Bayesian Modeling of Temporal Coherence in Videos for Entity Discovery and Summarization

Adway Mitra, Soma Biswas, and Chiranjib Bhattacharyya

Abstract—A video is understood by users in terms of entities present in it. *Entity Discovery* is the task of building appearance model for each entity (e.g., a person), and finding all its occurrences in the video. We represent a video as a sequence of tracklets, each spanning 10-20 frames, and associated with one entity. We pose Entity Discovery as *tracklet clustering*, and approach it by leveraging *Temporal Coherence* (TC): the property that temporally neighboring tracklets are likely to be associated with the same entity. Our major contributions are the first Bayesian nonparametric models for TC at tracklet-level. We extend Chinese Restaurant Process (CRP) to TC-CRP, and further to Temporally Coherent Chinese Restaurant Franchise (TC-CRF) to jointly model entities and temporal segments using mixture components and sparse distributions. For discovering persons in TV serial videos without meta-data like scripts, these methods show considerable improvement over state-of-the-art approaches to tracklet clustering in terms of clustering accuracy, cluster purity and entity coverage. The proposed methods can perform online tracklet clustering on streaming videos unlike existing approaches, and can automatically reject false tracklets. Finally we discuss *entity-driven video summarization*- where temporal segments of the video are selected based on the discovered entities, to create a semantically meaningful summary.

Index Terms—Bayesian nonparametrics, Chinese restaurant process, temporal coherence, temporal segmentation, tracklet clustering, entity discovery, entity-driven video summarization

1 Introduction

Online video repositories like Youtube, Dailymotion etc have been experiencing an explosion of user-generated videos. Such videos are often shot/recorded from the television by users, and uploaded onto these sites. They have very little metadata like dialogue scripts, or a textual summary/representation of the content. When an user searches these repositories by keywords, (s)he is suggested hundreds of videos, out of which (s)he may choose a small number for viewing. This has given rise to the topic of *Video Summarization* [41], which aims to provide the user a short but comprehensive *summary* of the video. However, the current state-of-the-art mostly provides a few keyframes as summary, which may not have much semantic significance. The high-level semantic information of videos that is most important to users is carried by *entities*- such as persons or other objects. With the recent progress in object detection [29], [35] in single images and videos, it is now possible to have a high-level representation of videos in terms of such entities. One effective way of summarization is to have a list of entities that appear frequently in a video. Further, an user may want to watch only a part of a video, for example wherever a particular person (or set of persons) appears, which motivate the tasks of Entity Discovery and Entity-driven Summarization [42] of videos.

The problem of *automated discovery of persons from videos along with all their occurrences* has attracted a lot of interest [36], [37], [38] in video analytics. Existing attempts try to leverage meta-data such as scripts [37], [38] and hence do not apply to videos available on the wild, such as TV-Series episodes uploaded by viewers on Youtube (which have no such meta-data). In this paper, we consider the completely unsupervised version- where we need to learn an appearance model for each entity (say a person) and also find all its occurences. Our goal is to design algorithms which can work on long videos, and for any type of entity. We pose it as *tracklet clustering*, as done in [25]. Tracklets [28] are formed by detections of an entity from a short contiguous sequence of 10-20 video frames. They have complex spatio-temporal properties. Given a video in the wild it is unlikely that the number of entities will be known, so the method should automatically adapt to unknown number of entities. To this end we advocate a *Bayesian non-parametric* clustering approach to Tracklet clustering and study its effectiveness in automated discovery of entities with all their occurrences in long videos. The main challenges are in modeling the spatio-temporal properties.

To the best of our knowledge the problem of entity discovery from videos "on the wild" has not been studied either in Machine Learning or in Computer Vision community. The existing methods are not appropriate either because they require number of clusters to be specified ([13], [24], [25]) or because they need additional information like scripts or annotated training videos ([37], [38]). Also, it must be clearly understood that the aim is different from *tracking*- where tracklets are linked on a short range, i.e.,

- *A. Mitra and C. Bhattacharyya are with the Department of Computer Science and Automation, Indian Institute of Science, Bangalore 560012, Karnataka, India. E-mail: {adway.cse, chiranjib.bhattacharyya}@gmail.com.*
- *S. Biswas is with the Electrical Engineering, Indian Institute of Science, Bangalore, Karnataka, India. E-mail: soma.biswas@ee.iisc.ernet.in.*

Manuscript received 29 Aug. 2015; revised 23 Mar. 2016; accepted 19 Apr. 2016. Date of publication 20 Apr. 2016; date of current version 13 Feb. 2017.
Recommended for acceptance by J. J Corso.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2016.2557785

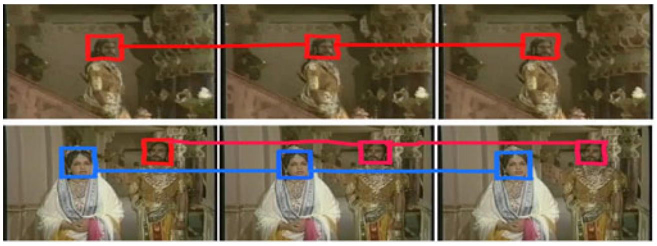

Fig. 1. Top: a window consisting of frames 20000,20001,20002, Bottom: another window- with frames 21000,21001,21002. The detections are linked on spatio-temporal basis to form tracklets. One person (marked with red) occurs in both windows, the other character (marked with blue) occurs only in the second. The two red tracklets should be associated though they are from non-contiguous windows.

over adjacent frames. But here we aim to link tracklets on a long range also, i.e., they need not be from adjacent frames.

To explain the spatio-temporal properties we introduce some definitions. A *track* is formed by detecting entities (like people's faces) in each video frame, and associating detections across a contiguous sequence of frames (typically a few hundreds in a TV series) based on *appearance* and *spatio-temporal* locality. Each track corresponds to a particular entity, like a person in a TV series. Forming long tracks is often difficult, especially if there are multiple detections per frame. This can be solved hierarchically, by associating the detections in a short window of frames (typically 10-20) to form *tracklets* [28] and then linking the tracklets from successive windows to form tracks. The *short-range association of tracklets* to form tracks is known as *tracking*. But in a TV series video, the same person may appear in different (non-contiguous) parts of the video, and so we need to associate tracklets on a *long-range* basis also (see Fig. 1). Moreover the task is complicated by lots of *false detections* which act as spoilers. Finally, the task becomes more difficult on streaming videos, where only one pass is possible over the sequence.

A major cue for this task comes from a very fundamental property of videos: *Temporal Coherence* (TC). This property manifests itself at detection-level as well as tracklet-level; at feature-level as well as at semantic-level. At detection-level this property implies that the visual features of the detections (e.g., appearance of an entity) are almost unchanged across a tracklet (See Fig. 2). At tracklet-level it implies that *spatio-temporally close (but non-overlapping) tracklets are likely to belong to the same entity* (Fig. 3). Additionally, *overlapping tracklets (that span the same frames), cannot belong to the same entity*. A tracklet can be easily represented as all the associated detections are very similar (due to detection-level TC). Such representation is not easy for a long track where the appearances of the detections may gradually change.

Contribution Broadly, this paper has two major contributions: it presents the first Bayesian nonparametric models for TC in videos, and also the first entity-driven approach to video modeling. To these ends, we explore tracklet clustering, an active area of research in Computer Vision, and advocate a Bayesian non-parametric(BNP) approach for it. We apply it to an important open problem: discovering entities (like persons) and all their occurrences from videos of movies and TV-series, in absence of any meta-data, e.g., scripts. We use a simple and generic representation leading to representing a video by a matrix, whose columns represent individual tracklets (unlike other works which represent an individual detection by a matrix column, and then try to encode the tracklet membership information). We propose Temporally Coherent-Chinese Restaurant process (TC-CRP), a BNP prior for enforcing coherence on the tracklets. Our method yields a superior clustering of tracklets over several baselines, on short benchmark videos as well as longer videos downloaded from Youtube. As an advantage it does not need the number of clusters in advance. It is also able to automatically filter out false detections, and perform the same task on *streaming videos*, which are impossible for existing methods of tracklet clustering. We extend TC-CRP to the Temporally Coherent Chinese Restaurant Franchise (TC-CRF), that jointly models short video segments and further improves the results. We show that the proposed methods can be applied to entity-driven video summarization, by selecting a few representative segments of the video in terms of the discovered entities.

Fig. 2. TC at Detection level: Detections in successive frames (linked to form a tracklet) are almost identical in appearance, i.e., have nearly identical visual features.

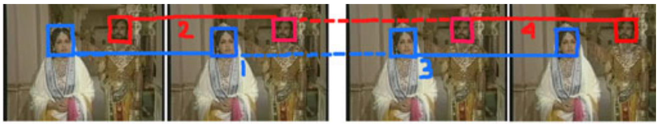

Fig. 3. TC at Tracklet level: Blue tracklets 1,3 are spatio-temporally close (connected by broken lines), and belong to the same person. Similarly red tracklets 2 and 4.

2 Problem Definition

In this section, we elaborate on our task of tracklet clustering for entity discovery in videos.

2.1 Notation

In this work, given a video, we fix beforehand the *type of entity* (eg. person/face, cars, planes, trees) we are interested in, and choose an appropriate detector like [29], [35], which is run on every frame of the input video. The detections in successive frames are then linked based on spatial locality, to obtain tracklets. At most R detections from R contiguous frames are linked like this. The tracklets of length less than R are discarded, hence all tracklets consist of R detections. We restrict the length of tracklets so that the appearance of the detections remain almost unchanged (due to detection-level TC), which facilitates tracklet representation. At $R = 1$ we work with the individual detections.

We represent a detection by a vector of dimension d. This can be done by downscaling a rectangular detection to $d \times d$ square and then reshaping it to a d^2-dimensional vector of pixel intensity values (or some other features if deemed appropriate). Each tracklet i is a collection of R detections $\{I_1^i, \ldots, I_R^i\}$. Let the tracklet i be represented by $Y_i = \frac{\sum_{j=1}^R I_j^i}{R}$. So finally we have N vectors (N: number of tracklets).

