Abstract—A tracklet is a short sequence of detections of an entity of interest, such as a person's face, in contiguous frames of a video. In this paper we consider clustering tracklets in long videos, an important problem of current interest in Computer Vision. It involves identifying tracklets in short segments of the video and finding associations between tracklets over the entire video. These tasks require careful modelling of spatio-temporal cues. State of the art [27] proposes a probabilistic model which incorporates the spatio-temporal cues by parametrizing them through clique potentials in HMRF. The attendant learning and inference problems make the model un-wieldy resulting in failure to handle long videos with many cluster labels. In this paper we circumvent the problem of explicitly encoding spatio-temporal cues by exploiting Temporal Coherence (TC). The major contribution of the paper is to develop Temporally Coherent Chinese Restaurant Process (TC-CRP), a novel Bayesian Non-parametric (BNP) prior which models Temporal Coherence. To the best of our knowledge this is the first work which models TC in a BNP framework and thus makes a significant addition to BNP priors suited for video analytics. On an interesting problem of discovering persons and their tracklets from user-uploaded videos of long TV series episodes from Youtube we show that TC-CRP is very effective. It can also filter out tracklets resulting from false detections. We explore alternative approaches based on low-rank matrix recovery and constrained subspace clustering, but find these to be very slow and less accurate than our method. Finally, we show that TC-CRP can be useful in Low-rank Matrix Recovery when the desired matrix has sets of identical columns.

I. INTRODUCTION

Online video-hosting sites like Youtube are full of videos uploaded by the users. Many of these videos are movies or TV serial episodes, shot directly from the television. Such videos often suffer from low resolution/noise, based on the quality of TV transmission and camera. Also, they do not have associated meta-data like scripts. Users of the websites may want to know the actors/characters appearing in a video without watching the full video (since they can be quite long). Also, they may want to watch only those parts of the video which involve a certain character. This motivates the task of automatically discovering the persons in a video, and also the tracks associated with each person. In this work, we represent persons by their faces, as often done in relevant literature [23] [24].

A track is formed by detecting faces [23] in each video frame, and associating detections across a sequence of frames based on appearance and spatio-temporal locality. In case of TV series episodes, each track corresponds to a particular person. 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 to form tracklets [27] 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 our task, the same person may appear in different (non-continuous) parts of the video, and so we need to associate tracklets on a long-range basis also (see Figure 1). Moreover, on most internet videos of movies/TV series, the task is complicated by the low resolution/noise, and lots of false face detections which act as spoilers.

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 (i.e. appearance of a person) are almost unchanged across a tracklet. At tracklet-level it implies that spatio-temporally close (but non-overlapping) tracklets are likely to have same label (i.e. belong to the same person). (See Fig 3 for illustration) Additionally, there is one more constraint: overlapping tracklets (that cover the same frames), cannot belong to the same person. Because of TC at detection-level, a tracklet can be easily represented, unlike a long track where the face appearance may gradually change due to facial expressions/poses.

Most video analysis techniques exploit TC in some way or
In this paper, we propose a Bayesian nonparametric approach to model it at tracklet-level. Following the Computer Vision literature [5] [30] [23] we represent each face detection as a d-dimensional vector, by rescaling and reshaping the rectangular detections. Exploiting TC at detection-level and feature-level, we represent each tracklet with the mean of the detection vectors associated with it. We view each tracklet as a data-point that is generated from one of an unknown number of mixture components from a Dirichlet Process. Each component corresponds to a person. Dirichlet Process helps in the long-range association of tracklets, based on appearance. To capture the short-term associations which involves the spatio-temporal cues stated above, we propose a variant- TC-CRP (Temporally Coherent Chinese Restaurant Process). Our approach has several advantages over existing methods. First of all it does not need to know the number of persons. Secondly, it can handle noise in the form of missing pixels in the tracklets [29], and automatically filter out several non-face tracklets. Moreover, although Bayesian Nonparametric methods are considered slow, our formulation involves a Change Variable which considerably speeds up inference [21]. As a result it can handle videos with longer duration and more persons than the state-of-the-art method [24] for simultaneous face clustering and tracklet linking which uses Hidden Markov Random Fields. This method requires the number of persons to be specified beforehand, which is not possible in arbitrary internet videos. Also, it fails to model or utilize TC, and assigns labels to a subset of the detections instead of the tracklets. The tracklets are used to provide constraints on label assignments to pairs of detections, which are encoded as clique potentials. It takes a long time to encode all these constraints, and the inference involves cumbersome matrix computations and is incapacitated by numerical issues when the number of tracklets and persons is even slightly high.

