_1 matrix, while Q is N_2xN_2; so the resulting matrices P^*, Q^* and R are square matrices of dimension $(N_1 + N_2 - C)$. The matrix R, in particular, can be expressed by Eq. (1). In turn, each row R_i $(i = 1, \ldots, (N1 + N2 - C))$ is expressed as in Eq. (3), where the parameters α_i and β_i are:

$$\alpha_i = \begin{cases} \alpha_{Fi} & \text{if } i = 1, \ldots, C \\ 1 & \text{if } i = C+1, \ldots, N_1 \\ 0 & \text{otherwise} \end{cases}$$

$$\beta_i = \begin{cases} \beta_{Fi} & \text{if } i = 1, \ldots, C \\ 0 & \text{if } i = C+1, \ldots, N_1 \\ 1 & \text{otherwise} \end{cases}$$

The set of the first C parameters α_F and β_F are those relative to the matching states and have to satisfy the usual relation $\alpha_{Fi} + \beta_{Fi} = 1$ for $i = 1, \ldots, C$.

4. MIXING FACTOR

With the formulation given in Section 2, no transitions between states that were not previously existing are introduced. In fact, referring to the example in Fig. 2(a) it is clear that it is not possible for the chain to jump from S_1 to S_7 without first passing through S_A or S_B.

Depending on the intended application, it could be desirable for the fused MC to allow direct transitions between the non-fused states of the constituent Markov chains even if this implies to slightly perturb the pre-existing structure. To take into account this possibility, we define a *mixing factor* $0 \leq \Phi \leq 1$ that determines the probability with which the fused MC can evolve by doing multiple steps rather than a single one as before. In this scenario, starting from a given state the chain evolves with the following rule (N controls the maximum number of steps):

- with probability $(1 - \sum_{i=1}^{N} \Phi^i)$, the chain evolves with the usual transition matrix R;
- with probability Φ, the chain evolves doing 2 steps, by applying the 2-steps transition matrix (R^2);
- ...
- with probability Φ^N, the chain evolves for $N+1$ steps, by applying the $(N+1)$-steps transition matrix (R^{N+1}).

These operations transform the transition matrix R in a new matrix $\tilde{R}$, that can be obtained as:

$$\tilde{R} = \left(1 - \sum_{i=1}^{N} \Phi^i\right) R + \Phi R^2 + \Phi^2 R^3 + \ldots + \Phi^N R^{N+1} \quad (4)$$

In particular, for the $N = 1$ (2-steps) case, $\tilde{R}$ in Eq. (4) can be written as:

$$\tilde{R} = (1 - \Phi) R + \Phi R^2 \quad (5)$$

It is clear that if in Eq. (5) Φ has a near-zero value, few "double steps" are allowed and the two original MCs are effectively isolated. On the other hand, when Φ increases, it is more likely that a 2-steps evolution occurs, thus allowing direct jumps from states that were not previously connected. The value of Φ of course can not be set automatically but has to be chosen empirically by inspecting the output videos and verifying that the mixing degree is appropriate.

5. EXPERIMENTAL RESULTS

Operating as in [6], we extracted from the movie 71 different LSUs that had been re-clusterized into 71 SSUs. Among those, only 37 were non-trivial (that is, not composed of just a single shot) and we took this ensemble as our operative set. For each SSU in the set, we investigated if there was at least another SSU with one or more matching semantic clusters, which corresponds to matching states of the associated MCs. Of the 37 considered SSUs, just 33 had at least another SSU with one or more matching clusters, i.e. the other 4 had no matching clusters among all the others SSUs. We first define the distance between SSUs, obtained averaging the cross-distances between all the shots belonging to the SSUs. The shots distance is in turn computed by extracting a codebook of visual words, which is obtained by dividing shot keyframes in square blocks and then running a Tree-Structured Vector Quantization algorithm to LUV color space values of the blocks. The codebook size is determined by controlling the distortion on the reconstructed keyframe. Finally, the shots distance is defined by averaging the distortion increase caused by representing each shot using the codebook of the other (see [13]).

To evaluate our SSU fusion method, we have performed two different kinds of assessment: in the first we analyze the correctness of our proposed method with an objective measure, while in the second we analyze the performance through a session of subjective user tests.

In the first part of our evaluation, we have investigated the degree of distortion introduced by our technology. In the same way we defined the distance between SSUs, the distortion of an SSU is defined as the mean of the cross-distances between all the shots that belong to it; we expect that the operation of merging two SSUs into one fused SSU will increase the distortion measure, especially if we limit ourselves to those SSUs that have at least one matching cluster. The results of the analysis are shown in Fig. 3. As it can be observed, all the SSUs resulting from the fusion process have a distortion that is comparable with the distortion of the original movie SSUs , except for some rare cases where it moves away more significantly. Furthermore, it has to be noted that the obtained distortions are much lower than the general distortion, that is the expected value if we build a new SSU choosing its shots at random. This result assures that the content of a fused SSU has at least a plausible video aspect (w.r.t. those SSUs originally present in the movie), since the obtained SSU distortion almost always assumes a plausible value.

The results of the second part of our evaluation, instead, are illustrated in Table 1. We asked to five interviewees to play the authoring role and assess the content generated from the fused SSUs, by watching output clips obtained by performing a random walk through the shots of the fused SSU and evaluating if some kind of meaning could be attached to the resulting scene. Two sets of output clips have been obtained by considering, in addition to pairs of SSUs having the best (lowest) associated distance, also those pairs having the 2nd-best associated distance. The results pertaining to these two sets are shown in the rows of Table 1, which reports the average clip acceptance ratio and the user agreement (the overlap between users acceptance decisions), both expressed in percentage. As expected, it can be observed that the accepted SSUs in the second row are less than those in the first; therefore, confining the analysis only to the nearest SSU in the fusion process is useful, since as SSUs with higher visual distance are fused, the resulting output clips could be more confusing for the interviewee and therefore it is more difficult to give a global meaning to the generated narrative scene. Also, the obtained results in the first row are encouraging and

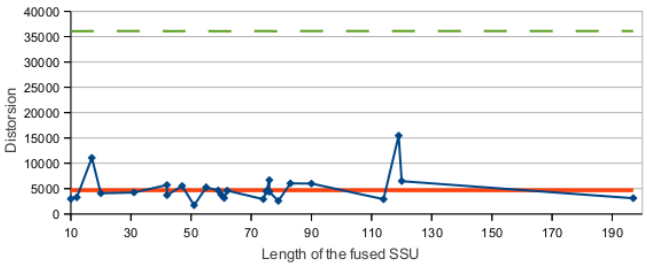


Fig. 3. Trend of the fused SSUs distorsion in function of their length. Here the dotted line represents the global distorsion of the movie (calculated between all its shots), the bold line represents the mean of the distorsions of all the SSUs originally present in the movie and the remaning one is the measured distorsion of the fused SSUs. For clearity, we sorted the obtained values by length of the fused SSUs.

Clips set	*Accept [%]*	*Agreement [%]*
Nearest SSU Case	55	63
2nd-Nearest SSU Case	42	70

Table 1. Users acceptance and agreement aggregated ratios for both output clips sets, expressed in percentages. The users were asked to grade the clips from 1 to 5.

show, by employing the proposed method in the movietelling framework, that fusing Markov chains which share common states is a viable solution to obtain new meaningful expanded models: in fact the results show that among the 33 proposed new scenes, 18 have an accettable meaning attached. This results in 18 new scenes available to the system proposed in [7] that can successfully use this new video material for constructing its new filmic variants of the baseline movie.

Moreover, some examples of the output clips obtained through the fusion method can be found online at [14]. There, both the original SSUs and the fused one can be displayed, along with a suggested narrative meaning of the resulting new scene. The latter also includes subtitles that identify the shots original provenience to better highlight the mixing process. More information could be found directly on the web page.

6. CONCLUSIONS

In this paper we have proposed a general methodology to address the issue of mixing pairs of Markov chains. As MCs are used to model two instances of a given process, in the case that such models share common states it could be useful to obtain an unique model that in some way inherits the structures of the two original Markov chains. Although this is a very general concept and as such it could be applied wherever the above situation applies, we framed the idea into the IRIS NoE movietelling prototype context of filmic story variants generation by recombining the original content of a baseline movie. In this case, since each constructed variant can usefully exploit the SSU (MC) structure that were already present in the movie to have an adequate consistency, we conveniently apply our fusion method to mix SSUs sharing high-level concepts as individuated by semantic tags. The effectiveness of the proposed technique has been demonstrated through experimental results obtained on a set of generated filmic variants.

7. REFERENCES

[1] J. Norris. *Markov Chains*. Cambridge Univ. Press, 1997.

[2] D. Zhang and X. Zhang. Study on forecasting the stock market trend based on stochastic analysis method. In *Int. Journal of Business and Management*, 2009.

[3] A. N. Langville and C. D. Meyer. *Google PageRank and Beyond*. Princeton University Press, 2009.

[4] L. Xie, S. F. Chang, A. Divakaran, and H. Sun. Structure analysis of soccer video with hidden markov models. *Proc. ICASSP*, 2:4096–4099, 2002.

[5] J. Huang, Z. Liu, and Y. Wang. Joint video scene segmentation and classification based on hidden markov model. In *IEEE Trans. on Multimedia*, 2005.

[6] S. Benini, P. Migliorati, and R. Leonardi. Statistical skimming of feature films. In *Int. Journal of Digital Multimedia Broadcasting*, 2010.

[7] A. Piacenza, F. Guerrini, N. Adami, R. Leonardi, J. Teutenberg, J. Porteous, and M. Cavazza. Generating story variants with constrained video recombination. *Proc. ACMMM*, pages 223–232, 2011.

[8] M. Radford. MGM Home Ent. (Europe) Ltd., 2004. The Merchant of Venice (film adaptation).

[9] M. M. Yeung and B.-L. Yeo. Time-constrained clustering for segmentation of video into story units. *Proc. ICPR*, 3:375–380, 1996.

[10] S. Benini, P. Migliorati, and R. Leonardi. Hidden markov models for video skim generation. *Proc. WIAMIS*, 2007.

[11] C. Cotsaces, N. Nikolaidis, and I. Pitas. Video shot detection and condensed representation. *IEEE Signal Processing Magazine*, 23(2):28–37, 2006.

[12] J. Porteous and al. Interactive storytelling via video content recombination. *Proc. ACMMM 2010*, pages 1715–1718, 2010.

[13] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic, 1992.

[14] Online. www.ing.unibs.it/alberto.piacenza/fusion.