The tracklets can be sorted topologically based on their spatio-temporal positions, like starting and ending frame

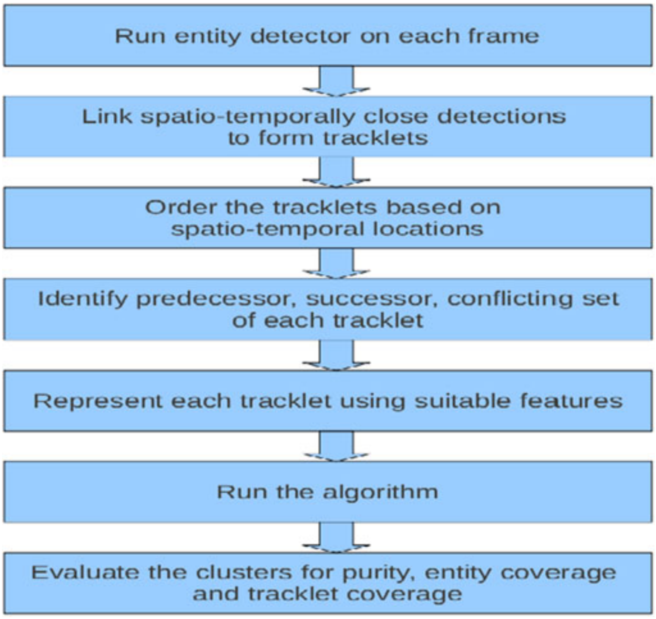
Run entity detector on each frame
Link spatio-temporally close detections to form tracklets
Order the tracklets based on spatio-temporal locations
Identify predecessor, successor, conflicting set of each tracklet
Represent each tracklet using suitable features
Run the algorithm
Evaluate the clusters for purity, entity coverage and tracklet coverage

Fig. 4. A flow-chart illustrating the overall approach, especially the preprocessing.

indices. Each tracklet i has a *predecessor tracklet* $prev(i)$ and a *successor tracklet* $next(i)$ based on the spatio-temporal locations of tracklets in the video. A tracklet i is the predecessor of tracklet j if the last detection of i appears in a frame closer to the frame containing the first detection of j, and also if the spatial coordinates of the last detection of tracklet i are closer to those of the first detection of tracklet j, compared to the last detection of any other tracklet. Similarly, *successor* is also defined. Each tracklet has an unique predecessor and successor. For example, in Fig. 3, the tracklet length $R = 2$, and there are 4 tracklets. They are numbered based on their spatio-temporal locations. $prev(3) = 1$ and $prev(4) = 2$, while $succ(1) = 3$ and $succ(2) = 4$.

Each tracklet i has a conflicting set of tracklets $CF(i)$ which span frame(s) that overlap with the frames spanned by i. Each detection (and tracklet) is associated with an entity, which are unknown in number, but presumably much less than the number of detections (and tracklets). These entities are represented by vectors, say $\phi_1, \phi_2, \ldots, \phi_K$ - the *appearance models* for the entities. Each tracklet i is associated with an entity indexed by Z_i, i.e., $Z_i \in \{1, 2, \ldots, K\}$. So, in Fig. 3 $CF(1) = 2, CF(2) = 1, CF(3) = 4, CF(4) = 3$. Also, $Z_1 = Z_3 = 1, Z_2 = Z_4 = 2$. The system is illustrated in Fig. 4 by a flow-chart.

2.2 Entity Discovery

Let each video be represented as a sequence of d-dimensional vectors $\{Y_1, \ldots, Y_N\}$ along with the set $\{prev(i), next(i), CF(i)\}_{i=1}^{N}$. We aim to learn the vectors $\{\phi_1, \phi_2, \ldots, \}$ and the assignment variables $\{Z_i\}_{i=1}^{N}$. In addition, we have *constraints* arising out of *temporal coherence* and other properties of videos. Each tracklet i is likely to be associated with the entity that its predecessor or successor is associated with, except at shot/scene changepoints. Moreover, a tracklet i cannot share an entity with its conflicting tracklets $CF(i)$, as the same entity cannot occur twice in the same frame. This notion is considered in relevant literature [13], [24]. Mathematically, the constraints are:

$$\begin{aligned} Z_{prev(i)} = Z_i = Z_{next(i)} \forall i \in \{1, \ldots, N\} w.h.p. \\ Z_i \notin \{Z_j : j \in CF(i)\} \forall i \in \{1, \ldots, N\}. \end{aligned} \tag{1}$$

These constraints give the task a flavour of non-parametric constrained clustering with must-link and don't-link constraints. *Learning a ϕ_k-vector is equivalent to discovering an entity by learning its appearance model, and its occurences are found by learning the set of associated tracklets $\{i : Z(i) = k\}$.*

Finally, the video frames can be grouped into short segments, based on the starting frame numbers $F(1), F(2), \ldots, F(N)$ of the N tracklets. Consider any tracklet i and its successor $j = succ(i)$, with starting frames $F(i)$ and $F(j)$. If the gap between frames $F(i)$ and $F(j)$ is larger than some threshold (much longer than the tracklet length R), then we consider a new temporal segment of the video starting from $F(j)$, and add j to a list of *changepoints* (CP). No tracklet can be present across segments. The beginning of a new temporal segment does not necessarily mean a scene change, the large gap between frames $F(i)$ and $F(j)$ may be caused by failure of detection or tracklet creation. The segment index of each tracklet i is denoted by $S(i)$.

3 Related Work

Person discovery in videos is a task which has recently received attention in Computer Vision. Cast Listing [36] is aimed to choose a representative subset of the face detections or face tracks in a movie/TV series episode. Another task is to label *all the detections* in a video, but this requires movie scripts [37] or labelled training videos having the same characters [38]. Scene segmentation and person discovery are done simultaneously using a generative model in [44], but once again with the help of scripts. An unsupervised version of this task is considered in [24], which performs *face clustering* in presence of spatio-temporal constraints as already discussed. For this purpose they use a Markov Random Field, and encode the constraints as clique potentials. Another recent approach to face clustering is [13] which incorporates some spatio-temporal constraints into subspace clustering.

Tracklet association tracking is a core topic in computer vision, in which a target object is located in each frame based on appearance similarity and spatio- temporal locality. A more advanced task is *multi-target tracking* [33], in which several targets are present per frame. A tracking paradigm that is particularly helpful in multi-target tracking is *tracking by detection* [34], where object-specific detectors like [35] are run per frame (or on a subset of frames), and the detection responses are linked to form tracks. From this came the concept of *tracklet* [28] which attempts to do the linking hierarchically. This requires pairwise similarity measures between tracklets. Multi-target tracking via tracklets is usually cast as Bipartite Matching, which is solved using Hungarian Algorithm. Tracklet association and face clustering are done simultaneously in [25] using HMRF. The main difference of face/tracklet clustering and person discovery is that, the number of clusters to be formed is not known in the latter.

A somewhat related task was attempted more recently in [46], where the aim is to find "topical objects" or objects that appear repeatedly in the video. This may be used to identify leading actors in a movie/TV-series. They make use of a parametric Bayesian approach (topic model). However, the method is quite different, as they do not consider temporal coherence but instead focus on co-occurring low-level visual features.

Independent of videos, *Constrained Clustering* is itself a field of research. Constraints are usually *must-link and don't-link*, which specify pairs which should be assigned the same cluster, or must not be assigned the same cluster. The constraints can be hard [7] or soft/probabilistic [8]. Constrained Spectral Clustering has also been studied recently [11], [12], which allow constrained clustering of datapoints based on arbitrary similarity measures.

All the above methods suffer from a major defect- the number of clusters needs to be known beforehand. A way to avoid this is provided by *Dirichlet Process*, which is able to identify the number of clusters from the data. It is a mixture model with infinite number of mixture components, and each datapoint is assigned to one component. A limitation of DP is that it is exchangeable, and cannot capture sequential structure in the data. For this purpose, a Markovian variation was proposed: Hierarchical Dirichlet Process- Hidden Markov Model (HDP-HMM). A variant of this is the *sticky* HDP-HMM (sHDP-HMM) [18], which was proposed for temporal coherence in speech data for the task of speaker diarization, based on the observation that successive datapoints are likely to be from the same speaker and so should be assigned to the same component. Another Bayesian nonparametric approach for sequential data is the Distance-Dependent Chinese Restaurant Process (DDCRP) [20], which defines distances between every pair of datapoints, and each point is linked to another with probability proportional to such distances. A BNP model for subset selection is Indian Buffet Process (IBP) [21], a generative process for a sequence of binary vectors. This has been used for selecting a sparse subset of mixture components (topics) in Focussed Topic Modelling [22] as the Compound Dirichlet Mixture Model.

Finally, *Video Summarization* has been studied for a few years in the Computer Vision community. The aim is to provide a short but comprehensive summary of videos. This summary is usually in the form of a few *keyframes*, and sometimes as a short segment of the video around these keyframes. A recent example is [41] which models a video as a matrix, each frame as a column, and each keyframe as a *basis vector*, in terms of which the other columns are expressed. A more recent work [39] considers a kernel matrix to encode similarities between pairs of frames, uses it for *Temporal Segmentation* of the video, assigns an importance label to each of these segments using an SVM (trained from segmented and labelled videos), and creates the summary with the important segments. However, such summaries are in terms of low-level visual features, rather than high-level semantic features which humans use. An attempt to bridge this gap was made in [42], which defined movie scenes and summaries in terms of characters. This work used face detections along with *movie scripts* for semantic segmentation into shots and scenes, which were used for summarization. Very recently, semantic summarization has been attempted in case of surveillance [43], where frequently appearing traffic movement patterns forming the basis of summarization.

4 Generative Process for Tracklets

We now explain our Bayesian Nonparametric model TC-CRP to handle the spatio-temporal constraints (Eq. 1) for tracklet clustering, and describe a generative process for videos based on tracklets.

4.1 Bayesian Nonparametric Modelling

In Section 2, we discussed the vectors $\phi_1, \phi_2, \ldots$ each of which represent an entity. In this paper we consider a Bayesian approach with Gaussian Mixture components $\mathcal{N}(\phi_k, \Sigma_1)$ to account for the variations in visual features of the detections, say face detections of a person. As already mentioned, number of components K is not known beforehand, and must be discovered from the data. That is why we consider nonparametric Bayesian modelling. Also, as we shall see, this route allows us to elegantly model the temporal coherence constraints. In this approach, we shall represent entities as mixture components and tracklets as draws from such mixture components.

Dirichlet Process [14] has become an important clustering tool in recent years. Its greatest strength is that unlike K-means, it is able to discover the correct number of clusters. Dirichlet Process is a distribution over distributions over a measurable space. A discrete distribution P is said to be distributed as $DP(\alpha, H)$ over space A if for every finite partition of A as $\{A_1, A_2, \ldots, A_K\}$, the quantity $\{P(A_1), \ldots, P(A_K)\}$ is distributed as $Dirichlet(\alpha H(A_1), \ldots, \alpha H(A_K))$, where α is a scalar called *concentration parameter*, and H is a distribution over A called Base Distribution. A distribution $P \sim DP(\alpha, H)$ is a discrete distribution, with infinite support set $\{\phi_k\}$, which are draws from H, called *atoms*.

4.2 Modeling Tracklets by Dirichlet Process

We consider H to be a $d-dimensional$ multivariate Gaussian with parameters μ and Σ_0. Each atom corresponds to an entity (e.g. a person). The generative process for the set $\{Y_i\}_{i=1}^N$ is then as follows:

$$P \sim DP(\alpha, H); X_i \sim P, Y_i \sim \mathcal{N}(X_i, \Sigma_1) \forall i \in [1, N]. \quad (2)$$

Here X_i is an atom. Y_i is a tracklet representation corresponding to the entity, and its slight variation from X_i (due to effects like lighting and pose variation) is modelled using $\mathcal{N}(X_i, \Sigma_1)$.

Using the constructive definition of Dirichlet Process, called the Stick-Breaking Process [15], the above process can also be written equivalently as

$$\hat{\pi}_k \sim Beta(1, \alpha), \pi_k = \hat{\pi}_k \prod_{i=1}^{k-1} (1 - \hat{\pi}_{i-1}), \phi_k \sim H \forall k \in [1, \infty)$$
$$Z_i \sim \pi, Y_i \sim \mathcal{N}(\phi_{Z_i}, \Sigma_1) \forall i \in [1, N]. \quad (3)$$

Here π is a distribution over integers, and Z_i is an integer that indexes the component corresponding to the tracklet i. Our aim is to discover the values ϕ_k, which will give us the entities, and also to find the values $\{Z_i\}$, which define a clustering of the tracklets. For this purpose we use collapsed Gibbs Sampling, where we integrate out the P in Equation 2 or G in Equation 3. The Gibbs Sampling Equations $p(Z_i | Z_{-i}, \{\phi_k\}, Y)$

and $p(\phi_k|\phi_{-k},Z,Y)$ are given in [16]. For Z_i,

$$p(Z_i=k|Z_{-i},\phi_k,Y_i)\propto p(Z_i=k|Z_{-i})p(Y_i|Z_i=k,\phi) \quad (4)$$

Here, $p(Y_i|Z_i=k,\phi)=\mathcal{N}(Y_i|\phi_k,\Sigma_1)$ is the data likelihood term. We focus on the part $p(Z_i=k|Z_{-i})$ to model TC.

4.3 Temporally Coherent Chinese Restaurant Process

In the generative process (Equation 3) all the Z_i are drawn IID conditioned on π. Such models are called *Completely Exchangeable*. This is, however, often not a good idea for sequential data such as videos. In Markovian Models like sticky HDP-HMM, Z_i is drawn conditioned on π and Z_{i-1}. In case of DP, the independence among Z_i-s is lost on integrating out π. After integration the generative process of Eq. 3 can be redefined as

$$\phi_k\sim H\forall k;\, Z_i|Z_1,\dots,Z_{i-1}\sim CRP(\alpha);\, Y_i\sim\mathcal{N}(\phi_{Z_i},\Sigma_1). \quad (5)$$

The predictive distribution for $Z_i|Z_1,\dots,Z_{i-1}$ for Dirichlet Process is known as Chinese Restaurant Process (CRP). It is defined as $p(Z_i=k|Z_{1:i-1})=\frac{N_k^i}{N-1+\alpha}$ **if** $k\in\{Z_1,\dots,Z_{i-1}\}$; $=\frac{\alpha}{N-1+\alpha}$ **otherwise** where N_k^i is the number of times the value k is taken in the set $\{Z_1,\dots,Z_{i-1}\}$.

We now modify CRP to handle the Spatio-temporal cues (Eq. 1) mentioned in the previous section. In the generative process, we define $p(Z_i|Z_1,\dots,Z_{i-1})$ with respect to $prev(i)$, similar to the Block Exchangeable Mixture Model as defined in [19]. Here, with each Z_i we associate a *binary change variable* C_i. If $C_i=0$ then $Z_i=Z_{prev(i)}$, i.e., the tracklet identity is maintained. But if $C_i=1$, a new value of Z_i is sampled. Note that every tracklet i has a temporal predecessor $prev(i)$. However, if this predecessor is spatio-temporally close, then it is more likely to have the same label. So, the probability distribution of change variable C_i should depend on this closeness. In TC-CRP, we use two values (κ_1 and κ_2) for the Bernoulli parameter for the change variables. We put a threshold on the spatio-temporal distance between i and $prev(i)$, and choose a Bernoulli parameter for C_i based on whether this threshold is exceeded or not. Note that maintaining tracklet identity by setting $C_i=0$ is equivalent to *tracking*.

Several datapoints (tracklets) arise due to false detections. We need a way to model these. Since these are very different from the Base mean μ, we consider a separate component $Z=0$ with mean μ and a very large covariance Σ_2, which can account for such variations. The Predictive Probability function(PPF) for TC-CRP is defined as follows:

$$\begin{aligned} T(Z_i=k|Z_{1:i-1},C_{1:i-1},C_i=1) &= 0 \textbf{ if } k\in\{Z_{CF(i)}\}-\{0\}\\ &\propto\beta \textbf{ if } k=0\\ &\propto n_{k1}^{ZC} \textbf{ if } k\in\{Z_1,\dots,Z_{i-1}\}, k\notin\{Z_{CF(i)}\}\\ &\propto\alpha \textbf{ otherwise}, \end{aligned} \quad (6)$$

where $Z_{CF(i)}$ is the set of values of Z for the set of tracklets $CF(i)$ that overlap with i, and n_{k1}^{ZC} is the number of points j ($j<i$) where $Z_j=k$ and $C_j=1$. The first rule ensures that two overlapping tracklets cannot have same value of Z. The second rule accounts for false tracklets. The third and fourth rules define a CRP restricted to the changepoints where $C_j=1$. The final tracklet generative process is as follows:

Algorithm 1. TC-CRP Tracklet Generative Process

```
$\phi_k\sim\mathcal{N}(\mu,\Sigma_0)\ \forall k\in[1,\infty)$
for $i=1:N$ do
    if $dist(i,prev(i))\le thres$ then
        $C_i\sim Ber(\kappa1)$
    else
        $C_i\sim Ber(\kappa2)$
    end if
    if $C_i=1$ then
        draw $Z_i\sim T(Z_i|Z_1,\dots,Z_{i-1},C_1,\dots,C_{i-1},\alpha)$
    else
        $Z_i=Z_{prev(i)}$
    end if
    if $Z_i=0$ then
        $Y_i\sim\mathcal{N}(\mu,\Sigma_2)$
    else
        $Y_i\sim\mathcal{N}(\phi_{Z_i},\Sigma_1)$
    end if
end for
```

where T is the PPF for TC-CRP, defined in Eq. 6.

4.4 Inference

Inference in TC-CRP can be performed easily through Gibbs Sampling. We need to infer C_i, Z_i and ϕ_k. As C_i and Z_i are coupled, we sample them in a block for each $i\in[1,N]$ as done in [19]. If $C_{i+1}=0$ and $Z_{i+1}\neq Z_{i-1}$, then we must have $C_i=1$ and $Z_i=Z_{i+1}$. If $C_{i+1}=0$ and $Z_{i+1}=Z_i$, then $Z_i=Z_{i+1}$, and C_i is sampled from $Bernoulli(\kappa)$. In case $C_{i+1}=1$ and $Z_{i+1}\neq Z_{i-1}$, then $(C_i=a,Z_i=k)$ with probability proportional to $p(C_i=a)p(Z_i|Z_{-i},C_i=a))p(Y_i|Z_i=k,\phi_k)$. If $a=0$ then $p(Z_i=k|Z_{-i},C_i=1)=1$ if $Z_{i-1}=k$, and 0 otherwise. If $a=1$ then $p(Z_i|Z_{-i},C_i=a))$ is governed by TC-CRP. For sampling ϕ_k, we make use of the Conjugate Prior formula of Gaussians, to obtain the Gaussian posterior with mean $(n_k\Sigma_1^{-1}+\Sigma^{-1})^{-1}(\Sigma_1^{-1}Y_k+\Sigma^{-1}\mu)$ where $n_k=|\{i:Z_i=k\}|$, and $Y_k=\sum_{i:Z_i=k}Y_i$. Finally, we update the hyperparameters μ and Σ after every iteration, based on the learned values of $\{\phi_k\}$, using Maximum Likelihood estimate. κ_1,κ_2 can also be updated, but in our implementation we set them to 0.001 and 0.1 respectively, based on empirical evaluation on one held-out video. The threshold $thres$ was also similarly fixed.

5 Generative Process for Video Segments

In the previous section, we considered the entire video as a single block, as the TCCRP PPF for any tracklet i involves (Z,C)-values from all the previously seen tracklets throughout the video. However, this need not be very accurate, as in a particular part of the video some mixture components (entities) may be more common than anywhere else, and for any i, Z_i may depend more heavily on the Z-values in temporally close tracklets than the ones far away. This is because, a TV-series video consists of *temporal segments* like scenes and shots, each characterized by a subset of persons (encoded by binary vector B_S). The tracklets attached to a segment s cannot be associated with persons not listed by

B_s. To capture this notion we propose a new model: Temporally Coherent Chinese Restaurant Franchise (TC-CRF) to model a video temporally segmented by S (see Section 2).

5.1 Temporally Coherent Chinese Restaurant Franchise

Chinese Restaurant Process is the PPF associated with Dirichlet Process. Hierarchical Dirichlet Process (HDP) [17] aimed at modelling *grouped data sharing same mixture components*. It assumes a group-specific distribution π_s for every group s. The generative process is:

$$\hat{p}_k \sim Beta(1,\alpha), p_k = \hat{p}_k \prod_{i=1}^{k-1}(1-\hat{p}_{i-1}), \phi_k \sim H\; \forall k \in [1,\infty)$$
$$\pi_s \sim p \forall s \in [1,M]; Z_i \sim \pi_{S(i)}, Y_i \sim \mathcal{N}(\phi_{Z_i}, \Sigma_1) \forall i \in [1,N], \tag{7}$$

where datapoint i belongs to the group $S(i)$. The PPF corresponding to this process is obtained by marginalizing the distributions p and $\{\pi\}$, and is called the *Chinese Restaurant Franchise* process, elaborated in [17]. In our case, we can modify this PPF once again to incorporate TC, analogously to TC-CRP, to have Temporally Coherent Chinese Restaurant Franchise (TC-CRF) Process.

Algorithm 2. TC-CRF Tracklet Generative Process

```
$\phi_k \sim \mathcal{N}(\mu, \Sigma_0)$
for $s = 1 : M$ do
    $B_s \sim IBP(\gamma, B_1, \ldots, B_{s-1})$
end for
for $i = 1 : N$ do
    if $S_i = S_{prev(i)}$ then
        if $dist(i, prev(i)) \le thres$ then
            $C_i \sim Ber(\kappa 1)$
        else
            $C_i \sim Ber(\kappa 2)$
        end if
    else
        $C_i = 1$
    end if
    if $C_i = 1$ then
        draw $Z_i \sim TF(Z_i|B_{S(i)}, Z_1, \ldots, Z_{i-1}, C_1, \ldots, C_{i-1}, \alpha)$
    else
        $Z_i = Z_{prev(i)}$
    end if
    if $Z_i = 0$ then
        $Y_i \sim \mathcal{N}(\mu, \Sigma_2)$
    else
        $Y_i \sim \mathcal{N}(\phi_{Z_i}, \Sigma_1)$
    end if
end for
```

In our case, a group corresponds to a temporal segment, and as already mentioned, we want a binary vector B_s, which indicates the components that are active in segment s. But HDP assumes that all the components are shared by all the groups, i.e., any particular component can be sampled in any of the groups. We can instead try *sparse modelling* by incorporating $\{B_s\}$ into the model, as done in [22] for Focussed Topic Models. For this purpose we put an IBP [21] prior on the $\{B_s\}$ variables, where $p(B_{sk} = 1|B_1, \ldots, B_{s-1}) \propto n_k$ where n_k is the number of times component k has been sampled in all scenes before s, and $p(B_{sk_{new}}|B_1, \ldots, B_{s-1}) \propto \gamma$. The *TC-CRF* PPF is then as follows:

$$\begin{aligned} TF(Z_i = k|B_s, Z_{1:i-1}, C_{1:i-1}, C_i = 1) &= 0 \textbf{ if } k \in \{Z_{CF(i)}\} - \{0\} \\ &= 0 \textbf{ if } B_{sk} = 0 \\ &\propto \beta \textbf{ if } k = 0 \\ &\propto n^{SZC}_{sk1} \textbf{ if } B_{sk} = 1, k \in \{Z\}_s, k \notin \{Z_{CF(i)}\} \\ &\propto \alpha \textbf{ if } B_{sk} = 1, k \notin \{Z\}_s, k \notin \{Z_{CF(i)}\}, \end{aligned} \tag{8}$$

where $s = S(i)$, the index of the temporal segment to which the datapoint i belongs. Based on TC-CRF, the generative process of a video, in terms of temporal segments and tracklets, is given below: where *TF* is the PPF for TC-CRF, and $S(i)$ is the temporal segment index associated with tracklet i.

5.2 Inference

Inference in TC-CRF can also be performed through Gibbs Sampling. We need to infer the variables $\{B\}$, $\{C\}$, $\{Z\}$ and the components $\{\phi\}$. In segment s, for a datapoint i where $C_i = 1$, a component ϕ_k may be sampled with $p(B_{sk} = 1, Z_i = k|B_{-sk}, Z_{-i}) \propto n^{SZC}_{sk1}$, which is the number of times ϕ_k has been sampled within the same segment. If ϕ_k has never been sampled within the segment but has been sampled in other segments, $p(B_{sk} = 1, Z_i = k|B_{-sk}, Z_{-i}) \propto \alpha n_k$, where n_k is the number of segments where ϕ_k has been sampled (Corresponding to $p(B_{sk}) = 1$ according to IBP), and α is the CRP parameter for sampling a new component. Finally, a completely new component may be sampled with probability proportional to α. Note that $p(B_{sk} = 0, Z_i = k) = 0 \forall k$.

6 Relationship with Existing Models

TC-CRP draws inspirations from several recently proposed Bayesian nonparametric models, but is different from each of them. It has three main characterestics: 1) Changepoint-variables $\{C\}$ 2) Temporal Coherence and Spatio-temporal cues 3) Separate component for non-face tracklets. The concept of changepoint variable was used in Block-exchangeable Mixture Model [19], which showed that this significantly speeds up the inference. But in BEMM, the Bernoulli parameter of changepoint variable C_i depends on $Z_{prev(i)}$ while in TC-CRP it depends on $dist(i, prev(i))$. Regarding spatio-temporal cues, the concept of providing additional weightage to self-transition was introduced in sticky HDP-HMM [18], but this model does not consider change-point variables. Moreover, it uses a transition distribution P_k for each mixture component k, which increases the model complexity. Like BEMM [19] we avoid this step, and hence our PPF (Eq. (6)) does not involve $Z_{prev(i)}$. DDCRP [20] defines distances between every pair of datapoints, and associates a new datapoint i with one of the previous ones $(1, \ldots, i-1)$ based on this distance. Here we consider distances between a point i and its predecessor $prev(i)$ only. On the other hand, DDCRP is unrelated to the original DP-based CRP, as its PPF does not consider n^Z_k: the number of previous datapoints assigned to component k.

Hence our method is significantly different from DDCRP. Finally, the first two rules of TC-CRP PPF are novel.

TC-CRF is inspired by HDP [17]. However, once again there are three differences mentioned above hold good. In addition, the PPF of TC-CRF itself is different from Chinese Restaurant Franchise as described in [17]. The original CRF is defined in terms of two concepts: tables and dishes, where tables are local to individual restaurants (data groups) while dishes (mixture components) are global, shared across restaurants (groups). Also individual datapoints are assigned mixture components indirectly, through an intermediate assignment of tables. The concept of table, which comes due to marginalization of group-specific mixture distributions, results in complex book-keeping, and the PPF for datapoints is difficult to define. Here we avoid this problem, by skipping tables and directly assigning mixture components to datapoints in Eq. (8). Inspiration of TC-CRF is also drawn from IBP-Compound Dirichlet Process [22]. But the inference process of [22] is complex, since the convolution of the DP-distributed mixture distribution and the sparse binary vector is difficult to marginalize by integration. We avoid this step by directly defining the PPF (Eq. (8)) instead of taking the DP route. This approach of directly defining the PPF was taken for DD-CRP [20] also.

7 Experiments on Person Discovery

One particular entity discovery task that has recently received a lot of attention is person discovery from movies/ TV series. We carried out extensive experiments for person discovery on TV series videos of various lengths.

Benchmark Videos First of all, we made use of three videos that have been used recently for the related task of *face clustering*, in [24] and [25]. These are *Frontal*, *Notting Hill* and *Big Bang Theory- Season 1 Episode 1*. These videos have only 4 or 5 persons. Notting-Hill video has 4,660 face detections and Frontal has 4,267. In contrast, Big Bang Theory (Season 1 Episode 1) has 25,523 face detections. In these videos, the ground-truth clusterings are available.

New Videos We collected three episodes of The Big Bang Theory (Season 1). Each episode is 20-22 minutes long, and has 7-8 characters (occurring in at least 100 frames). We also collected 6 episodes of the famous Indian TV series "The Mahabharata" from Youtube. Each episode of this series is 40-45 minutes long, and has 15-25 prominent characters. So here, each character is an entity. These videos are much longer than those studied in similar works like [25], and have more characters. Also, these videos are challenging because of the somewhat low quality and motion blur. Transcripts or labeled training sets are unavailable for all these videos. As usual in the literature [24], [25], we represent the persons with their faces. We obtained face detections by running the OpenCV Face Detector on each frame separately. As described in Section 2 the face detections were all converted to grayscale, scaled down to 30×30, and reshaped to form 900-dimensional vectors. We considered tracklets of size $R = 10$ and discarded smaller ones. The dataset details are given in Table 1.

TABLE 1
Details of Datasets

Dataset	#Frames	#Detections	#Tracklets	#Entities	Entity Type
BBTs1e1	32,248	25,523	2,408	7	Person(Face)
BBTs1e3	31,067	21,555	1,985	9	Person(Face)
BBTs1e4	28,929	20,819	1,921	8	Person(Face)
Maha22	66,338	37,445	3,114	14	Person(Face)
Maha64	72,657	65,079	5,623	16	Person(Face)
Maha65	68,943	53,468	4,647	22	Person(Face)
Maha66	87,202	76,908	6,893	17	Person(Face)
Maha81	78,555	62,755	5,436	22	Person(Face)
Maha82	86,153	52,310	4,262	24	Person(Face)

7.1 Alternative Methods

Very recently, HMRF-based constrained clustering has been used for face clustering in [24] and [25]. This method also requires the number of clusters to be specified. However, we found that this method runs into numerical problems whenever the number of clusters is over 10, which is the case in many of our videos. So, we limited comparison with this method to only the benchmark videos, namely Notting-Hill, Frontal and BBTs1e1.

A recent method for face clustering using track information is WBSLRR [13] based on Subspace Clustering. Though in [13] it is used for clustering detections rather than tracklets, the change can be made easily. Apart from that, we can use K-means Clustering and Constrained Clustering as baselines, and we choose a recent method [12]. TC and frame conflicts are encoded as must-link and don't-link constraints respectively. A big problem is that the number of clusters to be formed is unknown. For this purpose, we note that the *tracklet matrix* formed by juxtaposing the tracklet vectors should be *approximately low-rank* because of the similarity of spatio-temporally close tracklet vectors. Such representation of a video as a low-rank matrix has been attempted earlier [4], [30]. We can find a low-rank representation of the tracklet matrix by any suitable method, and use the rank as the number of clusters to be formed in spectral clustering. We found that, among these the best performance is given by Sparse Bayesian Matrix Recovery (SBMR) [6]. Others are either too slow (BRPCA [5]), or recover matrices with ranks too low (OPTSPACE [3]) or too high (RPCA [4]). Finally, we compare against another well-known sequential BNP method- the sticky HDP-HMM [18].

7.2 Performance Measures

In case of face clustering, the measure used in [24] and [25] is *clustering accuracy*. However, this can be used only if the ground-truth clustering is known. For long videos this is difficult to annotate, so we need other measures.

The task of entity discovery with all their tracks is novel and complex, and has to be judged by suitable measures. We discard the clusters that have less than 10 assigned tracklets. It turns out that the remaining clusters cover about $85 - 95$ % of all the tracklets. Further, there are some clusters which have mostly (70 % or more) false (non-entity) tracklets. We discard these from our evaluation. We call the remaining clusters as *significant clusters*. *We say that a significant cluster k is "pure" if at least* 70 % *of the tracklets assigned to it belong to any one person A*(say Sheldon for a BBT video, or Arjuna for a Mahabharata video). We also declare that the cluster k and its corresponding mixture component ϕ_k corresponds to the person A. Also, then A is considered to be *discovered*. The threshold of purity was set to 70 percent

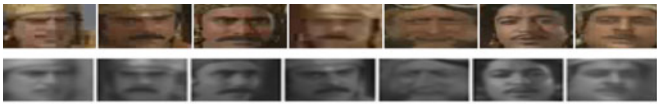

Fig. 5. Face detections (top), and the corresponding atoms (reshaped to square images) found by TC-CRP (bottom).

TABLE 2
Comparison of TCCRP and HMRF by Clustering Accuracy on the Benchmark Dataset

Dataset	TCCRP	HMRF
Frontal	0.74	**0.91**
Notting-Hill	**0.86**	0.77
BBTs1e1	**0.70**	0.66

TABLE 3
Purity Results for Different Methods

Dataset	TCCRF	TCCRP	sHDP -HMM	SBMR+ ConsClus	SBMR+ Kmeans	WBS -LRR
BBTs1e1	**0.88** (48)	0.75 (36)	0.84 (44)	0.67 (48)	0.77 (56)	0.73 (45)
BBTs1e3	**0.88** (50)	0.83 (40)	0.76 (37)	0.80 (15)	0.72 (46)	0.67 (43)
BBTs1e4	**0.93** (40)	0.89 (36)	0.83 (29)	0.77 (31)	0.74 (46)	0.71 (41)
Maha22	0.91 (67)	0.87 (69)	0.86 (74)	**0.94** (44)	0.65 (68)	0.83 (79)
Maha64	**0.95** (113)	0.92 (105)	0.91 (97)	0.85 (88)	0.82 (90)	0.75 (81)
Maha65	**0.97** (95)	0.89 (85)	0.90 (89)	0.86 (76)	0.86 (83)	0.82 (84)
Maha66	0.91 (76)	**0.96** (73)	0.95 (80)	0.87 (84)	0.86 (90)	0.81 (81)
Maha81	**0.89** (91)	**0.89** (88)	0.84 (95)	0.87 (84)	0.70 (86)	0.74 (78)
Maha82	**0.92** (52)	0.88 (50)	0.86 (58)	0.78 (63)	0.78 (65)	0.83 (64)

The number of significant clusters are written in brackets.

TABLE 4
Entity Coverage Results for Different Methods

Dataset	TCCRF	TCCRP	sHDP -HMM	SBMR+ ConsClus	SBMR+ Kmeans	WBS -LRR
BBTs1e1	**6**	**6**	5	5	4	4
BBTs1e3	**9**	7	6	8	6	7
BBTs1e4	6	**8**	**8**	6	6	**8**
Maha22	**14**	**14**	**14**	10	10	**14**
Maha64	**14**	13	**14**	11	13	13
Maha65	17	**19**	17	13	16	17
Maha66	13	**15**	13	9	10	11
Maha81	**21**	**21**	20	14	18	20
Maha82	**21**	19	20	10	12	16

TABLE 5
Tracklet Coverage Results for Different Methods

Dataset	TCCRF	TCCRP	sHDP -HMM	SBMR+ ConsClus	SBMR+ Kmeans	WBS -LRR
BBTs1e1	**0.82**	0.67	0.79	0.29	0.70	0.73
BBTs1e3	0.86	**0.88**	0.68	0.09	0.71	0.53
BBTs1e4	**0.92**	0.82	0.78	0.22	0.74	0.62
Maha22	**0.90**	**0.90**	0.86	0.43	0.68	0.69
Maha64	**0.93**	0.90	0.81	0.39	0.81	0.62
Maha65	**0.94**	0.85	0.91	0.40	0.80	0.68
Maha66	0.74	0.80	0.68	0.43	**0.84**	0.65
Maha81	**0.80**	0.75	0.66	0.46	0.60	0.50
Maha82	0.76	**0.81**	0.64	0.37	0.77	0.64

because we found this roughly the minimum purity needed to ensure that a component mean is visually recognizable as the entity (after reshaping to $d \times d$) (See Fig. 5). We measure the *Purity: fraction of significant clusters that are pure, i.e., correspond to some entity*. We also measure *Entity Coverage: the number of persons (entity) with at least 1 cluster (at least 10 tracklets) corresponding to them*. Next, we measure *Tracklet Coverage: the fraction of tracklets that are assigned to pure clusters*. Effectively, these tracklets are *discovered*, and the ones assigned to impure clusters are lost.

We also inspected the "false" clusters, i.e., the ones discarded for having over 70 percent false tracklets. Most of them were entirely made of such false tracklets, and less than 0.5 percent of the entity tracklets are "lost" by discarding these clusters. Also, none of the competing methods has any advantage over the others in this regard.

7.3 Comparison with HMRF on Benchmark Videos

First, we compare TC-CRP to HMRF [24], [25] in terms of face clustering accuracy. Here, the comparison is limited to the three benchmark videos where face clustering results by HMRF is available. It is not possible to compare the videos that we collected, since we do not have the ground-truth clustering for them.

In this comparison, for the parametric method (HMRF) the number of clusters is set to the known number of persons (as done in the papers). In case of TCCRP, this number is estimated. The results are shown in Table 2. Unfortunately, TCCRP always finds more clusters than the true number of entities, so that the clustering accuracy suffers. Still, TCCRP has the better performance on BBTs1e1 and Notting-Hill. Here, TCCRF is not used since the Frontal video is short, and for Notting-Hill we do not have the shot information.

7.4 Person Discovery Results

Next, we move on to person discovery, on the datasets that we discussed in Table 1. Here, the number of clusters is not provided to any method, and for parametric methods like Constrained Clustering and WBSLRR, this number is estimated as explained in Section 7.1. Here, HMRF is not applicable since it faces numerical problems due to the larger number of clusters to be formed. The results on the three novel measures discussed above are shown in Tables 3, 4, and 5. In terms of the three measures, TC-CRF is usually the most accurate, followed by TC-CRP, and then sHDP-HMM. This demonstrates that BNP methods are more suitable to the task. The constrained spectral clustering-based method is competitive on the purity measure, but fares very poorly in terms of tracklet coverage. This is because, it forms many small pure clusters, and a few very large impure clusters which cover a huge fraction of the tracklets. Thus, a large number of tracklets are lost. Also, it appears that K-means can be better than constrained clustering on the short BBT videos in terms of cluster purity, but the importance of the constraints becomes evident on the longer videos with more persons.

In the above experiments, we used tracklets with size $R = 10$. We varied this number and found that, for $R = 5$ and even $R = 1$ (dealing with detections individually), the performance of TC-CRF, TC-CRP and sHDP-HMM did not change significantly. On the other hand, the matrix returned

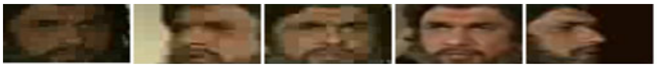

Fig. 6. Different atoms for different poses of same person.

by SBMR had higher rank (120-130 for $R = 1$) as the number of tracklets increased.

Number of significant clusters: It may be noted that the *number of significant clusters formed* is a matter of concern, especially from the user's perspective. A small number of clusters allow him/her to get a quick summary of the video. Ideally there should be one cluster per entity, but that is not possible due to the significant appearance variations (See Fig. 6). The number of clusters formed per video by the different methods is indicated in Table 3. It appears that none of the methods have any clear advantage over the others in this regard. However, for a direct comparison we make another experiment. Here, the number of clusters for the parametric methods is set in such a way that they form roughly the same number of significant clusters as the proposed method which gives best result (TCCRF or TCCRP). These results are shown in Table 6. Table 6 shows that, even for same number of significant clusters TCCRF/TCCRP continues to give best clustering purity.

TABLE 6
Cluster Purity Results for Different Methods When Number of Significant Clusters Is Same (Shown in Brackets)

Dataset	TC	ConsClus	Kmeans	WBSLRR
BBTs1e1(48)	**0.88**	0.67	0.84	0.73
BBTs1e3(50)	**0.88**	0.67	0.77	0.68
BBTs1e4(40)	**0.93**	0.68	0.73	0.71
Maha22 (67)	0.91	**0.96**	0.66	0.80
Maha64 (113)	**0.95**	0.88	0.87	0.83
Maha65 (95)	**0.97**	0.92	0.74	0.85
Maha66 (76)	**0.96**	0.87	0.80	0.78
Maha81 (91)	**0.89**	**0.91**	0.66	0.77
Maha82 (52)	**0.92**	0.75	0.75	0.80

The first method is either TCCRF or TCCRP- whichever does better.

7.5 Online Inference

We wanted to explore the case of streaming videos, where the frames appear sequentially and old frames are not stored. This is the online version of the problem, the normal Gibbs Sampling will not be possible. For each tracklet i, we will have to infer C_i and Z_i based on $C_{prev(i)}$, $Z_{prev(i)}$ and the $\{\phi_k\}$-vectors learnt from $\{Y_1, Y_2, \ldots, Y_{i-1}\}$. Once again, (C_i, Z_i) is sampled as a block as above, and the term $p(Z_i|Z_{-i}, C_i = a))$ follows from the TC-CRP PPF (Eq. 6). The same thing can be done for TC-CRF also. Instead of drawing one sample per data-point, an option is to draw several samples and consider the mode. In the absence of actual streaming datasets we performed the single-pass inference (Section 7.5) on two of the videos from each set- Mahabharata and Big Bang Theory. We used the same performance measures as above. The existing tracklet clustering methods discussed in Section 7.1 are incapable in the online setting, and sticky HDP-HMM is the only alternative. The results are presented in Table 7, which show TC-CRP to be doing the best on the Mahabharata videos and TC-CRF on the Big Bang Theory ones. Notably, the figures for TC-CRP and TC-CRF in the online experiment are not significantly lower than those in the offline experiment (except one or two exceptions), unlike sHDP-HMM. This indicates that the proposed methods converge quickly, and so are more efficient offline.

7.6 Outlier Detection / Discovery of False Tracklets

Face Detectors such as [29] are trained on static images, and applied on the videos on per-frame basis. This approach itself has its challenges [26], and the complex videos we consider in our experiments do not help matters. As a result, there is a significant number of *false (non-face) detections*, many of which occur in successive frames and hence get linked as tracklets. Identifying such junk tracklets not only helps us to improve the quality of output provided to the users, but may also help to adapt the detector to the new domain, by retraining with these new negative examples, as proposed in [27].

We make use of the fact that false tracklets are relatively less in number (compared to the true ones), and hence at least some of them can be expected to deviate widely from the mean of the tracklet vectors. This is taken care of in the TC-CRP tracklet model, through the component ϕ_0 that has very high variance, and hence is most likely to generate the unusual tracklets. We set this variance Σ_2 as $\Sigma_2 = c\Sigma_1$, where $c > 1$. The tracklets assigned $Z_i = 0$ are reported to be junk by our model. It is expected that high c will result in lower recall but higher precision (as only the most unusual tracklets will go to this cluster), and low c will have the opposite effect. We study this effect on two of our videos- Maha65 and Maha81 (randomly chosen) in Table 8 (See Fig. 7 for illustration). As baseline, we consider K-means or spectral clustering of the tracklet vectors. We may expect that one of the smaller clusters should contain mostly the junk tracklets, since faces are roughly similar (even if from different persons) and should be grouped together. However, for different values of K (2 to 10) we find that the

TABLE 7
Online (Single-Pass) Analysis on four Videos

Dataset	Maha65		
Measure	TC-CRF	TC-CRP	sHDPHMM
Purity	0.86 (56)	**0.89**(79)	0.84 (82)
Entity Coverage	14	15	**16**
Tracklet Coverage	0.75	**0.80**	0.77
Dataset	Maha81		
Measure	TC-CRF	TC-CRP	sHDPHMM
Purity	0.71 (55)	**0.84**(74)	0.70(57)
Entity Coverage	19	**21**	17
Tracklet Coverage	0.51	**0.62**	0.49
Dataset	BBTs1e1		
Measure	TC-CRF	TC-CRP	sHDPHMM
Purity	**0.87** (39)	0.73 (33)	0.50 (14)
Entity Coverage	**5**	3	3
Tracklet Coverage	**0.80**	0.65	0.40
Dataset	BBTs1e4		
Measure	TC-CRF	TC-CRP	sHDPHMM
Purity	**0.92** (45)	0.88 (32)	0.75(28)
Entity Coverage	**7**	6	**7**
Tracklet Coverage	**0.87**	0.81	0.67

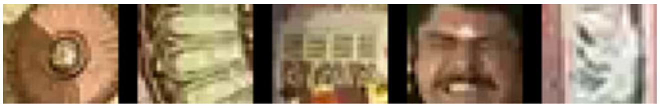

Fig. 7. Non-face tracklet vectors (reshaped) recovered by TC-CRP. Note that one face tracklet has been wrongly reported as non-face.

clusters are roughly of the same size, and the non-face tracklets are spread out quite evenly. Results are reported for the best K ($K = 10$ for both). Note that because of the large number of tracklets (Table I) it is difficult to count the total number of non-face ones. So for measuring *recall*, we simply mention the *number of non-face tracklets recovered (recall*)*, instead of the *fraction*. It is clear that TC-CRP significantly outperforms clustering on both precision and recall*.

7.7 Evaluation of TC Enforcement

The aim of TC-CRP and TC-CRF is to encourage TC at the semantic level, that spatio-temporally close but non-overlapping tracklets should belong to the same entity. In the Bayesian models like sHDP-HMM, TC-CRP and TC-CRF, these cues are modelled with probability distributions, in WBSLRR with convex regularization and in constrained clustering they are encoded as hard constraints. We now evaluate how well the different methods have been able to enforce these cues. We create ground-truth tracks by linking the tracklets which are spatio-temporally close to each other (with respect to the chosen threshold $thres$ in the generative process), and belong to the same entity. All the tracklets in each ground-truth track should be assigned to the same cluster. This is the task of *tracklet linking*. We measure *what fraction of the these ground-truth tracks have been assigned entirely to single clusters* by the different methods. We do not compare SBMR+ConsClus, since it uses hard constraints. The results are shown in Table 9. We find that TC-CRF is the best once again, followed by TC-CRP and sHDP-HMM. WBSLRR has significantly poorer performance, though it springs a surprise on BBTs1e1.

7.8 Scalability and Efficiency

The proposed methods TC-CRP and TC-CRF are far most scaleable than WBSLRR, because the latter requires several matrices for ADMM of size $N \times N$. The methods based on spectral clustering also require one $N \times N$ similarity matrix. The storage requirements for TC-CRP, sHDP-HMM and K-means is $O(N + K + DK)$, where D is the dimensionality and K is the number of clusters formed, which are both much smaller than N. The first component accounts for the assignments to discrete variables per tracklet (Z and C), the second component accounts for the CRP counts n^{ZC} and the last component accounts for the atoms or cluster centers. For TC-CRF, it is $O(N + SK + DK)$ where S is the number of temporal segments.

For WBSLRR the computational complexity per iteration is $O(N^2)$ and for SBMR it is $O(DN^3 + ND^3)$. For spectral clustering, the complexity is $O(N^3 + N^2D)$. For sHDP-HMM, TC-CRP and TC-CRF the complexity is $O(NDK)$ per iteration. Moreover, these methods are found to converge very fast. So it turns out that in terms of computational and space complexity also the proposed methods are far superior to the alternatives.

TABLE 8
Discovery of Non-Face Tracklets

Dataset	Maha65		Maha81	
Method	Precision	Recall*	Precision	Recall*
KMeans	0.22	73	0.19	39
Constrained Spectral	0.30	12	0.12	16
TCCRP (c=5)	0.98	79	0.57	36
TCCRP (c=4)	0.98	87	0.64	47
TCCRP (c=3)	0.95	88	0.62	54
TCCRP (c=2)	0.88	106	0.50	57

8 Discovery of Non-Person Entities

To emphasize the fact that our methods are not restricted to faces or persons, we used two short videos-one of cars and another of aeroplanes. The cars video consisted of five cars of different colors, while the aeroplanes video had six planes of different colors/shapes. These were created by concatenating shots of different cars/planes in the Youtube Objects datasets [45]. The objects were detected using the Object-specific detectors [35]. Since here the color is the chief distinguishing factor, we scaled the detections down to 30×30 and reshaped them separately in the 3 color channels to get 2,700-dimensional vectors. Here $R = 1$ was used, as these videos are much shorter, and using long tracklets would have made the number of data-points too low. Both videos have 750 frames. The *Cars* video has 694 detections, and the *Aeroplanes* video has 939 detections. The results are shown in Table 10 and Fig. 8. Once again, TC-CRP does well.

9 Semantic Video Summarization

In this section, we discuss how the above results on entity discovery can be used to obtain a sematic summary of the video. For this purpose we consider two approaches: entity-based and shot-based.

9.1 Entity-Based Summarization

The process of entity discovery via tracklet clustering results in formation of clusters. In case of the Bayesian methods like TC-CRF, TC-CRP and sHDP-HMM, each cluster can be represented by the mean vector of the corresponding mixture component. In case of non-Bayesian approaches like SBMR+Consclus and WBSLRR, it is possible to compute the cluster centers as the mean of the tracklet vectors assigned to each cluster. Each cluster vector ϕ_k can be reshaped to form a visual representation of the cluster. This representation of clusters provides us a visual list of the entities present in the video, which is what we call *entity-based summarization* of the video.

Any summary should have two properties: 1) It should be *concise* 2) It should be *representative*. Along these lines, an entity-based summary should have the property that it should cover as many entities as possible, with least number

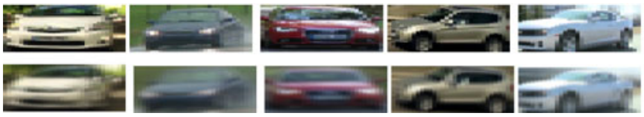

Fig. 8. Car detections (top), and the corresponding atoms (reshaped to square images) found by TC-CRP (bottom).

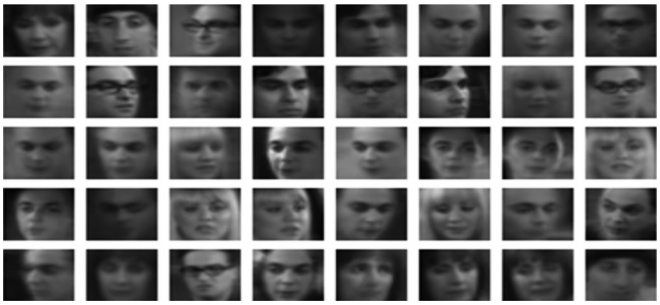

Fig. 9. Entity-based summarization of Big Bang Theory episode 4 using TC-CRF.

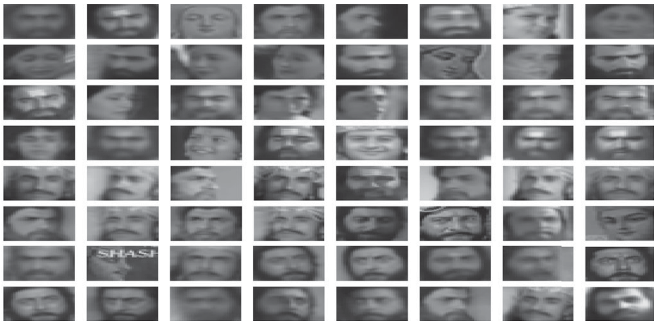

Fig. 10. Entity-based summarization of Mahabharata Episode 22 using TC-CRF. Each image is a reshaped cluster mean.

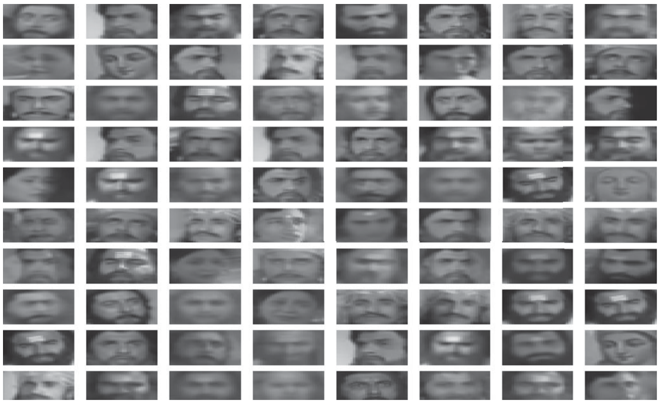

Fig. 11. Entity-based summarization of Mahabharata Episode 22 using WBSLRR. WBSLRR creates many more clusters than TC-CRF, but both discover the same number of persons (14). Hence the summary by TC-CRF is more concise.

of clusters. On the other hand, the selected clusters should cover a sufficiently large fraction of all the tracklets. In our evaluation of entity discovery (Section 7) we have measured *Entity Coverage*, *Tracklet Coverage* and *Number of significant clusters*. These same measures are useful in evaluating the summarization. Entity Coverage and Tracklet Coverage should be high, and number of significant clusters should be low (See Figs. 9, 10, and 11). To make the evaluation more comprehensive, we define two more measures: 1) *Conciseness:* defined as the ratio of Entity Coverage to the number of significant clusters, and 2) *Representativeness:* defined as the ratio of the Tracklet Coverage to the number of significant clusters.

The results are shown in the Tables 11 and 12. We find that in terms of Conciseness, TC-CRP turns out to be the best,

TABLE 9
Fraction of Ground Truth Tracks That Are Fully Linked

Dataset	TCCRF	TCCRP	sHDPHMM	WBSLRR
BBTs1e1	0.65	0.54	0.42	**0.93**
BBTs1e3	**0.74**	0.71	0.59	0.22
BBTs1e4	**0.72**	0.69	0.54	0.34
Maha22	**0.83**	0.81	0.80	0.61
Maha64	**0.80**	**0.80**	0.79	0.55
Maha65	**0.86**	0.81	0.81	0.63
Maha66	**0.86**	0.79	0.78	0.52
Maha81	**0.86**	0.82	0.83	0.61
Maha82	**0.89**	0.86	0.84	0.64

TABLE 10
Purity, Entity Coverage and Tracklet Coverage Results for Different Methods on Cars and Aeroplanes Videos

Dataset	TCCRP	sHDPHMM	SBMR+ ConsClus	WBSLRR
Cars	0.94 (35)	0.92 (12)	**1.00** (54)	0.24 (21)
Aeroplanes	**0.95** (43)	0.87 (15)	0.84 (44)	0.21 (24)
Dataset	**TCCRP**	**sHDPHMM**	**SBMR+ ConsClus**	**WBSLRR**
Cars	**5**	**5**	**5**	2
Aeroplanes	**6**	5	**6**	4
Dataset	**TCCRP**	**sHDPHMM**	**SBMR+ ConsClus**	**WBSLRR**
Cars	0.73	0.69	**1.00**	0.04
Aeroplanes	**0.93**	0.70	0.88	0.09

while the other methods are all comparable when averaged across the videos. In terms of Representativeness, TC-CRP is once again the best by a long way, while TC-CRF and sHDP-HMM are at par. The non-Bayesian methods are way behind.

9.2 Shot-Based Summarization

Another way of summarization is by a collection of *shots*. [42] follows this approach, and selects a subset of the shots based on the total number of characters (entities), number of prominent characters (entities) etc. A shot is a contiguous sequence of frames that consist of the same set of entities. It is possible to organize the video into temporal segments based on the cluster indices assigned to the tracklets. In a frame f, let $\{Z\}_f$ denote the set of cluster labels assigned to the tracklets that cover frame f. For two successive frames $f1$ and $f2$, if $\{Z\}_{f1} = \{Z\}_{f2}$ we say that they belong to the same temporal segment, i.e., $T(f2) = T(f1)$. But if $\{Z\}_{f1} \neq \{Z\}_{f2}$, then we start a new temporal segment, i.e., $T(f2) = T(f1) + 1$. By this process, the frames of the video are partitioned into temporal segments. The cluster labels are supposed to correspond to entities, so each temporal segment should correspond to a shot. Each such segment can be easily represented with any one frame, since all the frames in a segment contain the same entities. This provides us a *shot-based summarization* of the video.

As in the case with entities, once again a large number of temporal segments are created by this process, with several adjacent segments corresponding to the same set of entities. This happens because often several clusters are formed for

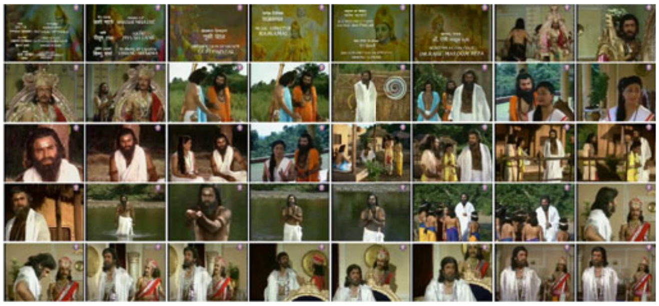

Fig. 12. Shot-based summarization of Mahabharata Episode 22 using TC-CRF, using one frame per significant segment.

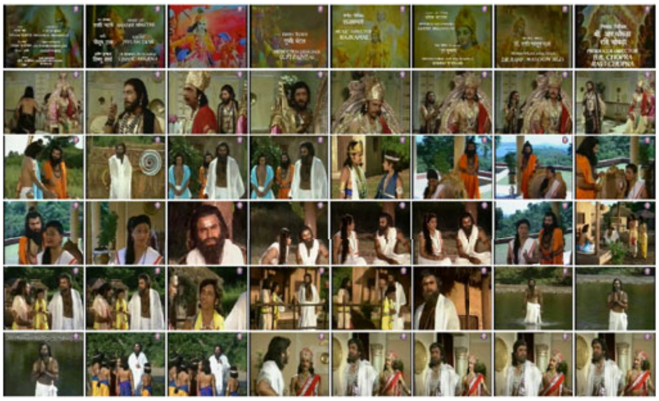

Fig. 13. Shot-based summarization of Mahabharata Episode 22 using SBMR+ConsClus. SBMR+ConsClus creates more significant segments to cover roughly the same set of true shots as TC-CRF, so TC-CRF summary is more concise.

TABLE 11
Conciseness Results for Different Methods for Entity-Based Summarization

Dataset	TCCRF	TCCRP	sHDPHMM	SBMR+ ConsClus	WBSLRR
BBTs1e1	0.13	**0.17**	0.11	0.10	0.09
BBTs1e3	0.18	0.18	0.16	**0.53**	0.16
BBTs1e4	0.15	0.22	**0.28**	0.19	0.20
Maha22	0.21	0.20	0.19	**0.23**	0.18
Maha64	0.12	0.12	0.14	0.11	**0.16**
Maha65	0.18	**0.22**	0.19	0.17	0.20
Maha66	0.17	**0.21**	0.16	0.11	0.14
Maha81	0.23	0.24	0.21	0.17	**0.26**
Maha82	**0.40**	0.38	0.34	0.16	0.25

TABLE 12
Representativeness ($\times 100$) Results for Different Methods for Entity-Based Summarization

Dataset	TCCRF	TCCRP	sHDPHMM	SBMR+ ConsClus	WBSLRR
BBTs1e1	1.7	**1.86**	1.80	0.60	1.60
BBTs1e3	1.72	**2.2**	1.83	0.60	1.23
BBTs1e4	2.3	2.28	**2.69**	0.71	1.51
Maha22	**1.39**	1.30	1.16	0.99	0.87
Maha64	0.82	**0.86**	0.84	0.44	0.76
Maha65	0.99	1.00	**1.02**	0.53	0.81
Maha66	0.97	**1.10**	0.85	0.51	0.80
Maha81	**0.88**	0.85	0.69	0.55	0.64
Maha82	1.46	**1.62**	1.10	0.59	1.00

TABLE 13
Conciseness Results for Different Methods for Shot-Based Summarization

Dataset	TCCRF	TCCRP	sHDPHMM	SBMR+ ConsClus	WBSLRR
BBTs1e1	**0.86**	0.80	0.75	0.74	0.80
BBTs1e3	**0.84**	0.74	0.71	0.82	0.64
BBTs1e4	0.67	0.57	0.55	**0.75**	0.60
Maha22	0.41	0.39	0.40	0.30	**0.53**
Maha64	0.32	**0.34**	**0.34**	0.26	0.27
Maha65	0.30	0.29	0.30	0.24	**0.32**
Maha66	0.14	0.14	0.12	0.11	**0.21**
Maha81	0.37	0.34	0.36	0.24	0.17
Maha82	0.36	0.33	0.38	0.23	**0.41**

TABLE 14
Representativeness ($\times 100$) Results for Different Methods for Shot-Based Summarization

Dataset	TCCRF	TCCRP	sHDPHMM	SBMR+ ConsClus	WBSLRR
BBTs1e1	0.72	0.73	0.75	0.77	**0.80**
BBTs1e3	0.68	0.63	0.64	0.68	**0.74**
BBTs1e4	0.88	**0.93**	0.89	0.75	0.87
Maha22	**0.66**	0.61	0.63	0.53	0.24
Maha64	**0.42**	0.41	0.40	**0.42**	0.23
Maha65	**0.47**	0.43	0.45	0.18	0.26
Maha66	0.43	0.42	0.42	**0.44**	0.10
Maha81	0.45	**0.46**	**0.46**	0.19	0.29
Maha82	**0.59**	**0.59**	0.57	0.38	0.24

the same entity. Analogous to Entity Coverage, we define *Shot Coverage* as the total number of true shots that have at least one temporal segment lying within it. We then define *significant segments* as those which cover a sufficient number (say 100) of frames. Finally, we define *Frame Coverage* as the fraction of the frames which come under the significant segments.

To evaluate such shot-based summarization, once again we need to consider the two basic properties: conciseness and representativeness. These are measured in exact analogy to the entity-based summarization discussed above (See Figs. 12 and 13). The *Conciseness* of the summary is defined as the ratio of the Shot Coverage to the number of significant segments, while the *Representativeness* of the summary is defined as the ratio of the Frame Coverage to the number of significant segments. The results are shown in Tables 13 and 14 . This time we find that in terms of representativeness TC-CRF leads the way, followed by TC-CRP. In terms of conciseness the best performance is given by WBSLRR, which however does poorly in terms of representativeness.

10 Conclusion

In this paper, we considered an entity-driven approach to video modelling. We represented videos as sequences of tracklets, each tracklet associated with an entity. We attempted entity discovery: appearance-modeling the prominent entities in a video, and discovering all their occurences.

We cast this as tracklet clustering and considered a Bayesian nonparametric approach which can automatically discover the number of clusters to be formed. We leveraged the Temporal Coherence property of videos to improve the clustering by our first model: TC-CRP. The second model TC-CRF was a natural extension to TC-CRP, to jointly model short temporal segments within a video, and further improve entity discovery. These methods have several additional abilities like performing online entity discovery efficiently, and detecting false tracklets. We used the discovered entities for semantic video summarization. An extension of this work would be to reduce the number of clusters formed per entity, without compromising on purity.

Acknowledgement

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

References

[1] E. Candes and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, pp. 717–772, 2009.

[2] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[3] R. Keshavan, A. Montanari, and S. Oh, "Matrix completion from a few entries," *IEEE Trans. Inform. Theory*, vol. 56, no. 6, pp. 2980–2998, Jun. 2010.

[4] J. Wright, S. Sastry, Y. Peng, A. Ganesh, and Y. Ma, "Robust principal component analysis," in *Proc. Advances Neural Inform. Process. Syst.*, 2009, pp. 1–9.

[5] X. Ding, L. He, and L. Carin "Bayesian robust principal component analysis," *IEEE Trans. Image Process.* vol. 20, no. 12, pp. 3419–3430, Dec. 2011.

[6] S. D. Babacan, M. Luessi, R. Molina, and A. K. Katsaggelos, "Sparse Bayesian methods for low-rank matrix estimation," *IEEE Trans. Signal Process.*, vol. 60. no. 8, pp. 3964–3977, Aug. 2012.

[7] K. KWagstaff, C. Cardie, S. Rogers, and S. Schrdl, "Constrained k-means clustering with background knowledge," in *Proc. Int. Conf. on Mach. Learning*, 2001, pp. 577–584.

[8] Z. Lu and T. Leen, "Penalized probabilistic clustering,"*Neural Comput.*, vol. 19 no. 6, pp. 1528–1567, 2007.

[9] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[10] S. Tierney, J. Gao, and Y. Yi Guo "Subspace clustering for sequential data," *IEEE Int. Conf. Comput. Vision Pattern Recognition*, 2014, pp. 1019–1026.

[11] X. Wang, B. Qian, and I. Davidson, "On constrained spectral clustering and its applications," *Data Mining Knowl. Discovery*, vol. 28, no. 1, pp. 1–30, 2104.

[12] J. Kawale and D. Boley "Constrained spectral clustering using L1 regularization," in *Proc. SIAM Conf. Data Mining* 2013, pp. 103–111.

[13] S. Xiao, M. Tan, and D. Xu, "Weighted block-sparse low rank representation for face clustering in videos," in *Proc. European Conf. Comput. Vis.*, 2014, pp. 123–138.

[14] T. S. Ferguson "A Bayesian analysis of some nonparametric problems," *Ann. Statist.*, vol. 1, pp. 209–230, 1973.

[15] J. Sethuraman, "A constructive definition of Dirichlet priors," Statistica Sinica, 1994, pp. 639–650.

[16] D. Gorur and C. E. Rasmussen, "Dirichlet process Gaussian mixture models: Choice of the base distribution," *J. Comput. Sci. Tech.*, vol. 25, pp. 653–664, 2010.

[17] Y. W. Teh, M. Jordan, M. Beal, and D. Blei, "Hierarchical Dirichlet processes," *J. Amer. Statist. Assoc.*, vol. 101, pp. 1566–1581, 2006.

[18] E. Fox, E. Sudderth, M. Jordan, and A. Willsky, "An HDP-HMM for systems with state persistence," in *Proc. Int. Conf. Mach. Learn.*, 2008, pp. 312–319.

[19] A. Mitra, B. N. Ranganath, and I. Bhattacharya, "A layered Dirichlet process for hierarchical segmentation of sequential grouped data," in *European Conf. Mach. Learn.*, 2013, pp. 465–482.

[20] D. M. Blei and P. I.Frazier, "Distance dependent Chinese restaurant processes," *J. Mach. Learning Res.*, vol. 12, pp. 2461–2488, 2011.

[21] T. Griffiths and Z. Grahramani, "Infinite latent feature models and the Indian buffet process," *Advances Neural Inform. Process. Syst.*, pp. 475–482, 2005.

[22] S. Williamson, C. Wang, K. Heller, D. Blei, "The IBP compound Dirichlet process and its application to focused topic modeling," in *Proc. Int. Conf. Mach. Learn.* 2010, pp. 1151–1158.

[23] L. Du, W. L. Buntine, and M. Johnson, "Topic segmentation with a structured topic model," in *Proc. North Amer. Assoc. Comput. Linguistics*, 2013, pp. 190–200.

[24] B. Wu, Y. Zhang, B-G. Hu, and Q. Ji, "Constrained clustering and its application to face clustering in videos," in *Proc. IEEE Int. Conf. Comput. Vision Pattern Recognition*, 2013, pp. 3507–3514.

[25] B. Wu, S. Lyu, B-G. Hu, and Q. Ji, "Simultaneous clustering and tracklet linking for multi-face tracking in videos," in *Proc. IEEE Int. Conf. Comput. Vision*, 2013, pp. 2856–2863.

[26] P. Sharma and R. Nevatia, "Efficient detector adaptation for object detection in a video," in *Proc. IEEE Int. Conf. Comput. Vision Pattern Recognition*, 2013, pp. 3254–3261.

[27] K. Tang, V. Ramanathan, Fei-Fei. Li, and D. Koller, "Shifting weights: Adapting object detectors from image to video," in *Proc. Advances Neural Inform. Process Syst.*, 2012, pp. 647–655.

[28] C. Huang, B. Wu, and R. Nevatia, "Robust object tracking by hierarchical association of detection responses," in *Proc. Eur. Conf. Comput. Vision*, 2008, 788–801.

[29] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Int. Conf. Comput. Vision Pattern Recognition*, 2001, pp. I-511–I-518.

[30] H. Ji, C. Liu, Z. Shen, and Y. Xu, "Robust video denoising using low rank matrix completion," in *Proc. IEEE Int. Conf. Comput. Vision Pattern Recognition* 2010, pp. 1791–1798.

[31] J. Wright, Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[32] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "RASL: Robust batch alignment of images by sparse and low-rank decomposition," in *Proc. IEEE Int. Conf. Comput. Vision Pattern Recognition*, 2010, pp. 763–770.

[33] B. Yang, and R. Nevatia, "An online learned CRF model for multi-target tracking," in *Proc. IEEE Int. Conf. Comput. Vision Pattern Recognition*, 2012, pp. 2034–2041.

[34] M. Andriluka, S. Roth, and B. Schiele, "People-tracking-by-detection and people-detection-by-tracking," in *Proc. IEEE Int. Conf. Comput. Vision Pattern Recognition*, 2008, pp. 1–8.

[35] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[36] O. Arandjelovic and R. Cipolla, "Automatic cast listing in feature-length films with anisotropic manifold space," in *Proc. IEEE Int. Conf. Comput. Vision Pattern Recognition*, 2006, pp. 1513–1520.

[37] Y. Zhang, C. Xu, H. Lu, and Y. Huang, "Character identification in feature-length films using global face-name matching," *IEEE Trans. Multimedia*, vol. 11, no. 7, pp. 1276–1288, Nov. 2009.

[38] M. Tapaswi, M. Bauml, and R. Stiefelhagen, "Knock! Knock! Who is it? Probabilistic person identification in TV-series," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, 2012, 2658–2665.

[39] D. Potapov, M. Douze, Z. Harchaoui, C. Schmid, "Category-specific video summarization," in *Proc. 13th Eur. Conf. Comput. Vision*, 2014, pp. 540–555.

[40] W-C Chiu, M. Fritz, "Multi-class video co-segmentation with a generative multi-video model," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, 2013, pp. 321–328.

[41] Y. Cong, J. Yuan, and J. Luo, "Towards scalable summarization of consumer videos via sparse dictionary selection," *IEEE Trans. Multimedia*, vol. 14, no. 1, pp. 66–75, Feb. 2012.

[42] J. Sang and C. Xu, "Character-based movie summarization," in *Proc. 18th ACM Int. Conf. Multimedia*, 2010, pp. 855–858.

[43] X. Xu, T. M. Hospedales, and S. Gong, "Discovery of shared semantic spaces for multi-scene video query and summarization," *arXiv:1507.07458*, 2015.

[44] C. Liang, C. Xu, J. Cheng, and H. Lu, "TVParser: An automatic TV video parsing method," *IEEE Int. Conf. Comput. Vision Pattern Recognition*, 2011, pp. 3377–3384.

[45] (2012). [Online]. Available: https://data.vision.ee.ethz.ch/cvl/youtube-objects/

[46] G. Zhao, J. Yuan, and G. Hua, "Topical video object discovery from keyframes by modeling word co-occurrence prior," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, 2013, pp. 1602–1609.

Adway Mitra received the PhD degree from Department of Computer Science and Automation, Indian Institute of Science, Bangalore. His research interests are in Machine Learning, especially Bayesian modeling of spatio-temporal processes in the context of Computer Vision and Climate Sciences. He has publications in DM/ML conferences like SDM, CIKM, ECML and IJCAI.

Soma Biswas received the PhD degree in computer vision from University of Maryland, College Park. She is an assistant professor in Electrical Engineering Department, Indian Institute of Science, Bangalore. Her research interests lie in Computer Vision, Pattern Recognition, Image Processing and related areas. She has several publications in leading conferences and journals of Computer Vision, like ICCV, CVPR, TPAMI etc.

Chiranjib Bhattacharyya received the PhD degree from the Department of Computer Science and Automation, Indian Institute of Science, Bangalore and post-doctoral experience in University of California, Berkeley. He is a professor in Department of Computer Science and Automation, Indian Institute of Science, Bangalore. His research interests are in Machine Learning and Convex Optimization. He has publications in leading conferences and journals like ICML, NIPS, ICDM, SDM and JMLR.