Some papers like [5] [30] represent a video as a matrix where each column represents a single face detection or a single frame. Here we can a matrix where each column corresponds to a tracklet, and so sets of columns corresponding to same persons should be very similar. Also, if the tracklets are arranged in temporal order, adjacent tracklets are likely to belong to the same person, and hence nearly identical. This suggests that we can approximate the matrix with a low-rank one with sets of identical columns, where adjacent columns are likely to be identical. However we find that existing low-rank matrix recovery methods are unable to handle this structural information. The low-rank matrices recovered by them must be further processed by clustering of the columns, using the rank as the number of clusters. A similar approach is subspace clustering, which computes a pairwise affinity matrix for the columns, followed by spectral clustering. In this case we can use Constrained Spectral Clustering [15] [16] for both, by coding the spatio-temporal cues as must-link and don’t-link constraints. However, these techniques are much slower than the proposed TC-CRP, and also give significantly inferior results. Our proposed method can be considered as a new solution to the task of low-rank matrix recovery, where there is such structural information on the columns. Here the rank of the matrix is modelled using Dirichlet Process which is a novel idea in the low-rank matrix recovery domain.

**Contribution** In this paper, we attack the novel problem of discovering entities (like persons) and all their tracklets from any video, in absence of any side information. We propose an efficient and scalable solution based on Chinese Restaurant Process- the first Bayesian Nonparametric model for Temporal Coherence in videos, that is also able to automatically filter out non-face detections, and does not need the number of persons in advance. Moreover we relate the problem to the Machine Learning problem of low-rank matrix recovery where the desired matrix has sets of identical columns, and show that TC-CRP is more effective than existing methods.

The paper is organized as follows. We discuss the relevant literature in Section 2. In Section 3 we formalize the notations and define the problem, and also relate it to low-rank matrix recovery and subspace clustering. In Section 4 we describe our model along with inference procedure. Experimental verification is described in Section 5. Finally in Section 6 we discuss how the proposed method can benefit Low-rank Matrix Recovery in the context of Computer Vision.

**II. RELATED WORK**

**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 [32], in which several targets are present per frame. A tracking paradigm that is particularly helpful in multi-target tracking is tracking by detection [33], where object-specific detectors like [34] 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 [27] 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.

**Person Discovery in Videos** is another task which has recently received attention in Computer Vision. Cast Listing [35] 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 or labelled training videos having the same characters. An unsupervised version of this task is considered in [23], 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. Tracklet association and face clustering are done simultaneously in [24] using HMRF.

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. A detailed survey is found in [38]. The constraints can be hard or soft/probabilistic [40]. Constrained Spectral Clustering has also been studied recently [15] [16], 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 (shHDP-HMM) [20], 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) [22], which defines distances between every pair of datapoints, and each point is linked to another with probability proportional to such distances.

We have already noted that tracklet association can be cast as low-rank matrix recovery. Low-rank matrix recovery has two aspects: completion and extraction. The completion problem is that, there is an unknown low-rank matrix \( M \) for which only a subset \( \Omega \) of the entries are known, and we have to recover the remaining entries. The problem is stated as \( \min_X rank(X) \text{ such that } X_{\Omega} = M_{\Omega} \). The extraction problem states that, there is a fully observed matrix \( X \) which is the sum of a low-rank matrix and a sparse matrix, which we need to find. Since \( rank \) is convex, Convex Optimization-based methods [1] [2] [3] model \( rank(X) \) as nuclear norm \( \|X\|_* \) - the \( \ell_1 \) norm of the singular values of \( X \). The convex optimization problem is then solved by Semidefinite Programming [1] or iterative thresholding [2] [5]. The paper [4] considers the case where a small subset of columns are arbitrarily corrupted, and aims to identify the corrupted columns in addition to completing the missing entries in the non-corrupted columns. This method also uses nuclear norm to model rank, and the matrix (2,1)-norm to model the column-sparse corrupting matrix. Apart from convex optimization, some Bayesian approaches [6] [9] also proceed by putting priors on the singular values. A rare non-spectral approach is considered in [7], which expresses low-rank matrix \( X \) as a product of \( A \) and \( B \), and put sparse priors on the columns of \( A \) and \( B \). They perform inference by variational techniques.

### III. Problem Definition, Notations and Approaches

Given a video, a face-detector [28] is run on every frame. The detections in successive frames are then linked based on spatial locality, to obtain tracklets [27]. At most \( R \) detections are linked like this. The tracklets of length less than \( R \) are discarded, hence all tracklets consist of \( R \) detections. All \( R \) detections in a tracklet are expected to correspond to the same person, and their visual representations are almost identical due to temporal coherence. We restrict the length of tracklets so that the appearance of the detected faces remain almost unchanged, which facilitates tracklet representation.

We represent a person’s face by a vector of dimension \( d \). This can be done by downsampling a rectangular face detection to \( d \times d \) square and then reshaping it to a \( d^2 \)-dimensional vector. Let vectors \( \phi_1, \phi_2, \ldots, \phi_K \) represent the persons’ faces. Each tracklet \( i \) is associated with a person \( Z_i \), and is a collection of \( R \) detections \( I_1^i, \ldots, I_R^i \), which are all \( d \)-dimensional vectors very similar to \( \phi_{Z_i} \). Let the tracklet be represented by \( Y_i = f(I_1^i, \ldots, I_R^i) \) for some function \( f \), like the mean. If the video has missing pixels due to noise, entries of the vectors \( I^i \) will be missing, in which case the corresponding entries of \( Y_i \) are also missing. So finally we have \( N \) vectors \( Y_i \) possibly with missing entries.

We aim to learn the vectors \( \phi_1, \ldots, \phi_K \) and the indices \( Z_i \) corresponding to each tracklet \( i \). Learning a \( \phi_k \)-vector is equivalent to discovering a person, and its associated tracklets are discovered by learning the set \( \{ i : Z(i) = k \} \).

#### A. Temporal Coherence through Dirichlet Process

We consider a Gaussian mixture with means \( \{\phi_1, \phi_2, \ldots, \phi_K\} \), and model every tracklet vector as \( Y_i \sim N(\phi_{Z_i}, \sigma^2 I) \). The tracklet vectors of each person are slightly different from each other due to slight variations of pose, illumination, noise etc, and such variations are captured through this Gaussian distribution. As \( K \) is unknown, we consider an Infinite Mixture Model like Dirichlet Process Mixture Model [17]. In case of a DP-distributed distribution \( P \), for \( N \) draws, the number of distinct values will be \( O(\log(N)) \), i.e. much lower than the number of draws. This ties well with the fact that there are many tracklets corresponding to the same person. The mixture model is useful for long-range association of tracklets, based on appearance/visual features only. In addition we need a way to capture the short-term associations, and the constraints described earlier- 1) Spatio-Temporally close Tracklets should be assigned to same mixture component, 2) Overlapping tracklets cannot be assigned to same component. For this purpose, we develop our model: Temporally Coherent Chinese Restaurant Process (TC-CRP). We explain this model in details in Section IV.
Ideally, the number of components $K$ learnt by TC-CRP should be equal to the number of persons (one component per person). In reality, in a TV series episode the same person can appear in multiple poses and facial expressions, and it is really difficult to capture such variations with a single mixture component (unless very robust face-specific features are used). So generally several components are expected to be created for most of the persons.

B. Low-rank Matrix Recovery

Apart from the Bayesian model, we also consider alternative approaches. Clearly, we can view the tracklets $\{Y_1, Y_2, \ldots, Y_N\}$ as columns of an incomplete matrix $Y$, and we want to approximate $Y$ with a matrix $X$, each of whose columns is from the set $\{\phi_1, \phi_2, \ldots, \phi_K\}$. Clearly, matrix $X$ is of rank $K$, and this problem is one of low-rank matrix recovery. If we can recover $X$, then we will be able to associate the tracklets accordingly. Additionally, we have false (non-face) detections, some of which get linked to form junk tracklets. Typically the vectors corresponding to such tracklets are quite different from the vectors $\{\phi_1, \ldots, \phi_K\}$, and can be considered as arbitrarily corrupted columns $A$. The task they considered was to identify the corrupted columns, which is equivalent to our goal of filtering out the non-face tracklets.

It turns out that low-rank matrix recovery $Y_{LR}$ based on existing methods like [4] [11] [7] [5] etc cannot provide a matrix which has its columns from a small set of vectors like $\{\phi_k\}$. It is necessary to further cluster the columns, maintaining the spatio-temporal cues. This can be done through Constrained Spectral Clustering [13] [16], using the spatio-temporal cues as must-link and don’t-link constraints. The rank $\text{rank}(Y_{LR})$ is used as the number of clusters to be formed. We use Constrained Spectral Clustering instead of other Constrained Clustering methods like [40] because spectral clustering can handle any arbitrary similarity measure, which is important as our tracklet vectors may have missing entries.

C. Subspace Clustering

It has been noted in recent Face Recognition literature [30] that a face vector of a person can be expressed as a sparse linear combination of the other face vectors in the dataset (particularly of the faces belonging to the same person). Since we represent each tracklet as the mean of the associated face vectors, it can also be considered that each tracklet vector $Y_i$ can be expressed as a sparse linear combination of the other tracklet vectors. Consider all the tracklet vectors $\{Y_i\}$ corresponding to a particular person $k$. As already noted, due to significant appearance variations it may take several $\phi_k$-vectors to cover these. This may be re-interpreted as, these tracklet vectors span an Union of Subspaces, in which each individual vector in the set lies. Our task of learning $Z_i$-values can now be cast as Subspace Clustering. For this there are methods like Low Rank Representation [12] and Sparse Subspace Clustering [11]. These methods learn a Coefficient Matrix $C$ where $Y = YC + B$ ($B$: sparse noise matrix), and use it to compute an affinity matrix $W$ between pairs of tracklet vectors. This matrix is then used for Spectral Clustering like Normalized Cuts [13]. In our case, we should once again use Constrained Spectral Clustering methods like [15] [16], to handle the spatio-temporal cues.

IV. Temporally Coherent Chinese Restaurant Process

Dirichlet Process [17] 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 $Dir(\alpha H(A_1), \ldots, \alpha H(A_K))$, where $\alpha$ 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.

A. Modeling Tracklets by Dirichlet Process

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

$$P \sim DP(\alpha, H); \ X_i \sim P, Y_i \sim N(X_i, \Sigma) \forall i \in [1, N]$$

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 modelled using $N(X_i, \Sigma_i)$.

Using the constructive definition of Dirichlet Process, called the Stick-Breaking Process [18], the above process can also be written equivalently as

$$\pi_k \sim \text{Beta}(1, \alpha), \ \pi_k \bigwedge_{i=1}^{k-1} (1 - \pi_{i-1}), \ \phi_k \sim H \ \forall k \in [1, \infty)$$

$$Z_i \sim \pi, Y_i \sim N(\phi_{z_i}, \Sigma_i) \forall i \in [1, N]$$

Here, $\pi$ 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 $\phi_k$, 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 (1) or $G$ in Equation (2). The Gibbs Sampling Equations $p(Z_i|Z_{-i}, \{\phi_k\}, Y)$ and $p(\phi_k|\phi_{-k}, Z, Y)$ are given in [19]. For $Z_i$, the equation is

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

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

B. Temporal Coherence through Chinese Restaurant Process

In the generative process (Equation (2)) 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 \( \pi \) and \( Z_{i-1} \).

In case of DP, the independence among \( Z_i \)'s is lost on integrating out \( \pi \). After integration the generative process of Eq \( \ref{eq:1} \) can be redefined as

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

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

\[
p(Z_i = k|Z_{i-1}) = \frac{N_i^k}{N_i + \alpha} \quad \text{if} \quad k \in \{Z_1, \ldots, Z_{i-1}\} = \frac{N_i^k}{N_i + \alpha} \quad \text{otherwise}
\]

where \( N_i^k \) 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 in the previous section. To model cue \( \mathbf{1} \), we define \( prev(i) \) for each tracklet \( i \). For this purpose tracklets are ordered sequentially according to their starting/ending frames with ties resolved arbitrarily. For any tracklet \( i \), \( prev(i) \) is the tracklet \( j \) that preceeded \( i \) according to this order. 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 \( \cite{21} \). 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 (\( \kappa_1 \) and \( \kappa_2 \)) 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 (non-face) detections. We need a way to model these. Since these are very different from the Base mean \( \mu \), we consider a separate component \( Z = 0 \) with mean \( \mu \) and a very large covariance \( \Sigma_2 \), which can account for such variations. The Predictive Probability function (PPF) for TC-CRP is defined as follows:

\[
T(Z_i = k|Z_{i-1}, C_{i-1}, C_i = 0) = \begin{cases} 1, & \text{if} \quad k \in \{Z_F(i)\} - \{0\} \\ \propto \beta, & \text{if} \quad k = 0 \\ \propto n_{k1}^{2C}, & \text{if} \quad k \in \{Z_1, \ldots, Z_{i-1}\}, k \notin \{Z_F(i)\} \\ \propto \alpha, & \text{otherwise} \end{cases}
\]

where \( Z_F(i) \) is the set of values of \( Z \) for the set of tracklets \( F(i) \) that overlap with \( i \), and \( n_{k1}^{2C} \) 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:

where \( T \) is the PPF for TC-CRP, defined in Eq \( \ref{eq:5} \).

\[
\begin{align*}
1: \phi_k & \sim \mathcal{N}(\mu, \Sigma) \quad \forall k \in [1, \infty) \\
2: \text{for} \ i = 1 : N \text{ do} \\
3: \quad \text{if} \ dist(i, prev(i)) \leq \text{thres} \text{ then} \\
4: \quad \quad C_i \sim \text{Ber}(\kappa_1) \\
5: \quad \text{else} \ \\
6: \quad \quad C_i \sim \text{Ber}(\kappa_2) \\
7: \end{if} \\
8: \quad \text{if} \ C_i = 1 \text{ then} \\
9: \quad \quad \text{draw} \ Z_i \sim T(Z_i|Z_1, \ldots, Z_{i-1}, C_1, \ldots, C_{i-1}, \alpha) \\
10: \quad \text{else} \\
11: \quad \quad Z_i = Z_{prev(i)} \\
12: \end{if} \\
13: \quad \text{if} \ Z_i = 0 \text{ then} \\
14: \quad \quad Y_i \sim \mathcal{N}(\mu, \Sigma_2) \\
15: \quad \text{else} \\
16: \quad \quad Y_i \sim \mathcal{N}(\phi_{Z_i}, \Sigma_1) \\
17: \end{if} \\
18: \text{end for}
\end{align*}
\]

C. 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) Changepoint 2) Spatio-temporal cues 3) Separate component for non-face tracklets. The concept of changepoint was used in Block-exchangeable Mixture Model \( \cite{21} \), 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 \( \cite{20} \), but this model does not consider changepoints. Moreover, it uses a transition distribution \( P_t \) for each mixture component \( k \), which increases the model complexity. Like BEMM \( \cite{21} \), we avoid this step, and hence our PPF (Eq \( \ref{eq:5} \)) does not involve \( Z_{prev(i)} \). DDCRP \( \cite{24} \) 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 \( n_k^2 \): 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.

D. Inference

Inference in TC-CRP can once again be performed through Gibbs Sampling. We need to infer \( C_i, Z_i \), and \( \phi_k \). As \( C_i \) and \( Z_i \) are coupled, we sample them in a block for each \( i \in [1, N] \) as done in \( \cite{21} \). If \( C_{i+1} = 0 \) and \( Z_{i+1} \neq Z_i \), 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 \( \text{Bernoulli}(\kappa) \). 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 \( p(C_i = a)p(Z_i|Z_{i-1}, C_i = a) \).
strained Clustering. There are many low-rank matrix recovery techniques available. We found that among these the best performance is given by Sparse Bayesian Matrix Recovery (SBMR) \cite{7}. Others are either too slow (BRPCA \cite{6}), or recover matrices with ranks too low (OPTSPACE \cite{3}) or too high (RPCA \cite{5}). For constrained Spectral Clustering we use the recent method \cite{16}. For subspace clustering, we use Low Rank Representation \cite{12}, since the other recent alternative Sparse Subspace Clustering \cite{11} is very slow. For all of these methods, we use the implementations provided by the respective authors. Finally, we compare against another well-known sequential Bayesian nonparametric method- the sticky HDP-HMM \cite{20}.

B. Performance Measures

The task of person discovery with all their tracks is novel and complex, and has to be judged by suitable measures. It can be looked upon as a tracklet clustering problem and judged by clustering accuracy. However, the videos are large, and frame-level or tracklet-level ground truth labels are difficult to annotate. Instead, we judge the clusters themselves. We discard the clusters that have less than 10 assigned tracklets. It turns out that the selected clusters cover about 95\% of all the tracklets. There are some clusters which have mostly (70\% or more) non-face tracklets. We discard these from our evaluation.

We say that a cluster $k$ is "pure" if at least 70\% of the tracklets assigned to it belong to any one person $A$, and declare that the cluster $k$ and its corresponding atom $\phi_k$ corresponds to the person $A$. 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 an atom or center corresponding to a cluster is visually recognizable as the person (after reshaping to $d \times d$). We measure the Purity: fraction of the clusters that are pure, i.e. correspond to some person. We also measure Person Coverage: the number of persons with at least 1 cluster (at least 10 tracklets) corresponding to it. Next, we measure Tracklet Coverage: the fraction of tracklets that are assigned to pure clusters. Effectively, these tracklets are discovered, and the remaining ones (assigned to impure clusters) are lost. Finally, we measure the time taken by various methods.

C. Results

The results on the four measures discussed above are shown in Tables II,III,IV,V. We find that the state-of-the-art method \cite{24} based on HMRF runs into severe numerical issues on these large videos, due to the matrix computations involved. In fact it fails to yield valid results when the number of clusters to be formed is more than 10. All of our videos have more than 10 characters. On running them with number of characters set to 10, the clusters formed are all impure and arbitrary.

We find that in terms of time, the proposed TC-CRP is always the best. It is significantly (often around 100\%) faster than second-best sHDP-HMM, due to the change variable, as noted in \cite{21}. The other two methods are far slower, due to spectral clustering. In case of subspace clustering by LRR, the
affinity matrix computation is itself slow. In terms of the other three measures, TC-CRP is usually the most accurate, followed by sHDPHMM. The spectral clustering-based methods are competitive on the purity measure, but fare very poorly in terms of tracklet coverage. This is because, they form many small pure clusters, and a few very large impure clusters which cover about half the tracklets. Thus, a large number of tracklets are lost.

It may be noted that the number of 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 character, but that is not possible due to the significant appearance variations, as discussed in Section III (See Figure 5). The number of clusters formed per video by the different methods is indicated in Table II. 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 as the number of tracklets increased (due to shortening). For \( R = 1 \), the matrix-based methods created 120-130 clusters for all the videos. The codes and links to the videos are provided at [https://www.dropbox.com/sh/nqp9wy8sln84msw/AA8k4xKJc4Sisw8H-T1li0kHa](https://www.dropbox.com/sh/nqp9wy8sln84msw/AA8k4xKJc4Sisw8H-T1li0kHa)

D. Recovery of Missing Pixels

User-generated videos are often noisy and grainy, as they are often shot directly from the television. The quality of the camera can also be an issue. Such videos may have random pixels grossly corrupted, i.e., effectively missing. We find that if more than 20% of the pixels are missing at random, the face detector itself often fails, and hence the person and tracklet discovery will not work. So we test the performance of our method with 20% pixels missing at random. As already discussed in Section IV-E, TC-CRP can recover missing entries in the tracklet vectors. As benchmark, we consider Low-rank Matrix Completion methods like SBMR [7] and OPTSPACE [3]. However, SBMR is found to run out of memory, and OPTSPACE produces matrices with very low rank (5 or 6), which is clearly unrealistic as the number of persons are much more. In contrast, TC-CRP’s performance remains similar to those already reported in Tables II, III, IV, V.

E. Discovery of Non-Face Tracklets

Face Detectors such as [28] are trained on static images, and applied on the videos on per-frame basis. This approach itself has its challenges [25], and the complex videos we consider in our experiments do not help matters. As a result, there

| Dataset          | TCCRP | sHDPHM | SBMR+ConsClus | LRR+ConsClus |
|------------------|-------|--------|---------------|--------------|
| Maha22           | 0.97  | 0.86   | 0.94          | 0.87         |
| Maha64           | 0.92  | 0.91   | 0.85          | 0.91         |
| Maha65           | 0.89  | 0.90   | 0.86          | 0.96         |
| Maha66           | 0.96  | 0.95   | 0.87          | 0.92         |
| Maha81           | 0.89  | 0.84   | 0.87          | 0.88         |
| Maha82           | 0.88  | 0.86   | 0.78          | 0.75         |

### Table II

PURITY RESULTS FOR DIFFERENT METHODS. HMRF IS DROPPED, AS MENTIONED IN SECTION IV-C. THE NUMBER OF SIGNIFICANT CLUSTERS FOUND ARE WRITTEN IN BRACKETS

| Dataset          | TCCRP | sHDPHM | SBMR+ConsClus | LRR+ConsClus |
|------------------|-------|--------|---------------|--------------|
| Maha22           | 14    | 14     | 10            | 11           |
| Maha64           | 13    | 14     | 11            | 13           |
| Maha65           | 19    | 17     | 13            | 12           |
| Maha66           | 15    | 13     | 9             | 11           |
| Maha81           | 21    | 20     | 14            | 15           |
| Maha82           | 19    | 20     | 10            | 11           |

### Table III

PERSON COVERAGE RESULTS FOR DIFFERENT METHODS. HMRF IS DROPPED, AS MENTIONED IN SECTION IV-C

| Dataset          | TCCRP | sHDPHM | SBMR+ConsClus | LRR+ConsClus |
|------------------|-------|--------|---------------|--------------|
| Maha22           | 0.90  | 0.8648 | 0.4271        | 0.4640       |
| Maha64           | 0.9027| 0.8115 | 0.3923        | 0.3836       |
| Maha65           | 0.8494| 0.9083 | 0.3962        | 0.3551       |
| Maha66           | 0.8044| 0.6772 | 0.4293        | 0.4566       |
| Maha81           | 0.7520| 0.6599 | 0.4595        | 0.4765       |
| Maha82           | 0.8123| 0.6394 | 0.3693        | 0.3895       |

### Table IV

TRACKLET COVERAGE RESULTS FOR DIFFERENT METHODS. HMRF IS DROPPED, AS MENTIONED IN SECTION IV-C

| Dataset          | TCCRP | sHDPHM | SBMR+ConsClus | LRR+ConsClus |
|------------------|-------|--------|---------------|--------------|
| Maha22           | 314   | 527    | 1356          | 4941         |
| Maha64           | 568   | 900    | 6313          | 8733         |
| Maha65           | 420   | 877    | 3224          | 6352         |
| Maha66           | 436   | 832    | 12082         | 16362        |
| Maha81           | 604   | 1017   | 5541          | 6501         |
| Maha82           | 240   | 523    | 2358          | 7826         |

### Table V

TIME TAKEN (IN SECONDS) BY DIFFERENT METHODS. HMRF IS DROPPED, AS MENTIONED IN SECTION IV-C

---

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

**Fig. 5.** Different learnt atoms that correspond to different poses of the same person
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 \cite{25}.

Identifying the non-face detections is not easy, especially since they have been classified wrongly as faces with a well-known and time-tested detector. We make use of the fact that these are relatively less in number (compared to the true detections), and hence at least some of them can be expected to deviate widely from the mean of the detection vectors. This is taken care of in the TC-CRP tracklet model, through the component $\phi_0$ that has very high variance, and hence is most likely to generate the unusual tracklets. We set this variance $\Sigma_2$ 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 Mahal81 (randomly chosen). 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. Using much higher values of $K$ does form clusters which contain only non-face tracklets, but there is no way to identify them (based on size), and so the process cannot be automated. In Table VI we report the results for the smallest cluster formed for $K = 10$. 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 majorly outperforms clustering on both precision and recall*.

**VI. RELATION TO OTHER PROBLEMS**

In this section, we explore the scope of the proposed model beyond tracklet association.

**A. Low-rank Matrices with sets of identical Columns**

Low-rank matrices are quite commonly used in computer vision. They have been used for both still images \cite{31} and for videos \cite{5}. In case of still images, each column generally corresponds to the face of a person. In case of videos, it corresponds to a frame (for background subtraction), a subwindow in a frame (for denoising) or detector outputs from a frame (in this work). In all cases, the low-rank matrix approximation is considered because several columns of the data matrix are near-identical. For example, in case of background subtraction all columns of the low-rank matrix should be identical, as they correspond to the static background. In TV series or movie videos, due to the property of Temporal Coherence, successive frames generally contain the same entities (for example, persons), and changes occur only at shot boundaries. Hence, if we represent a video with a matrix where each column corresponds to an entity detection (arranged in order of track/frame indices), we can expect to find a low-rank approximation where successive columns to be identical except at the shot/track boundaries. We investigate if the low-rank matrices recovered by various methods for matrix recovery (completion and extraction) actually do have successive columns identical.

The existing methods mostly proceed by regularizing the nuclear norm, i.e. shrinking smaller singular values to 0. This reduces the rank and entry-wise error, but does not necessarily capture the structural property on the columns.

**Synthetic Matrices** We generate 50 basis vectors $\{\phi_k\}_{k=1}^{50}$ by sampling from the standard multivariate spherical Gaussian. Next, each column is generated by drawing from a basis vector from a multinomial distribution. In one version, all columns are drawn IID from this distribution (no temporal coherence).

In another version each column is drawn from a multinomial that emphasizes on the previous draw. In particular, if the column $X_i$ corresponds to basis vector $\phi_k$, then for column $X_{i+1}$ we sample $\phi_k$ with probability 0.9, and any of the basis vectors uniformly with probability 0.002, and thus temporal coherence exists. These columns constitute the original matrix $X_{original}$. We study matrices of dimensions $(200 \times 1000)$, as in most applications the number of datapoints is far larger than the dimension. We study the sensitivity of the methods to the fraction of missing values. We try various levels of incompleteness, and vary the fractions of missing entries from 0.1 to 0.7. The matrices are corrupted by additive zero-mean noise with variance 0.1 independently on the observed entries.

**Video Face Matrix** Next, we consider a small matrix $Y$ of face detections (reshaped to 900-dimensional vectors), taken from a user-uploaded Youtube video. $Y$ has 1000 columns. Due to temporal coherence of videos, successive frames contain the same character, except at the shot change points. However, between the change points the face vectors are near-identical. A set of detections from this video are shown in Figure 7. The matrix $Y$ has rank 900, because of
small movements and variations in noise levels across the frames. However, noting that there are only 3 characters and 12 change-points, between which the vectors are almost identical, it is expected that a low-rank approximation $X$ of $Y$ should clearly have rank at most 12. Also, between these change-points, the columns of $X$ should be identical.

**Rank-column Plot:** We consider the quantity $\bar{X}_i = \text{rank}(X_{1:i})$—the rank of the sub-matrix formed by the first $i$ columns of $X$. If $X$ has identical columns between the change-points, this quantity should remain fixed between these change-points, and may increase by 1 only at the change-points. Hence the plot of $\bar{X}_i$ versus $i$ should be a step-function, as shown in Figure 7. We call this plot as the Rank-column plot of $X$. We study the rank-column plot(Figure 8) of the estimated low-rank matrix $X$ returned by three recent methods for low-rank matrix approximation: Robust PCA [5], Bayesian Robust PCA [6] and Sparse Bayesian Robust PCA [7]. Surprisingly for all three methods, we observe: 1) The rank-column plot for none of the methods comes close to the expected step function. All three show similar plots: the rank rises monotonically and then flattens out. 2) For all three methods, the estimated “low-rank” matrix has rank much higher than the number of characters, and even the number of shot-change-points. Moreover, if the estimated matrix had rank $r$, then the sub-matrix formed by any set of $r$ columns had rank equal to $\min(r,m)$. Such behavior of the rank-column plot clearly shows that the existing low-rank matrix recovery methods are completely incapable of capturing the temporal coherence of videos.

**TC-CRP for Matrix Recovery** A better idea is to use a discrete distribution on the columns, where each column vector is chosen from a set of vectors. This is very similar to the TC-CRP model proposed in this paper. It models temporal coherence through the change variable that ensures successive columns to be identical, but if not desired, this property can be abolished by setting the Bernoulli parameter $\kappa$ to 1 (i.e. $C_i = \{Y_i\}$). Note that at any column $i$ the rank $\bar{X}_i$ increases from $\bar{X}_{i-1}$ if a new vector (different from $X_{1:i-1}$) is sampled. The value $\alpha$ in the PPF (Equation 5) regulates the probability of sampling of a new vector from the base distribution, so a smaller value of $\alpha$ ensures a lower rank.

**Evaluation:** We evaluate TC-CRP’s performance against the existing methods, for both the synthetic matrices (with and without TC) and the video face matrix. We measure the Frobenius norm error (FE) $||X_{\text{recovered}} - X_{\text{original}}||_F$, the rank error (RE) $\frac{\text{rank}(X_{\text{recovered}} - X_{\text{original}})}{\text{rank}(X_{\text{original}})}$. Also, as the original matrices have sets of identical columns, and the ones recovered by TC-CRP also have the same property, we compute the RAND index to evaluate the matching. As none of the existing low-rank recovery methods provide a matrix with identical columns, we compare TC-CRP’s clustering against Spectral Clustering [13], which requires a similarity matrix between pairs of datapoints. We define pairwise similarity $S(i,j) = \exp(-||X_i - X_j||_F^2)$ ($\Omega_i$: set of observed entries of $X_i$), and try out different values of $K$. The results for synthetic data are shown Tables VII and VIII. The rank-column plots are provided in Figure 8 for synthetic data and Figure 9 for faces. We see that on the synthetic data, not only does TC-CRP provide the perfect rank-column plots (which coincide with the true plots), but even in terms of Frobenius norm error, Rank error and RAND index, its performance is way ahead of the existing methods. For the face data also, its rank-column plot is roughly accurate, and increments around the shot change-points. The rank is 13, and the steps reasonably match with the shot change-points.

### B. Subspace Clustering

A problem related to low-rank matrix completion is Subspace Clustering, where each column vector of the data matrix $Y$ is considered to lie in an Union of Subspaces. The data matrix is expressed as $Y = YC + B$, where $C$ is the coefficient.

| Table VII: Comparison of Low-rank Matrix Completion Techniques with Varying Fractions of Missing Entries, in Absence of TC. | FE: Frobenius Norm Error, RE: Rank Error, RAND: Rand Index for Clustering |
|---|---|---|---|---|---|---|---|
| Method | FE | RE | RAND | Method | FE | RE | RAND |
| TC-CRP | 0.015 | 0.025 | 0.035 | RAND | 0.035 | 0.045 | 0.055 |
| RPCA | 0.020 | 0.030 | 0.040 | RAND | 0.040 | 0.050 | 0.060 |
| BRPCA | 0.025 | 0.035 | 0.045 | RAND | 0.045 | 0.055 | 0.065 |

| Table VIII: Comparison of Low-rank Matrix Completion Techniques with Varying Fractions of Missing Entries, in Presence of TC. | FE: Frobenius Norm Error, RE: Rank Error, RAND: Rand Index for Clustering |
|---|---|---|---|---|---|---|
| Method | FE | RE | RAND | Method | FE | RE | RAND |
| TC-CRP | 0.010 | 0.020 | 0.030 | RAND | 0.020 | 0.030 | 0.040 |
| RPCA | 0.015 | 0.030 | 0.040 | RAND | 0.030 | 0.040 | 0.050 |
| BRPCA | 0.020 | 0.035 | 0.045 | RAND | 0.040 | 0.050 | 0.060 |

Fig. 8. Rank-column plots for various methods. Left figure is for a matrix with 10% missing entries, and right figure for 50% missing entries. The Blue Line (True Plot) and the Black Line (proposed method) coincide.

Fig. 9. Left: Rank-column plots for SBMR(blue), RPCA(red) and BRPCA(green) for the test video. The estimated matrices all have rank much more than the number of shot segments (12), and do not exhibit the expected step function-like behavior. Right: Rank-column plot for TC-CRP(blue), and the shot number(red) which increments at the shot change-points. The rank is 13, and the steps reasonably match with the shot change-points.
matrix where the column-vector $C_i$ indicates the subspace membership of column $Y_i$ (the $i$-th datapoint). This representation has also been used in Computer Vision, most notably for motion segmentation. Once again, as several datapoints are almost identical, their corresponding coefficient vectors are also expected to be similar, and hence the coefficient matrix $C$ should again have sets of identical columns. No wonder, $C$ has been modelled as low-rank in the LRR formulation [12]. Also, in case of sequential data like videos, the successive datapoints are very similar and likely to have some subspace coefficients, which is handled in [14] by an additional penalty term. However, these methods also model the rank of $C$ using the nuclear norm, and as we have seen already, this cannot recover a coefficient matrix with sets of identical columns, as needed for clustering. Hence, they perform the clustering as a separate step, using spectral clustering with an affinity matrix constructed from $C$. However, spectral clustering is very slow.

A better option can be to use TC-CRP on the recovered $C$ to obtain its approximation having sets of identical columns. Sequential data can easily be taken care of, because TC-CRP models temporal coherence. In case of non-sequential data also, TC-CRP (with $\kappa = 1$) can produce sets of identical columns because it generates the columns from a discrete distribution. Note that this does not obviate the need to estimate the coefficient matrix $C$. This is because, TC-CRP in its current form models each datapoint $Z_i$ as a draw from a single Gaussian, and not as the linear combination of several such draws, which is required for this purpose.

VII. 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 characters and their tracks in videos without using any additional information. The formulation is not specific to faces, and can be applied to handle other entities also. Our method is capable of identifying tracklets that result from false detections, and this may be helpful in adapting pre-trained detectors to videos by providing negative examples, which is an active area of research. Further, we showed that TC-CRP is more appropriate than existing low-rank recovery methods in recovering low-rank matrices with sets of identical columns, as frequently needed in Computer Vision. Future work will involve developing TC-CRP to handle the general problems of matrix completion and subspace clustering, rather than restricting it to matrices having sets of identical columns.

REFERENCES

[1] Candes, E. & Recht, B. (2009) Exact Matrix Completion via Convex Optimization, Foundations of Computational Mathematics
[2] Cai, J.-F., Candés, E. J. and Shen, Z. (2010) A singular value thresholding algorithm for matrix completion, SIAM Journal of Optimization, Vol 20(4)
[3] Keshavan, R., Montanari, A. & Oh, S. (2010) Matrix Completion from a few entries, IEEE Transactions on Information Theory
[4] Chen, Y., Xu, H., Caramanis, C. & Sanghavi, S. (2011) Robust Matrix Completion with Corrupted Columns, Advances in Neural Information Processing Systems (NIPS)
[5] Wright, J., Sastry, S., Peng, Y., Ganesh, A. & Ma, Y. (2009) Robust Principal Component Analysis, Advances in Neural Information Processing Systems (NIPS)
[6] Ding, X., He, L. & Carin, L. Bayesian Robust Principal Component Analysis, IEEE Transactions on Image Processing
[7] