Temporally Coherent CRP: A Bayesian Non-Parametric Approach for Clustering Tracklets with applications to Person Discovery in Videos

Adway Mitra* Soma Biswas† Chiranjib Bhattacharyya‡

Abstract

Tracklet Clustering is central to several Computer vision tasks [17][20]. A video can be represented as a sequence of tracklets, each spanning over 10-20 successive video frames, and each tracklet is associated with one entity (eg. person in case of TV-serial videos). Tracklets are instances of data-types exhibiting rich spatio-temporal structure. Existing approaches model tracklets by deploying detailed parametric models with a large number of parameters, making the inference unwieldy. The task of *Person Discovery* in long TV-series videos (40-45 minutes) with many persons can be naturally posed as tracklet clustering, and existing approaches give unsatisfactory performance on it. In this paper we attempt to leverage *Temporal Coherence*(TC) of videos to improve tracklet clustering. TC is the fundamental property of videos that each tracklet is likely to be associated with the same entity as its predecessor or successor. We propose the first Bayesian nonparametric approach for modelling TC, which can automatically infer the number of clusters to be formed. The major contribution of this paper is Temporally Coherent Chinese Restaurant Process (TC-CRP), which extends CRP by using TC. On the task of discovering persons in TV serials via tracklet clustering, without meta-data such as scripts, TC-CRP shows up to 25% improvement in cluster purity compared to state-of-the-art parametric models, and upto 36% improvement in number of persons discovered. We use a simple representation of tracklets: a vector of very generic features (like pixel intensity) which can correspond to any type of entity (not necessarily person), and empirically demonstrate the utility of TC-CRP for discovering entities like cars and planes. Moreover, unlike existing approaches TC-CRP can perform online tracklet clustering on streaming videos with very little performance deterioration, and can also automatically reject outliers (tracklets resulting from false detections).

1 Introduction

In this paper we study the problem of Tracklet clustering. Tracklets are formed by detections of an entity (say a person) from a short contiguous sequence of 10-20 video frames. They have complex spatio-temporal properties. Effective clustering of Tracklets can lead to interesting applications in Computer vision [17][20].

The problem of *automated discovery of persons from videos along with all their occurrences* has attracted a lot of interest [27][28][29] in video analytics. This allows users to know the persons appearing in a long video without watching it fully, or to selectively watch those parts which contain a person of interest. Existing attempts try to leverage meta-data such as scripts [28][29] 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 pose this problem as *tracklet clustering*. Our goal is to design algorithms for tracklet clustering which can work on long videos. We should be able to handle any type of entity, not just person. 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 an *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 this problem has not been studied either in Machine Learning or in Computer Vision community.

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* [20] 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

*CSA Department, IISc
†EE Department, IISc
‡CSA Department, IISc

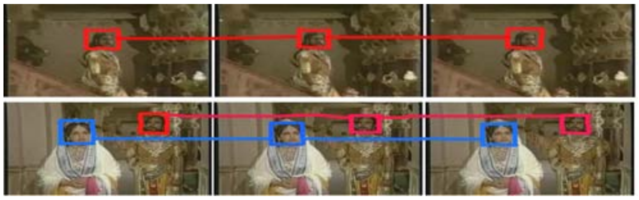

Figure 1: Top: a window consisting of frames 20000,20001,20002, Bottom: another window- with frames 21000,21001,21002. The detections are linked on spatio-temporal basis to form tracklets. One person (marked with red) occurs in both windows, the other (marked with blue) occurs only in the second. The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

Figure 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

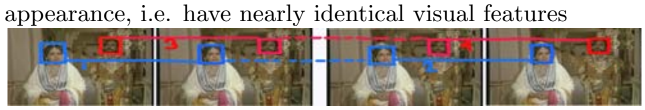

Figure 3: TC at Tracklet level: Blue tracklets 1,2 are spatio-temporally close (connected by broken lines), and belong to same character. Similarly red tracklets 3 and 4.

(see Figure 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 (eg. 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 cover 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 In this paper, 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 long videos, 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 encouraging temporal coherence on the tracklets. Our method yields a superior clustering of tracklets over several baselines especially on long videos. 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. To the best of our knowledge this is the first demonstration of using BNP methodology to model temporal coherence in videos, as well as for tracklet clustering. Finally, the proposed methodology is not application-specific and can be applied to any sequential data where the data-points are represented by vectors, and are temporally coherent at semantic level.

2 Problem Definition

In this section, we elaborate on our task of tracklet clustering for person discovery in videos, and generalize it to entity discovery in sequential data under constraints. We discuss the challenges, and review the related works on Tracklet Clustering, Person Discovery and Constrained Clustering.

2.1 Notation Image-based Object Detectors have become very powerful over the last few years. Entities like human faces or objects like cars, aeroplanes etc can be detected in individual images or video frames by specialized detectors such as [21] [26]. 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 the appropriate detector 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 approprate). Each tracklet i is a collection of R detections $\{I_1^i, \dots, I_R^i\}$. Let the tracklet i be represented by $Y_i = \frac{\sum_{j=1}^{R} I_j^i}{R}$. If the video has missing pixels due to noise, entries of the vectors I_j^i will be missing, in which case the corresponding entries of Y_i are also missing. So finally we have N vectors (N: number of tracklets) possibly with missing entries.

The tracklets can be sorted topologically based on their starting and ending frame indices, so that each tracklet i has a *predecessor tracklet* $prev(i)$ and a *successor tracklet* $next(i)$. Also each tracklet i has a conflicting set of tracklets $F(i)$ which are from frame(s) that overlap with 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 also are represented by vectors, say $\phi_1, \phi_2, \dots, \phi_K$. Each tracklet i is associated with an entity indexed by Z_i, i.e. $Z_i \in \{1, 2, \dots, K\}$.

2.2 Problem Statement Let each video be represented as a sequence of (possibly incomplete) d-dimensional vectors $\{Y_1, \dots, Y_N\}$ along with the set $\{prev(i), next(i), F(i)\}_{i=1}^{N}$. We aim to learn the vectors $\{\phi_1, \phi_2, \dots, \}$ 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 entities that its predecessor or successor are associated with. Moreover, a tracklet i cannot share an entity with its conflicting tracklets $F(i)$, as the same entity cannot occur twice in a same frame. This notion is considered in relevant literature [8] [16]. Mathematically, the constraints are:

$$\begin{aligned} & Z_{prev(i)} = Z_i = Z_{next(i)} \forall i \in \{1, \dots, N\} \\ & Z_i \notin \{Z_j : j \in F(i)\} \forall i \in \{1, \dots, N\} \end{aligned} \tag{2.1}$$

These constraints give the task a flavour of *non-parametric constrained clustering*. An interesting extension is to perform the task *online* i.e. when the datapoints arrive sequentially, and no past datapoint can be accessed once a new one has arrived.

Learning a ϕ_k-vector is equivalent to discovering an entity, and its associated tracklets are discovered by learning the set $\{i : Z(i) = k\}$. Thus the above problem of tracklet clustering can also be viewed upon as the general problem of *discovering entities with all their occurences in temporally coherent* sequential data. An additional extension of this task is to simultaneously reject the *outliers*- datapoints which are significantly different from the rest. In case of tracklet clustering such outliers are the tracklets corresponding to false detections. This particular problem can be easily linked to the task of *discovery of persons and their occurences* in TV-series videos without meta-data such as scripts, and without using any other training data. In this case, the persons can be represented by their face, and a Face Detector like [21] can be used. Discovery of outliers (non-face detections) can help to improve the results for the user, and also help in domain adaptation of the Face Detectors to such videos by serving as negative examples. The online version of the problem can find application in *streaming videos*.

2.3 Challenges The main challenge of tracklet clustering problem lies in handling of the temporal coherence and the conflicts mentioned above. This has been attempted recently in [16] and [8] through Markov Random Fields and Subspace Clustering respectively, though both of these methods involve computations with large matrices and are hence computationally expensive, and suitable only for reasonably short videos. Additionally, these methods need to know the number of clusters to use, which in general not known beforehand. Even if the number of persons in the episode is known (which is often not the case), it is too restrictive to use that number as the number of clusters, since some persons appear in various poses throughout the video and such variations cannot be captured through a single cluster. A better approach is to find the appropriate number of clusters from the data. Finally, none of the existing methods are capable of rejecting outliers and handling streaming videos.

2.4 Related Works Finally, we review the relevant literature. **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* [24], in which several targets are present per frame. A particularly helpful paradigm for multi-target tracking is *tracking by detection* [25], where object-specific detectors like [26] 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* [20] 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 attempts to link tracklets from contiguous frames only, unlike **tracklet clustering**. *It should be understood that tracklet clustering and tracking are different.*

Person Discovery in Videos is another task which has recently received attention in Computer Vision. Cast Listing [27] aims 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 [28] or labelled training videos having the same characters [29]. An unsupervised version of this task is considered in [16], aimed at face clustering in presence of spatio-temporal constraints. They use a Markov Random Field, and encode the constraints as clique potentials. Tracklet association and face clustering are done simultaneously in [17] using HMRF. A recent face clustering approach is WBSLRR [8] where the temporal constraints are encoded in the convex objective function, which is solved by ADMM. However, both [16] and [8] use the detections themselves as datapoints, instead of tracklets, and use track information through the constraints. Both encode the fact that detections in the same tracklet are likely to belong to the same entity, and that two detections in the same tracklet cannot share the same entity. But they do not encode the important observation that spatio-temporally close but non-overlapping tracklets are also likely to share the same entity. Moreover both methods involve large matrix operations, and are hence slow and memory-consuming.

Independent of videos, **Constrained Clustering** is itself a field of research [30]. 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 [31] or soft/probabilistic [32]. Constrained Spectral Clustering has also been studied recently [6] [7], 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 specified 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) [12], 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. However the type of constraints considered here 2.1 have never been studied in a BNP framework.

3 Temporally Coherent Chinese Restaurant Process

Dirichlet Process [9] 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 the *atoms*.

3.1 Modeling Tracklets by Dirichlet Process

We consider H to be a $d-dimensional$ multivariate Gaussian with parameters μ and Σ. The atoms correspond to faces of the persons. The generative process for the set $\{Y_i\}_{i=1}^N$ is then as follows:

$$P \sim DP(\alpha, H); X_i \sim P, Y_i \sim \mathcal{N}(X_i, \Sigma_1) \forall i \in [1, N] \tag{3.2}$$

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

Using the constructive definition of Dirichlet Process, called the Stick-Breaking Process [10], the above process can also be written equivalently as

$$\begin{aligned} &\hat{\pi}_k \sim Beta(1, \alpha), \pi_k = \hat{\pi}_k \prod_{i=1}^{k-1} (1 - \hat{\pi}_{i-1}), \phi_k \sim H \ \forall k \in [1, \infty) \\ &Z_i \sim \pi, Y_i \sim \mathcal{N}(\phi_{Z_i}, \Sigma_1) \forall i \in [1, N] \end{aligned} \tag{3.3}$$

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 persons' faces, 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 3.2 or π in Equation 3.3. The Gibbs Sampling Equations $p(Z_i|Z_{-i}, \{\phi_k\}, Y)$ and $p(\phi_k|\phi_{-k}, Z, Y)$ are given in [11]. For Z_i:

$$p(Z_i = k|Z_{-i}, \phi_k, Y_i) \propto p(Z_i = k|Z_{-i}) p(Y_i|Z_i = k, \phi) \tag{3.4}$$

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.

3.2 Temporal Coherence through Chinese Restaurant Process

In the generative process (Equation 3.3) all the Z_i are drawn IID conditioned on G. 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 3.3 can be redefined as

$$\phi_k \sim H \forall k \in [1,\infty)$$
$$Z_i|Z_1,\ldots,Z_{i-1} \sim CRP(\alpha); Y_i \sim \mathcal{N}(\phi_{Z_i},\Sigma_1) \tag{3.5}$$

The predictive distribution for $Z_i|Z_1,\ldots,Z_{i-1}$ 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** $k \in \{Z_1,\ldots,Z_{i-1}\}; = \frac{\alpha}{N-1+\alpha}$ **otherwise**

where N_k^i is the number of times the value k is taken in the set $\{Z_1,\ldots,Z_{i-1}\}$.

We now modify CRP to handle the Spatio-temporal cues mentioned earlier. To model TC, we use $prev(i)$ for each tracklet i, as defined in Section 2.1. In the generative process, we define $p(Z_i|Z_1,\ldots,Z_{i-1})$ with respect to $prev(i)$, similar to the Block Exchangeable Mixture Model as defined in [13]. 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 data-points (tracklets) arise due to false (non-face) 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:

$$\begin{aligned} T(Z_i = k|Z_{1:i-1}, C_{1:i-1}, C_i = 1) &= 0 \textbf{ if } k \in \{Z_{F(i)}\} - \{0\} \\ &\propto \beta \textbf{ if } k = 0 \\ &\propto n_{k1}^{ZC} \textbf{ if } k \in \{Z_1,\ldots,Z_{i-1}\}, k \notin \{Z_{F(i)}\} \\ &\propto \alpha \textbf{ otherwise} \end{aligned} \tag{3.6}$$

where $Z_F(i)$ is the set of values of Z for the set of tracklets $F(i)$ 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 non-face 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 3.1.

```
φ_k ~ N(μ, Σ) ∀k ∈ [1, ∞)
for i = 1 : N do
    if dist(i, prev(i)) ≤ thres then
        C_i ~ Ber(κ1)
    else
        C_i ~ Ber(κ2)
    end if
    if C_i = 1 then
        draw Z_i ~ T(Z_i|Z_1, ..., Z_{i-1}, C_1, ..., C_{i-1}, α)
    else
        Z_i = Z_prev(i)
    end if
    if Z_i = 0 then
        Y_i ~ N(μ, Σ_2)
    else
        Y_i ~ N(φ_{Z_i}, Σ_1)
    end if
end for
```

where T is the PPF for TC-CRP, defined in Eq 3.6.

3.3 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 characteristics: 1) Change Variable 2) Spatio-temporal cues 3) Separate component for false/outlier tracklets. The concept of change variable C_i was used in Block-exchangeable Mixture Model [13], 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 [12], but this model does not consider the change variable C_i. Moreover, it uses a transition distribution P_k for each mixture component k, which increases the model complexity. Like BEMM [13] we avoid this step, and hence our PPF (Eq 3.6) does not involve $Z_{prev(i)}$. DDCRP [14] defines distances between every pair of datapoints, and associates a new data-point 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_k^Z: the number of previous datapoints assigned to component k. Hence our method is significantly different from DDCRP. The first two rules of TC-CRP PPF are novel.

3.4 Inference Inference in TC-CRP can once again be performed 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 [13]. If $C_{next(i)} = 0$ and $Z_{next(i)} \neq Z_{prev(i)}$, then we must have $C_i = 1$ and $Z_i = Z_{next(i)}$. If $C_{next(i)} = 0$ and $Z_{next(i)} = Z_i$, then $Z_i = Z_{next(i)}$, and C_i is sampled from $Bernoulli(\kappa)$. In case $C_{next(i)} = 1$ and

$Z_{next(i)} \neq Z_{i-1}$, 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 $Z_{prev(i)} = k$, 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 0.2 respectively (based on empirical experience on a test video). Similarly we fix the value of *thres* empirically.

3.5 Completion of Missing Entries Finally we consider the case where the tracklet representations Y_i have missing entries, Let Y_{Ω_i} be the observed part of Y_i. In that case, the generative process of this vector will be $Y_{\Omega_i} \sim \mathcal{N}(\phi_{Z_i\Omega_i}, \sigma_1^2 I)$, where $\phi_{Z_i\Omega_i}$ is the projection of ϕ_{Z_i} to the dimensions Ω_i. Here we use isotropic Gaussians, $\Sigma = \sigma^2 I$ and $\Sigma_1 = \sigma_1^2 I$, so that we can compute the posterior mean *independently for each dimension*. Similarly, during the learning of ϕ_k, only the observed parts $\{Y_{\Omega_i} : Z_i = k\}$ are used. Let Ω denote the set of observed entries. Then, for dimension d, the posterior mean of ϕ_{kd} is given by $\frac{\frac{Y_{kd}}{\sigma_1^2} + \frac{\mu}{\sigma^2}}{\frac{n_{kd}}{\sigma_1^2} + \frac{1}{\sigma^2}}$, where $n_{kd} = |\{i : Z_i = k, (i, d) \in \Omega\}|$, and $Y_{kd} = \sum_{i:Z_i=k,(i,d)\in\Omega} Y_{id}$.

3.6 Online Inference In 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 3.6). Instead of drawing one sample per data-point, an option is to draw several samples and consider the mode.

4 Experimental Validation

We carried out extensive experiments on 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 characters (occurring in at least 50 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 have 15-25 prominent characters (occurring in at least 100 frames). These videos are much longer than those studied in similar works like [17], 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 [16][17], we represent the characters 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.

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 5 cars of different colors, while the aeroplanes video had 6 planes of different colors/shapes. These were created by concatenating shots of different cars/planes in the Youtube Objects datasets [15]. The objects were detected using the Object-specific detectors [26]. 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 2700-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. The dataset details are given in Table 1. [1]

4.1 Alternative Methods A recent method for face clustering using track information is WBSLRR [8] based on Subspace Clustering. Though in [8] it is used for clustering detections rather than tracklets, the change can be made easily. Apart from that, we can use Constrained Clustering as a baseline, and we choose a recent method [7]. 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 [2] [22]. 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) [4]. Others are either too slow (BRPCA [3]), or recover matrices with ranks too low (OPTSPACE [1]) or too high (RPCA [2]). Finally, we compare against another well-known BNP model for sequential data- the sticky HDP-HMM [12].

[1] The appendix, code and data are available at `http://clweb.csa.iisc.ernet.in/adway`

4.2 Performance 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 (5 in case of Cars/Aeroplanes). 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 cluster k is "pure" if at least* 70% *of the tracklets assigned to it belong to any one entity* A(say Sheldon for a BBT video, or Arjuna for a Mahabharata video, or the silvery car for the Cars video). We also declare that the cluster k and its corresponding mixture component ϕ_k corresponds to the entity A. Also, then A is considered to be *discovered*. The threshold of purity was set to 70% 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. 4, 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 entities with at least 1 pure significant cluster corresponding to it.* Next, we measure *Tracklet Coverage: the fraction of tracklets that are assigned to pure significant clusters.* Effectively, these tracklets are *discovered*, and the remaining ones (in short/impure clusters) are lost.

4.3 Results The results on the three measures discussed above are shown in Tables 2,3,4. In terms of the three measures, TC-CRP is usually the most accurate, followed by sHDPHMM. 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. However on the Cars video it does not produce any large impure cluster, and hence returns the best performance. Curiously, WBSLRR is found to be quite competent on the TV-series videos but not on the Car and Aeroplane videos, perhaps because of their high dimensionality (2700 instead of 900) and relatively few tracklets.

It may be noted that the *number of significant clusters formed* is a matter of concern, especially from the user' 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, as discussed in Section 2 (See Figure 6). The number of clusters formed per video by the different methods

Dataset	#Frames	#Detections	#Tracklets	#Entities	Entity Type
BBTs1e1	32248	25523	2408	7	Person(Face)
BBTs1e3	31067	21555	1985	9	Person(Face)
BBTs1e4	28929	20819	1921	8	Person(Face)
Maha22	66338	37445	3114	14	Person(Face)
Maha64	72657	65079	5623	16	Person(Face)
Maha65	68943	53468	4647	22	Person(Face)
Maha66	87202	76908	6893	17	Person(Face)
Maha81	78555	62755	5436	22	Person(Face)
Maha82	86153	52310	4262	24	Person(Face)
cars	750	694	694	5	Car
aeroplanes	750	939	939	6	Aeroplane

Table 1: Details of datasets

Figure 4: Face detections (top), and the corresponding atoms (reshaped to square images) found by TC-CRP (bottom)

is indicated in Table 2. It appears that none of the methods have any clear advantage over the others in this regard. 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-CRP and sHDP-HMM did not change significantly. On the other hand, the matrix returned by SBMR had higher rank (120-130 for $R = 1$) as the number of tracklets increased. Regarding *running time*, TC-CRP was fastest, and converged faster than the more complex SHDPHMM. WBSLRR and constrained clustering involved matrix operations and were much slower.

4.4 Online Inference / Performance on Streaming Videos We wanted to explore the case of streaming videos, where the frames appear sequentially and old frames are not stored. In the absence of actual streaming datasets we performed the single-pass inference (Sec 3.6) 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 Sec 4.1 are incapable in the online setting, and sticky HDP-HMM is the only alternative. The results are presented in Table 5, which show TC-CRP to be doing distinctly better. Notably, the figures for TC-CRP in the online experiment are not significantly lower than those in the offline experiment, unlike sHDP-HMM. This indicates that TC-CRP converges quicker, and so is more efficient offline.

Figure 5: Car detections (top), and the corresponding atoms found by TC-CRP (bottom)

Figure 6: Different atoms for different poses of same person

Dataset	TCCRP	sHDPHMM	SBMR+ ConsClus	WBSLRR
BBTs1e1	0.75 (36)	**0.84** (44)	0.67 (48)	0.73 (45)
BBTs1e3	**0.83** (40)	0.76 (37)	0.80 (15)	0.67 (43)
BBTs1e4	**0.89** (36)	0.83 (29)	0.77 (31)	0.71 (41)
Maha22	0.87 (69)	0.86 (74)	**0.94** (44)	0.83 (79)
Maha64	**0.92** (105)	0.91 (97)	0.85 (88)	0.75 (81)
Maha65	0.89 (85)	**0.90** (89)	0.86 (76)	0.82 (84)
Maha66	**0.96** (73)	0.95 (80)	0.87 (84)	0.81 (81)
Maha81	**0.89** (88)	0.84 (95)	0.87 (84)	0.74 (78)
Maha82	**0.88** (50)	0.86 (58)	0.78 (63)	0.83 (64)
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)

Table 2: Purity results for different methods. The number of significant clusters are written in brackets

Dataset	TCCRP	sHDPHMM	SBMR+ ConsClus	WBSLRR
BBTs1e1	**6**	5	5	4
BBTs1e3	**7**	6	**8**	**7**
BBTs1e4	**8**	**8**	6	**8**
Maha22	**14**	**14**	10	**14**
Maha64	13	**14**	11	13
Maha65	**19**	17	13	17
Maha66	**15**	13	9	11
Maha81	**21**	20	14	20
Maha82	19	**20**	10	16
Cars	**5**	**5**	**5**	2
Aeroplanes	**6**	5	**6**	4

Table 3: Entity Coverage results for different methods

Dataset	TCCRP	sHDPHMM	SBMR+ ConsClus	WBSLRR
BBTs1e1	0.67	**0.79**	0.29	0.73
BBTs1e3	**0.88**	0.68	0.09	0.53
BBTs1e4	**0.82**	0.78	0.22	0.62
Maha22	**0.90**	0.86	0.43	0.69
Maha64	**0.90**	0.81	0.39	0.62
Maha65	0.85	**0.91**	0.40	0.68
Maha66	**0.80**	0.68	0.43	0.65
Maha81	**0.75**	0.66	0.46	0.50
Maha82	**0.81**	0.64	0.37	0.64
Cars	0.73	0.69	**1.00**	0.04
Aeroplanes	**0.93**	0.70	0.88	0.09

Table 4: Tracklet Coverage results for different methods

Dataset	Maha65		Maha81	
Measure	TC-CRP	sHDPHMM	TC-CRP	sHDPHMM
Purity	**0.89**(79)	0.84 (82)	**0.84**(74)	0.70(57)
Entity Coverage	15	**16**	**21**	17
Tracklet Coverage	**0.80**	0.77	**0.62**	0.49
Dataset	BBTs1e1		BBTs1e4	
Measure	TC-CRP	sHDPHMM	TC-CRP	sHDPHMM
Purity	**0.73**(33)	0.50 (14)	**0.88**(32)	0.75(28)
Entity Coverage	3	3	6	**7**
Tracklet Coverage	**0.65**	0.40	**0.81**	0.67

Table 5: Online (single-pass) analysis on 4 videos

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

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

Table 6: Discovery of non-face tracklets

4.5 Outlier Detection / Discovery of Non-Face Tracklets Face Detectors such as [21] are trained on static images, and applied on the videos on per-frame basis. This approach itself has its challenges [18], 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 [19].

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 6 (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 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*.

5 Conclusion

In this paper we proposed TC-CRP: a Bayesian Nonparametric route to model temporal coherence in videos. We showed its application in tracklet association, for the task of discovery of entities and their tracks in videos without using any additional information. Our method is capable of identifying tracklets that result from false detections, and this may be helpful in adapting pretrained detectors to videos by providing negative examples, which is an active area of research. It can also perform online tracklet clustering on streaming videos without significant deterioration in performance.

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

References

[1] Keshavan,R., Montanari,A. & Oh,S. (2010), *Matrix Completion from a few entries*, IEEE Transactions on Information Theory

[2] Wright,J., Sastry,S., Peng,Y., Ganesh,A. & Ma, Y. (2009), *Robust Principal Component Analysis*, Advances in Neural Information Processing Systems (NIPS)

[3] Ding, X., He, L. & Carin, L. (2009), *Bayesian Robust Principal Component Analysis*, IEEE Transactions on Image Processing

[4] Babacan,S.D., Luessi,M., Molina,R. & Katsaggelos,A.K. (2012), *Sparse Bayesian Methods for Low-Rank Matrix Estimation*, IEEE Transactions on Signal Processing

[5] Shi,J. & Malik,J. (2000), *Normalized cuts and image segmentation*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), Vol 22(8)

[6] Wang,X., Qian,B. & Davidson,I. (2014), *On Constrained Spectral Clustering and its Applications*, Data Mining and Knowledge Discovery, Vol 28(1)

[7] Kawale,J. & Boley,D. (2013,) *Constrained Spectral Clustering using L1 Regularization*, SIAM Conference on Data Mining (SDM)

[8] Xiao1,S., Tan,M., & Xu,D. (2014), *Weighted Block-Sparse Low Rank Representation for Face Clustering in Videos*, European Conference on Computer Vision (ECCV)

[9] T. S. Ferguson (1973), *A Bayesian Analysis of some Nonparametric Problems*, The Annals of Statistics.

[10] Sethuraman,J. (1991), *A Constructive Definition of Dirichlet Priors*

[11] Gorur,D. & Rasmussen,C.E. (2010), *Dirichlet process Gaussian mixture models: Choice of the base distribution*, Journal of Computer Science and Technology

[12] Fox,E., Sudderth,E., Jordan,M. & Willsky,A. (2008), *An HDP-HMM for Systems with State Persistence*, Intl. Conf. on Machine Learning (ICML)

[13] Mitra,A., Ranganath,B.N., & Bhattacharya,I. (2013), *A Layered Dirichlet Process for Hierarchical Segmentation of Sequential Grouped Data*, European Conference on Machine Learning (ECML-PKDD)

[14] Blei,D.M. & Frazier,P.I. (2011), *Distance Dependent Chinese Restaurant Processes*, Journal of Machine Learning Research, Vol 12

[15] `http://people.ee.ethz.ch/presta/youtube-objects/website/youtube-objects.html`

[16] Wu,B., Zhang,Y., Hu,B-G & Ji,Q. (2013), *Constrained Clustering and Its Application to Face Clustering in Videos*. IEEE Intl. Conf on Computer Vision and Pattern Recognition (CVPR)

[17] Wu,B., Lyu,S., Hu,B-G & Ji,Q. (2013), *Simultaneous Clustering and Tracklet Linking for Multi-Face Tracking in Videos*, IEEE Intl. Conf on Computer Vision (ICCV)

[18] Sharma,P. & Nevatia,R. (2013), *Efficient Detector Adaptation for Object Detection in a Video*, IEEE Intl. Conf on Computer Vision and Pattern Recognition (CVPR)

[19] Tang,K., Ramanathan,V., Li,Fei-Fei & Koller,D. (2012), *Shifting Weights: Adapting Object Detectors from Image to Video*, Advances in Neural Information Processing Systems (NIPS)

[20] Huang,C., Wu,B., & Nevatia,R. (2008), *Robust object tracking by hierarchical association of detection responses*, European Conf on Computer Vision (ECCV)

[21] Viola,P. & Jones,M. (2001), *Rapid object detection using a boosted cascade of simple features*, IEEE Intl. Conf on Computer Vision and Pattern Recognition (CVPR)

[22] Ji,H., Liu,C., Shen,Z. & Xu,Y. (2010), *Robust Video Denoising using Low Rank Matrix Completion*, IEEE Intl. Conf on Computer Vision and Pattern Recognition (CVPR)

[23] Wright,J., Yang,Y., Ganesh,A., Sastry,S.S., & Ma,Y. (2009), *Robust Face Recognition via Sparse Representation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 31(2)

[24] Yang,B. & Nevatia,R. (2012), *An Online Learned CRF Model for Multi-Target Tracking*, IEEE Intl Conf Computer Vision and Pattern Recognition (CVPR)

[25] Andriluka,M., Roth,S. & Schiele,B. (2008),*People-tracking-by-detection and people-detection-by-tracking*, IEEE Intl Conf. Computer Vision and Pattern Recognition (CVPR)

[26] Felzenszwalb,P.F., Girshick,R.B., McAllester,D. & Ramanan,D. (2010), *Object Detection with Discriminatively Trained Part-based Models*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 32(9)

[27] Arandjelovic,O. & Cipolla,R. (2006), *Automatic Cast Listing in feature-length films with Anisotropic Manifold Space*, IEEE Intl Conf. Computer Vision and Pattern Recognition (CVPR)

[28] Zhang,Y., Xu,C., Lu,H. & Huang,Y. (2009), *Character Identification in Feature-length Films using Global Face-name Matching* IEEE Transactions on Multimedia Vol 11(7)

[29] Tapaswi,M., Bauml,M. & Stiefelhagen,R. (2008),: *Knock! Knock! Who is it? Probabilistic Person Identification in TV-Series*, IEEE Conf. Computer Vision and Pattern Recognition (CVPR)

[30] `http://www.cs.albany.edu/~davidson/Publications/KDDSlides.pdf`

[31] KWagstaff,K., Cardie,C., Rogers,S. & Schrdl,S. (2001), *Constrained k-means clustering with background knowledge*, Intl Conf. on Machine Learning (ICML)

[32] Lu,Z. & Leen,T. (2007), *Penalized Probabilistic Clustering*, Neural Computation Vol. 19(6)