# EntScene: Nonparametric Bayesian Temporal Segmentation of Videos Aimed at Entity-Driven Scene Detection

**Adway Mitra**
CSA Department
Indian Institute of Science
Bangalore, India
{adway,chiru}@csa.iisc.ernet.in

**Chiranjib Bhattacharyya**
CSA Department
Indian Institute of Science
Bangalore, India

**Soma Biswas**
EE Department
Indian Institute of Science
Bangalore, India
soma.biswas@ee.iisc.ernet.in

## Abstract

In this paper, we study Bayesian techniques for entity discovery and temporal segmentation of videos. Existing temporal video segmentation techniques are based on low-level features, and are usually suitable for discovering short, homogeneous shots rather than diverse scenes, each of which contains several such shots. We define scenes in terms of semantic entities (eg. persons). This is the first attempt at entity-driven scene discovery in videos, without using meta-data like scripts. The problem is hard because we have no explicit prior information about the entities and the scenes. However such sequential data exhibit *temporal coherence* in multiple ways, and this provides implicit cues. To capture these, we propose a Bayesian generative model- *EntScene*, that represents entities with mixture components and scenes with discrete distributions over these components. The most challenging part of this approach is the inference, as it involves complex interactions of latent variables. To this end, we propose an algorithm based on Dynamic Blocked Gibbs Sampling, that attempts to jointly learn the components and the segmentation, by progressively merging an initial set of short segments. The proposed algorithm compares favourably against suitably designed baselines on several TV-series videos. We extend the method to an unexplored problem: temporal co-segmentation of videos containing same entities.

## 1 Introduction

Naturally occurring sequential data often have an important property- *Temporal Coherence* (TC), i.e. successive datapoints in the sequence are semantically related. For example in a video, successive frames show the same objects, except at a few *changepoints*. Detecting such changepoints, i.e. temporally segmenting the video into semantically coherent subsequences helps in video summarization [Potapov *et al.*, 2014]. Existing approaches to temporal video segmentation [Potapov *et al.*, 2014; Tierney *et al.*, 2014] are based on similarities of low-level visual features of the frames. In this paper, we consider *entity-driven temporal segmentation*, where each temporal segment should be associated with one or more entities, like persons, objects or actions, to help semantic summarization as attempted recently in [Mitra *et al.*, 2014]. Also such temporal segmentation can help users who want to watch only specific parts of a video instead of the entire video. A major challenge is that these entities, or even their number, are unknown and need to be learnt from data. Another challenge is to model the structures which the temporal segments have.

Consider the video of a TV series episode with a reasonably small but unknown number of persons. A TV serial episode is formed of a few temporal segments called *scenes or acts*- where a small subset of the persons are present. Such a video can be represented by the sequence formed by detecting the faces of the persons in all frames, as done in [Mitra *et al.*, 2014]. This is a *semantic video representation*, as it focusses only on the entities of semantic interest (in this case persons' faces). We can define a scene in terms of the persons it contains. Our task is to simultaneously discover the persons and segment the video into the scenes based on these persons. Another task is *Co-segmentation*, or mutually-aided joint segmentation of multiple videos that are known to have temporal segments in common.

Using the approach of [Mitra *et al.*, 2014] it is possible to discover the entities, assign each detection to an entity and segment the sequence based on these assignments. This is equivalent to temporal segmentation of the video into *shots*, where each shot is associated with an entity. But a video is hierarchically organized [Del Fabro and Böszörmenyi, 2013] and each scene is a sequence of several shots. In a TV series episode successive shots within a scene may alternate between the entities (persons) in roughly cyclical patterns. For example during a two-person discourse the camera focusses on one person when she speaks, then on the second person, then back to the first and so on. In Fig 1, shot changes occur after frames 2,3,4,6,8, but scene change occurs after frame 6 only. Temporally segmenting videos into scenes has been studied [Del Fabro and Böszörmenyi, 2013] but no existing approach defines scenes in terms of semantic entities.

In this paper we explore *Bayesian modeling* for the task, which provides an elegant way to model the various temporal structures discussed above. As in [Mitra *et al.*, 2014] we represent the video as a sequence of entity detections (obtained by a suitable detector) and the entities by mixture compo-

nents. Since the number of such components is unknown, we use Bayesian Nonparametrics (BNP) [Teh *et al.*, 2006; Fox *et al.*, 2008]. Due to the TC property, successive datapoints (entity detections) are likely to be generated by the same mixture component. For entity-driven scene discovery, we represent each scene as a sparse distribution over the components (entities). Neither the segment end-points nor the number of segments is known. However, we use an initial over-segmentation of the video which respects the true segmentation, (the initial set of changepoints contains all the true changepoints) and try to improve it.

The main **contribution** of this paper is a Bayesian generative model- *EntScene*, for entities and scenes in a video. To the best of our knowledge, this is the first entity-driven approach to modelling video scenes, as well as the first Nonparametric Bayesian approach to scene discovery. We also propose an inference algorithm (MI-BGS), which jointly learns the entities and segmentation. Inference is based on *Blocked Gibbs Sampling* with dynamic blocking of variables (explored recently in [Venugopal and Gogate, 2013]), and proceeds by starting with an initial over-segmentation and progressively merging segments till convergence. We also consider suitable baselines for inference, such as a split-merge inference approach where previously merged segments may be split again (inspired by Topic Segmentation Model [Du *et al.*, 2013]), and another algorithm where the entities are first discovered (like TCCRF [Mitra *et al.*, 2014]), and then segments are inferred based on them. We evaluate the approach on several TV-series videos of varying lengths, using novel evaluation measures. We also explore entity-driven temporal co-segmentation of similar videos.

## 2 Related Works

Much of the work on temporal video segmentation studied so far [Potapov *et al.*, 2014; Tierney *et al.*, 2014; Del Fabro and Böszörmenyi, 2013] has been about frame similarity based on observed features rather than unknown entities. Discovery of entities was recently attempted by [Mitra *et al.*, 2014] which represents the video as a sequence of *tracklets* [Huang *et al.*, 2008] created from entity detections, as discussed in Section3. It proposes a Bayesian nonparametric approach to cluster these tracklets and discover the entities, but it does not consider temporal segmentation.

The simplest **Bayesian model for sequence segmentation** was the *Product Partition Model*(PPM) [Barry and Hartigan, 1993], which assume that given a partition of the sequence, the data within each partition are IID. This was improved upon by sticky HDP-HMM [Fox *et al.*, 2008], a Markovian model where the assignment of mixture component to each datapoint depends on the component assigned to the previous one. However unlike PPM it does not partition the data into segments and represent each segment with a distribution over the components. LaDP [Mitra *et al.*, 2013] assigns a mixture distribution and a mixture component to each datapoint conditioned on the assignments to the previous datapoint, and thus encourages TC at both levels. But neither sHDP-HMM nor LaDP model the segments explicitly and segmentation is achieved as a by-product of the assignments. In contrast, Topic Segmentation Model [Du *et al.*, 2013] attempts to model segments with mixture distributions. It starts with an initial set of candidate change-points and tries to identify the true ones. However, it uses a fixed number of mixture components, and does not model TC in the assignment of components to the datapoints.

The main challenge for the Bayesian approach lies in the **inference**. Exact inference algorithms have been considered for PPM[Fearnhead, 2006]. However, the complex hierarchical models require approximate inference. sHDP-HMM [Fox *et al.*, 2008] and LaDP [Mitra *et al.*, 2013] perform inference by Gibbs Sampling, where the latent variable assignments to each datapoint are sampled conditioned on the assignments to its neighboring datapoints. For Topic Segmentation Model [Du *et al.*, 2013] a split-merge inference algorithm is considered, where each initial changepoint is provided a binary variable which indicates whether or not it is a true changepoint. This variable is sampled along with the mixture component assignments during inference.

A Bayesian model for **co-segmentation** of sequences is the Beta Process Hidden Markov Model (BP-HMM) [Willsky *et al.*, 2009] that considers mixture components to be shared by sequences. It has been used for modelling human actions in videos. However, it does not model TC or temporal structures like scenes. Spatial Co-segmentation of videos through Bayesian nonparametrics has been studied recently [Chiu and Fritz, 2013], using Distance-dependent Chinese Restaurant Process [Blei and Frazier, 2011] to model spatial coherence.

## 3 Temporal Video Segmentation

Consider the episode of a TV-series, with many persons. We can run a face detector on each frame, and link spatio-temporally close ones to form tracklets [Huang *et al.*, 2008]. We consider tracklets spanning $r$ frames. Normally $5 \leq r \leq 20$, and at $r = 1$ we have individual detections. The detections within each tracklet are visually similar due to temporal coherence. It is possible to represent each detection as a feature vector. We represent each tracklet $i$ by the tuple $(R_i, Y_i)$ where $Y_i$ is the mean feature vector of the associated detections, and $R_i$ is the set of indices of the frames spanned by $i$. Note that there can be several face detections per frame, and hence the $R$-sets of different tracklets can overlap. The tracklets are ordered sequentially using the indices of their starting frames (ties resolved at random), and for each tracklet $i$ we define predecessor $pred(i)$ and successor $succ(i)$. If the temporal gap between any tracklet $i$ and $pred(i)$ is too large, we set $pred(i) = -1$ (similarly for $succ(i)$). Let $\{F_j\}_{j=1}^{M}$ be the set of frames with at least one associated tracklet, arranged in ascending order of frame index.

Next, define *latent variables* $Z_i$ as the index for the mixture component associated with tracklet $i$ and $S_j$ as the index for the mixture distribution associated with frame $F_j$. Temporal coherence property suggests that with high probability, $Z_{pred(i)} = Z_i = Z_{succ(i)}$ hold for all datapoints $i$ for which $pred(i)$ and $succ(i)$ are defined. Temporal coherence holds at frame-level also, as follows:

$$\{Z\}_{j-1} = \{Z\}_j = \{Z\}_{j+1} \quad (1)$$

$$S_{j-1} = S_j = S_{j+1} \quad (2)$$

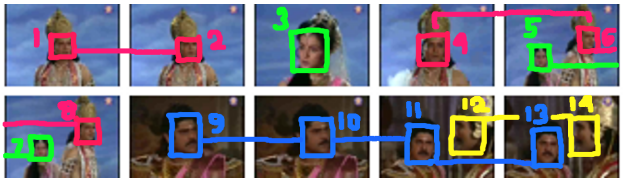

Figure 1: The set-up for entity(person)-driven scene discovery. One colour is used per person. So here, $pred(2) = 1, pred(3) = -1, pred(6) = 4, pred(7) = 5$, $pred(11) = 10$, $pred(14) = 12$ etc; $succ(1) = 2, succ(2) = -1, succ(6) = 8, succ(11) = 13$ etc. Then $Z(1) = Z(2) = Z(4) = Z(6) = Z(8) = 1, Z(3) = Z(5) = Z(7) = 2, Z(9) = Z(10) = Z(11) = Z(13) = 3, Z(12) = Z(14) = 4$; $S(1) = \ldots = S(6) = 1, S(7) = \ldots = S(10) = 2$. Further, the level-1 chagepoints are $\{1, 3, 4, 5, 7, 9\}$, while the level-2 changepoints are $\{1, 7\}$

Here $\{Z\}_j = \{Z_i : F(j) \in R_i\}$, i.e. $\{Z\}_j$ is the set of $Z$-variables corresponding to all tracklets that cover frame $F(j)$. With slight abuse of notation, $\{Z\}_s$ denotes the set of all $Z$-variables associated with all frames satisfying $S_j = s$. We call the frames where the Condition 1 does not hold as *Level-1 changepoints* and the ones where the Condition 2 does not hold as the *Level-2 changepoints*. The *hierarchical segmentation problem* is to find these changepoints. An interval of frames $\{F(j1), \ldots, F(j2)\}$ is a *level-1 segment* (shot) if $\{Z\}_{j1} = \{Z\}_{j1+1} = \ldots = \{Z\}_{j2}$, but $\{Z\}_{j1} \neq \{Z\}_{j1-1}$ and $\{Z\}_{j2} \neq \{Z\}_{j2+1}$. In this case, $j1$ and $j2+1$ are level-1 changepoints. Similarly, an interval of frames $\{F(j3), \ldots, F(j4)\}$ is a *level-2 segment* (scene) if $S_{j3} = \ldots = S_{j4}$, but $S_{j3} \neq S_{j3-1}$ and $S_{j4} \neq S_{j4+1}$. In this case, $j3$ and $j4+1$ are level-2 changepoints. $CP1$ and $CP2$ are *Candidate Frames* like shot changepoints[1] which may be Level-1 or Level-2 changepoints respectively.

**Example:** Temporal video segmentation is illustrated in Figure 1. We show 10 frames from 2 scenes, each having 1 or 2 face detections corresponding to 4 persons. The detections are numbered 1-14, and linked based on spatio-temporal locality, as shown by the coloured lines. Here the tracklets are individual detections, i.e. $R = 1$.

**Challenges**: Segmentation of a video into scenes is difficult, especially if the scene involves multiple persons. This is because all the persons are usually not seen together in any frame, and appear in turns. When a new person appears, it is not known whether it is within the current scene or the beginning of a new scene. If the persons appearing hitherto in the current scene appear again after the new person, then we know that the same scene is continuing. Hence, a single forward pass over the sequence is usually not enough for scene segmentation, and iterative approaches are more effective. Moreover, in videos the same person often appears in different poses, and so several mixture components may be formed for the same person. The pose change of a person within a scene may be interpreted as the appearance of a new person, and perhaps also the start of a new scene. As a result, person-driven temporal segmentation of a video into scenes is difficult, and risks oversegmentation.

[1] http://johmathe.name/shotdetect.html

# 4 *EntScene* Model and Inference

We now come to a generative process for videos. We focus on TV-series videos, and our entities are the persons, whom we represent with their faces. One part of this generative process is modeling the observed data (the tracklets resulting from face detections). We model the semantic entities (i.e. persons) as mixture components $\{\phi_k\}$, and the datapoints (tracklets) are drawn from these components. We represent the face tracklets as vectors $\{Y_i\}$ of pixel intensity values. Tracklet $i$ is associated with person $Z_i$, and according to our model, $Y_i \sim \mathcal{N}(\phi_{Z_i}, \Sigma)$.

## 4.1 Modeling Temporal Structure

The more complex part of the generative process is the modeling of Temporal Coherence, at the levels of scene and track.

**Temporal Coherence at scene level** The frame-specific scene variables $S_j$ can be Markovian [Fox *et al.*, 2008] conditioned on its predecessor $S_{j-1}$. Frame $F(j)$ and its associated tracklets remain in the current scene $S_{j-1}$ with probability $\kappa$, or start a new scene $S_{j-1}+1$ with probability $(1-\kappa)$.

$$S_j \sim \kappa\delta_{S_{j-1}} + (1-\kappa)\delta_{S_{j-1}+1} \tag{3}$$

If $F(j)$ and $F(j-1)$ have a tracklet in common, then $S_j = S_{j-1}$, as scene change cannot happen in the middle of a running tracklet.

**Modeling of a Scene** Each level-2 segment (scene) $s$ has to be modeled as a distribution $G_s$ over mixture components (persons). In case of TV series videos,a person can appear in several scenes. Such sharing of components can be modeled using like Hierarchical Dirichlet Process [Teh *et al.*, 2006], using $H$ as base distribution (Gaussian) and $\{\alpha_s\}$ as segment-specific concentration parameters.

$$\phi_k \sim H\forall k;\ G \sim GEM(\alpha);\ G_s \sim DP(\alpha_s, G)\forall s \tag{4}$$

A sparse modeling can be considered, where each level-2 segment selects a sparse subset of the components using a Beta-Bernoulli process [Griffiths and Ghahramani, 2005; Hughes *et al.*, 2012; Williamson *et al.*, 2010]. Then each segment $s$ has an associated binary vector $B_s$ which indicates which components are active in $s$.

$$\beta_k \sim Beta(1, \beta)\forall k; B_{sk} \sim Ber(\beta_k)\forall s, k \tag{5}$$

**Temporal Coherence at tracklet level** For assigning mixture component $Z_i$ to tracklet $i$, the temporal coherence can be maintained using a Markovian process once again. In this case, $i$ is assigned either the component of its predecessor $pred(i)$ or a component sampled from $G_s$, restricted to the ones active in $s$ ($s$ is the segment containing frames in $R_i$).

$$Z_i \sim \rho\delta_{Z_{pred(i)}} + (1-\rho)(B_s \circ G_s) \tag{6}$$

where $B_s$ is the sparse binary vector. As $G_s$ is discrete (Dirichlet Process-distributed), multiple draws from it may result in sampling a component several times in the same segment $s$. This is desirable in TV series videos, since a particular person is likely to appear repeatedly in a scene.Based on all these, the entity-driven generative process for TV-series videos is given in Algorithm 1.

**Algorithm 1** *EntScene* Generative Model

$\phi_k \sim \mathcal{N}(\mu, \Sigma_0)$, $\beta_k \sim Beta(1, \beta)$ for $k = 1, 2, \ldots, \infty$
$G \sim GEM(\alpha)$
**for** $j = 1$ to $M$ **do**
  $S_j \sim \kappa\delta_{S_{j-1}} + (1-\kappa)\delta_{S_{j-1}+1}$
  **if** $j = 1$ or $S_j \neq S_{j-1}$ **then**
    $B_{sk} \sim Ber(\beta_k)\ \forall k\ (s = S_j)$
    $G_s \sim DP(\alpha_s, G)$
  **end if**
**end for**
**for** $i = 1 : N$ **do**
  if $pred(i) = -1$ set $\rho = 0$
  $Z_i \sim \rho\delta_{Z_{pred(i)}} + (1-\rho)(B_{S_j} \circ G_{S_j})\ (j \in R_i)$
  $Y_i \sim \mathcal{N}(\phi_{Z_i}, \Sigma)$
**end for**

## 4.2 Merge Inference by Blocked Gibbs Sampling

As mentioned earlier, *hierarchical segmentation* is to discover the frames where Equation 1 or Equation 2 is violated. For this purpose, we need to infer the $\{S_j\}$ and $\{Z_i\}$ variables. The complete likelihood function in terms of the random variable discussed above can be written as

$$p(Y, Z, S, B, \Phi, \beta, G, G_0) \propto \prod_{k=1} p(\beta_k)p(\phi_k) \times p(G)$$
$$\times \prod_{j=2}^{M} p(S_j|S_{j-1}) \times \prod_s p(G_s|G) \times \prod_{s,k} p(B_{sk}|\beta_k)$$
$$\times \prod_{i=1}^{N} p(Z_i|Z_{pred(i)}, S_{F(i)}, \{B_s\}, \{G_s\})p(Y_i|Z_i, \Phi) \quad (7)$$

We can collapse some of these variables, like $\{\beta_k\}$, $\{\Phi\}$,$\{G_s\}$ and $G$, in which case the $B$ variables can be handled using the Indian Buffet Process, and the $Z$ variables using the Chinese Restaurant Process. In that case, the likelihood function can be written as:

$$p(Y, Z, S, B) \propto \prod_{j=2}^{M} p(S_j|S_{j-1}) \times \prod_s p(B_s|B_1, \ldots, B_{s-1})$$
$$\times \prod_{i=1}^{N} p(Z_i|Z_1, \ldots, Z_{pred(i)}, \{B\}, \{S\})p(Y_i|Z_i, Y_1, \ldots, Y_{i-1}) \quad (8)$$

For inference we use Blocked Gibbs Sampling as several variables are usually strongly coupled, and must be sampled together. We form blocks *dynamically* using the $S$ variables. Clearly the scene boundaries occur at frames where the $S$-variable changes, i.e. where $S_j \neq S_{j-1}$, and each value $s$ of $S$ defines a segment. A block $BL(s)$ is formed as $\{B_{s-1}, B_{s+1}, B_s, \{Z\}_s, \{S\}_s\}$. We first infer $\{S\}_s$ using Eq 3 and the marginal likelihood of the data $\{Y\}_s$. We try to merge segment $s$ with either segment $(s-1)$ or segment $(s+1)$ (or leave it alone), so the state-space of $\{S\}_s$ is only $\{s-1, s, s+1\}$. After sampling $\{S\}_s$, we sample $\{B\}$ and $\{Z\}$ variables using Eq 8. After each iteration, the blocks are re-defined according to the new assignment of $\{S\}$ variables. Since the aim is always to merge each segment with its neighbors, the number of segments should reduce till convergence. We can use $CP1$ and $CP2$ to initialize $\{Z\}$ and $\{S\}$ respectively for Gibbs Sampling, thus getting an initial segmentation. We know that if frames $a$ and $b$ are two successive points in $CP2$, then obviously there is no changepoint between them, i.e. $a < j < j' < b \Rightarrow S_a = S_j = S_{j'}$. This considerably reduces the search space for segments and allows us to keep merging the segments progressively (till convergence). The process is explained in Algorithm 2.

The various parts of Eq 8 can be computed using the inference equations of Indian Buffet Process [Griffiths and Ghahramani, 2005] for $\{B\}_s$ and TC-CRF [Mitra *et al.*, 2014] for $\{Z\}_s$. The convolution of $G_s$ with the sparse binary vector $B_s$ in Eq 6 poses a major challenge as it cannot be collapsed by integration, as noted in [Williamson *et al.*, 2010]. We suggest an approximate PPF (like the TC-CRF PPF) for Eq 6 for easy inference. With every datapoint $i$ we can consider an auxilliary binary variable $C_i$ which takes the value 0 with probability $\rho$, and $C_i = 0 \Rightarrow Z_i = Z_{pred(i)}$. In segment $s$, for a datapoint $i$ where $C_i = 1$, a component $\phi_k$ may be sampled with $p(B_{sk} = 1, Z_i = k) \propto n^s_k$, which is the number of times $\phi_k$ has been sampled at other points $i'$ satisfying $C_{i'} = 1$ within the same segment. If $\phi_k$ has never been sampled within the segment but has been sampled in other segments, $p(B_{sk} = 1, Z_i = k) \propto \alpha n_k$, where $n_k$ is the number of segments where $\phi_k$ has been sampled (Corresponding to $p(B_{sk}) = 1$ according to IBP). Finally, a completely new component may be sampled with probability proportional to $\alpha_0$. Note that $p(B_{sk} = 0, Z_i = k) = 0 \forall k$.

**Algorithm 2** Merge Inference Algorithm by Blocked Gibbs Sampling (MI-BGS)

Initialize segments $S$ using CP2; Initialize $B$,$Z$;
Estimate components $\hat{\phi} \leftarrow E(\phi|B, Z, S, Y)$;
**while** Number of segments not converged **do**
  **for** each segment $s$ **do**
    Sample $\{S\}_s \in \{s-1, s, s+1\} \propto p(\{Y\}_s|Z, B, S, \hat{\phi})$
    Sample $(B_s, \{Z\}_s) \propto p(B_s, \{Z\}_s|B_{-s}, \{Z\}_{-s}, S, Y, \hat{\phi})$
  **end for**
  Re-number the $S$-variables, update components $\hat{\phi} \leftarrow E(\phi|Z, B, S, Y)$;
**end while**

## 4.3 Alternative Inference Algorithms

Having described our main inference algorithm, we discuss two alternatives, which can serve as baselines.

**Split-Merge Inference (SpMI)** By the MI-BGS algorithm the number of segments keeps decreasing and then converges. This property is desireable as it helps in quick convergence. But two segments can never split after they are merged once, which may come as a disadvantage in case of a wrong merge. The Topic Segmentation Model (TSM) [Du *et al.*, 2013] allows split-merge inference by a bernoulli random variable $U_s$ with each initial segment $s$ from $CP2$, which indicate whether or not a new segment should start from $s$, i.e. if $\{Z\}_s$ and $\{Z\}_{s-1}$ should be modelled with the same distribution. To change $U_s$ from 0 to 1 is to split segments $s$ and $(s-1)$, and the reverse change is to merge them. The process is explained in Algorithm 3. Note that $\{U_s\}$-variables are a reparametrization of $\{S\}$, so that the new joint distributions can be found easily from Equation 8.

**Sweep-Merge Inference (SMI)** Both MI-BGS and SpMI aim to *jointly discover the entities and segment the sequence*. A simpler alternative (baseline) can be to perform the entity discovery first, disregarding the $S$-variables, as done in [Mitra *et al.*, 2014], and then separately infer the segmentation $\{S\}$, conditioned on the already-assigned $\{B\}$ and $\{Z\}$. To

**Algorithm 3** Split-Merge Inference Algorithm by Blocked Gibbs Sampling (SpMI-BGS)

Initialize segments $S$ using CP2; Initialize $B$,$Z$;
Estimate components $\hat{\phi} \leftarrow E(\phi|B, Z, U, Y)$;
**while** Number of segments not converged **do**
  **for** each segment $s$ **do**
    Sample $(U_s, B_s, \{Z\}_s) \propto p(U_s, B_s, \{Z\}_s|U_{-s}, B_{-s}, \{Z\}_{-s}, Y, \hat{\phi})$
  **end for**
  Update components $\hat{\phi} \leftarrow E(\phi|B, Z, U, Y)$;
**end while**

infer $\{S\}$, we make a single sweep from left to right, attempting to *merge* the initial segments defined by $CP2$. For every initial segment $s$, we propose to merge it into the currently running level-2 segment $c$, using a common binary vector $B^{merge}$ for all datapoints in the proposed merged segment and component assignments $\{Z^{merge}\}$ to the datapoints in $s$. We may accept or reject the merger proposal based on how well $(B^{merge}, \{Z^{merge}\})$ can model the data $Y_{c\cup s}$ in the merged segments $(c, s)$, compared to modeling them as separate segments. The merger probability is enhanced by temporal coherence (Eq 3). If we accept it, we will merge slice $s$ into level-2 segment $c$, and set $S_j = c$ for all frames $j$ occurring within $s$. If we reject it, we start a new level-2 segment $(c+1)$, and set $S_j = c+1$ for all frames $j$ within $s$. The process is explained in Algorithm 4.

**Algorithm 4** Sweep-Merge Inference Algorithm (SMI)

Initialize segments $S$ using CP2;
**for** all initial segments $s$ **do**
  $(B_s, \{Z\}_s) \sim p(B_s, \{Z\}_s|B_{-s}, \{Z\}_{-s}, Y)$
**end for**
Estimate components $\hat{\phi} \leftarrow E(\phi|Z, B, S, Y)$;
Set *current segment* $c = 1, \{S\}_1 = 1$;
**for** each initial segment $s$ **do**
  Sample $(B_c^{merge}, \{Z^{merge}\}_s) \propto p(B_c^{merge}, \{Z^{merge}\}_s|Y, \hat{\phi}, \{Z\}_c)$
  Accept/reject the merger based on data likelihood
  **if** merger accepted **then**
    $\{Z\}_s = \{Z^{merge}\}_s, \{S\}_s = c, B_c = B_c^{merge}$;
  **else**
    $\{S\}_s = c+1$;Set $(c+1)$ as current segment;
  **end if**
**end for**

# 5 Experiments on Temporal Segmentation

## 5.1 Datasets and Preprocessing

We carried out extensive experiments on TV series episode videos of various lengths. We collected three episodes of The Big Bang Theory (Season 1). Each episode is 20-22 minutes long, and has 7-8 persons (occurring in at least 100 frames). We also consider 6 episodes, each 40-45 minutes long, of the famous Indian TV series- the *Mahabharata*. On each video, face detection is performed by a standard face detector [Viola and Jones, 2001] and these detections are linked based on spatio-temporal locality to form tracklets of size $r = 10$. Each tracklet corresponds to a datapoint $i$, which is represented as a 900-dimensional vector $Y_i$ of pixel values, which is the mean vector of the associated detections. Their frame indices $R_i$ are used to order them and define $pred(i)$ and $succ(i)$. Frames where a new tracklet starts but are not spanned by any previously running tracklets, comprise our $CP1$. Also, the video can be segmented into shots based on frame differences [2], and the frames on these shot boundaries provide $CP2$. The task is to segment the video into scenes. As already discussed, each person is represented by a mixture component, and each scene by a mixture distribution. However, there is a lot of variation in pose and appearance of the detected faces, throughout the video, and hence *often several mixture components are formed per person*. The hyperparameters like $\alpha$ and $\beta$ provide some control over the number of components learnt. After tuning them on one episode, we found an optimal setting, where we were able to cover $80 - 85\%$ of the tracklets with 80-90 components. $\kappa$, $\rho$ etc are also fixed by tuning on one episode.

## 5.2 Performance Measures

A gold-standard segmentation is created manually at the level of scenes (level-2), and we evaluated the inferred segmentation against this. But gold-standard segmentation is difficult to annotate in level-1, as the videos are long, and there are too many level-1 segments. So at this level our evaluation is about the quality of the mixture components learnt.

**Evaluation of mixture components** Among the learnt components, we select only those components that have at least 10 assigned tracklets overall and at least 2 detections in any of the learnt level-2 segments, and reject the rest. This is because we are interested only in persons that have reasonable screen presence. We attribute a selected mixture component to person $A$ if $70\%$ of the tracklets assigned to that component belong to person $A$. This is because, we observe that if a component's associated tracklets are at least $70\%$ pure then the corresponding mean vector $\phi_k$ resembles the person's face well enough for identification. For large components (200 or more associated tracklets), we observe that $60\%$ purity is enough for identifiability. We measure as *Cluster Purity (CP)*, the fraction of the selected components which can be assigned to a person. We also measure as *Person Coverage (PC)*, what fraction of the persons with at least 10 tracklets, have been represented by at least one selected component. On these measures we compare our inference algorithms with *sticky HDP-HMM [Fox* et al*., 2008]*: the existing BNP model best suited to learning of mixture components and segmentation.

**Evaluation of Segmentation into Scenes** We evaluate the *number of level-2 segments formed ($NS2$)*, and the *sequence segmentation error measure* $P_k$. $P_k$ is the probability that two tokens, $k$ positions apart, are inferred to be in the same segment when they are actually in different segments in the gold standard, and vice versa. This is measured as $S2$, averaged over three values of $k$ (maximum, minimum and average scene lengths). A third measure is *segment purity* $(SP2)$, which is the fraction of the discovered segments which lie

[2] http://johmathe.name/shotdetect.html

| Video | SMI | | MI-BGS | | SpMI | | sHDP-HMM | |
|---|---|---|---|---|---|---|---|---|
| | CP | PC | CP | PC | CP | PC | CP | PC |
| BBTe1 | **0.84** | **6** | 0.78 | 5 | 0.80 | **6** | **0.84** | 5 |
| BBTe3 | 0.91 | 8 | 0.94 | 8 | **0.96** | **10** | 0.76 | 6 |
| BBTe4 | 0.89 | 6 | **0.91** | **8** | 0.90 | **8** | 0.83 | **8** |
| Maha22 | **0.96** | 12 | 0.89 | **14** | 0.94 | 13 | 0.86 | **14** |
| Maha64 | **0.94** | **14** | 0.92 | 13 | 0.91 | 12 | 0.91 | **14** |
| Maha65 | 0.88 | 16 | 0.83 | 15 | **0.90** | **18** | **0.90** | 17 |
| Maha66 | 0.91 | 14 | 0.81 | **15** | 0.89 | **15** | **0.95** | 13 |
| Maha81 | **0.86** | **22** | **0.86** | 21 | 0.85 | 20 | 0.84 | 20 |
| Maha82 | **0.93** | 19 | 0.89 | 19 | 0.81 | **20** | 0.86 | **20** |

Table 1: Learning Mixture Components (persons) by SMI, MI-BGS, SpMI and sHDP-HMM

| Video | SMI | | MI-BGS | | SpMI | |
|---|---|---|---|---|---|---|
| | CP-RC2 | CP-PR2 | CP-RC2 | CP-PR2 | CP-RC2 | CP-PR2 |
| BBTe1 | **0.78** | 0.26 | 0.78 | **0.30** | 0.33 | 0.22 |
| BBTe3 | 0.77 | 0.24 | **0.85** | 0.23 | 0.85 | **0.30** |
| BBTe4 | 0.75 | **0.32** | **0.83** | 0.26 | 0.75 | 0.24 |
| Maha22 | 0.71 | 0.21 | **0.76** | **0.24** | 0.53 | 0.20 |
| Maha64 | **0.88** | 0.16 | 0.82 | 0.17 | 0.71 | **0.23** |
| Maha65 | 0.78 | 0.20 | **0.87** | **0.27** | 0.74 | 0.23 |
| Maha66 | 0.80 | 0.13 | **0.87** | 0.16 | 0.47 | **0.19** |
| Maha81 | 0.55 | 0.19 | **0.80** | **0.22** | 0.75 | 0.16 |
| Maha82 | 0.32 | 0.15 | **0.72** | **0.38** | 0.48 | 0.26 |

Table 2: Recall and Precision of segment boundaries, using alignment threshold to be 20% of the average scene length

entirely within a scene (i.e. a single gold standard segment).

We can look upon segmentation as a *retrieval problem*, and define the *Precision and Recall of level-2 changepoints (CP-RC2 and CP-PR2)*. Let $j$ be the starting frame index of an inferred segment $s$, i.e. $S_{j-1} \neq S_j$. Then, if there exists $(j0, s0)$ such that $j0$ is the starting frame index of a gold-standard segment $s0$ satisfying $|F(j) - F(j0)| < threshold$ then inferred changepoint $(j, s)$ is *aligned* to gold standard changepoint $(j0, s0)$. The formal definitions are:

$$\textbf{Precision} = \frac{\textbf{\#inferred segment boundaries aligned to a true segment boundary}}{\textbf{\#inferred segment boundaries}}$$

$$\textbf{Recall} = \frac{\textbf{\#true segment boundaries aligned to an inferred segment boundary}}{\textbf{\#true segment boundaries}}$$

The data, code and illustrations of the measures can be found at `http://clweb.csa.iisc.ernet.in/adway`

### 5.3 Results

The component evaluation results are shown in Table 1, and the segmentation results in Tables 2,3,4. In terms of component evaluation, all the methods (including sHDP-HMM) are comparable. Averaged across all the videos, SMI leads in terms of Cluster Purity, while sHDP-HMM is the worst. In terms of Person Coverage, all methods are almost at par when averaged across the videos. At level-2 (i.e. scenes), we see that MI-BGS clearly performs better than SMI and SpMI on precision and recall of segment boundaries (CP-PR2, CP-RC2) and also fares best on the segmentation error (S2). However, SMI is found to be better in terms of segment purity (SP2), which is understandable since it produces a large number (NS2) of pure but small segments. On the other hand, SpMI produces a small number of segments, but they are often inaccurate, resulting in its poor performance in terms of all the measures. This happens because many adjacent initial segments keep splitting and merged, resulting in failure to choose changepoints. The results show that jointly

| Video | SMI | | | MI-BGS | | | SpMI | | |
|---|---|---|---|---|---|---|---|---|---|
| | S2 | NS2 | SP2 | S2 | NS2 | SP2 | S2 | NS2 | SP2 |
| BBTe1 | 0.14 | 51 | **0.77** | **0.09** | 44 | 0.67 | 0.19 | **25** | 0.61 |
| BBTe3 | 0.10 | 40 | 0.74 | **0.08** | 46 | **0.88** | 0.10 | **30** | 0.68 |
| BBTe4 | 0.11 | 26 | 0.71 | 0.12 | 37 | 0.79 | 0.13 | 35 | 0.81 |
| Maha22 | 0.16 | **41** | 0.82 | **0.12** | 53 | **0.84** | 0.15 | 73 | 0.74 |
| Maha64 | 0.19 | 94 | **0.91** | 0.19 | 81 | 0.89 | **0.18** | **50** | 0.77 |
| Maha65 | 0.18 | 87 | **0.82** | **0.16** | **71** | 0.71 | 0.19 | 72 | **0.82** |
| Maha66 | **0.12** | 87 | 0.82 | 0.20 | 79 | **0.90** | 0.19 | **35** | 0.78 |
| Maha81 | 0.23 | **56** | **0.88** | **0.15** | 68 | 0.78 | 0.20 | 89 | 0.82 |
| Maha82 | 0.15 | 50 | **0.77** | **0.07** | **46** | 0.71 | 0.19 | 69 | 0.68 |

Table 3: Segmentation error (S2), number of segments formed (NS2) and segment purity (SP2) at level 2

| Video | SMI | | MI-BGS | | SpMI | |
|---|---|---|---|---|---|---|
| | CP-RC2 | CP-PR2 | CP-RC2 | CP-PR2 | CP-RC2 | CP-PR2 |
| BBTe1 | 0.61 | 0.21 | **0.72** | **0.28** | 0.22 | 0.15 |
| BBTe3 | 0.23 | 0.07 | **0.77** | **0.21** | 0.54 | 0.19 |
| BBTe4 | 0.58 | **0.25** | **0.67** | 0.21 | 0.50 | 0.16 |
| Maha22 | 0.29 | 0.09 | **0.65** | **0.20** | 0.24 | 0.09 |
| Maha64 | 0.59 | 0.10 | **0.76** | **0.16** | 0.41 | 0.13 |
| Maha65 | 0.35 | 0.09 | **0.57** | **0.18** | 0.48 | 0.15 |
| Maha66 | 0.47 | 0.08 | **0.53** | 0.10 | 0.27 | **0.11** |
| Maha81 | 0.35 | 0.12 | **0.55** | **0.15** | 0.50 | 0.11 |
| Maha82 | 0.24 | 0.12 | **0.28** | **0.15** | 0.16 | 0.09 |

Table 4: Recall and Precision of segment boundaries, using alignment threshold to be 200 frames (about 8 seconds)

discovering the entities and the segmentation (MI-BGS) is better than doing them separately (SMI) in terms of segmentation, and in terms of entity discovery they are comparable.

In general the number of segments formed (NS2) is quite high compared to the actual number of scenes, and this affects the precision values for all the methods. This is because of the challenges of scene discovery mentioned in Section 3.

## 6 EntScene for Temporal Co-segmentation

*EntScene* can be extended to multiple videos, which share the same persons (with similar facial appearances). This may be done by allowing the videos to share the mixture components $\{\phi_k\}$, though the weights may differ. In that case, the inference process (SMI, SpMI or MI-BGS) can consider all the initial segments induced by $CP2$ from all the sequences together, and estimate the shared components while initializing $\{B\}$ and $\{Z\}$ variables accordingly. Using shared mixture components allow us to easily find out common persons and temporal segments from the videos. If they are modeled separately, discovery of common persons and segments require matching the sets of mixture components from the different videos.

For this we collect a set of videos corresponding to 3 episodes of the TV series *The Big Bang Theory*. For each episode, we have a main video (full episode) and a short video showing snippets. Every such pair of videos contain the same persons in same facial appearances, and hence fits our case. Our aim is to temporally segment both videos, and find the correspondences between the two sets of temporal segments.

We first create initial segmentations of all the videos using their respective shot boundaries ($CP2$). Next, for each pair of videos from same episodes we learn the mixture components together, and use these common components to *identify similar segments (that contain the same persons) across the pairs*. The binary vector $B_s$ learnt for every segment $s$ is

| Method | BBTe1 | BBTe3 | BBTe4 |
|---|---|---|---|
| Co-Modeling | **0.72** | 0.76 | **0.67** |
| Individual | 0.58 | **0.77** | 0.64 |

Table 5: Segment Matching precision for Co-Modeling and separate modeling of video pairs, using SMI. For MI-BGS and SpMI also, the same trend is seen

used for this purpose. We say that a segment $s_i^a$ from video $a$ and another segment $s_j^b$ from video $b$ are similar based on the Hamming distance of the corresponding $B$-vectors. Every pair of matched segments can then be classified as *good* or *bad* according to a gold-standard matching of such segments, and the *Matching Precision* (fraction of matches that are *good*) can be measured. As baseline, we repeat this for the case where the two videos in a pair are modeled individually, and then the two sets of learnt mixture components are matched based on $\ell_2$-distance of their mean vectors. The results are shown in Table 5, which show that co-modeling performs clearly better than individual modeling in this case.

## 7 Conclusion

In this paper we described *EntScene*: a generative model for entities and scenes in a video, and proposed inference algorithms for discovery of entities and scenes by hierarchical temporal segmentation. This is the first attempt at entity-driven scene modelling and temporal segmentation in videos. We also proposed alternative inference algorithms, and considered the novel task of entity-driven temporal cosegmentation of videos. In our experiments we used only one type of entities (persons), but our method should work for any type of entity such that every instance of it can be modelled with a single vector. Also, the proposed inference algorithms may be useful for other kinds of sequential data, apart from videos.

**Acknowledgements** This research is partially supported by grants from Department of Science and Technology (Government of India) and Infosys.

## References

[Barry and Hartigan, 1993] D Barry and J A Hartigan. A bayesian analysis for change point problems. *Journal of the American Statistical Association*, 88(421):309–319, 1993.

[Blei and Frazier, 2011] D M Blei and P I Frazier. Distance dependent chinese restaurant processes. *The Journal of Machine Learning Research*, 12:2461–2488, 2011.

[Chiu and Fritz, 2013] W-C Chiu and M Fritz. Multi-class video co-segmentation with a generative multi-video model. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[Del Fabro and Böszörmenyi, 2013] M. Del Fabro and L. Böszörmenyi. State-of-the-art and future challenges in video scene detection: a survey. *Multimedia systems*, 19(5):427–454, 2013.

[Du *et al.*, 2013] L Du, W L Buntine, and M Johnson. Topic segmentation with a structured topic model. In *HLT-NAACL*, pages 190–200, 2013.

[Fearnhead, 2006] P Fearnhead. Exact and efficient bayesian inference for multiple changepoint problems. *Statistics and Computing*, 16(2):203–213, 2006.

[Fox *et al.*, 2008] E B Fox, E B Sudderth, M I Jordan, and A S Willsky. An hdp-hmm for systems with state persistence. In *International Conference on Machine Learning (ICML)*, pages 312–319, 2008.

[Griffiths and Ghahramani, 2005] T Griffiths and Z Ghahramani. Infinite latent feature models and the indian buffet process. In *Advances in Neural Information Processing Systems (NIPS)*, 2005.

[Huang *et al.*, 2008] C. Huang, B. Wu, and R. Nevatia. Robust object tracking by hierarchical association of detection responses. In *Computer Vision–ECCV 2008*. 2008.

[Hughes *et al.*, 2012] M C Hughes, E B Sudderth, and E B Fox. Effective split-merge monte carlo methods for nonparametric models of sequential data. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1295–1303, 2012.

[Mitra *et al.*, 2013] A Mitra, BN Ranganath, and I Bhattacharya. A layered dirichlet process for hierarchical segmentation of sequential grouped data. In *European Conference on Machine Learning and Knowledge Discovery in Databases (ECML)*, pages 465–482. 2013.

[Mitra *et al.*, 2014] A Mitra, S Biswas, and C Bhattacharyya. Temporally coherent chinese restaurant process for discovery of persons and corresponding tracklets from user-generated videos. *arXiv preprint arXiv:1409.6080*, 2014.

[Potapov *et al.*, 2014] D Potapov, M Douze, Z Harchaoui, and C Schmid. Category-specific video summarization. In *Computer Vision–ECCV 2014*, pages 540–555. 2014.

[Teh *et al.*, 2006] Y W Teh, M I Jordan, M J Beal, and D M Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476), 2006.

[Tierney *et al.*, 2014] S Tierney, J Gao, and Y Guo. Subspace clustering for sequential data. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1019–1026, 2014.

[Venugopal and Gogate, 2013] D Venugopal and V Gogate. Dynamic blocking and collapsing for gibbs sampling. In *International Conference on Uncertainty and Artificial Intelligence (UAI)*, 2013.

[Viola and Jones, 2001] P Viola and M Jones. Rapid object detection using a boosted cascade of simple features. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I–511. IEEE, 2001.

[Williamson *et al.*, 2010] S Williamson, C Wang, K Heller, and D Blei. The ibp compound dirichlet process and its application to focused topic modeling. In *International Conference on Machine Learning (ICML)*, 2010.

[Willsky *et al.*, 2009] A S Willsky, E B Sudderth, M I Jordan, and E B Fox. Sharing features among dynamical systems with beta processes. In *Advances in Neural Information Processing Systems (NIPS)*, 2009.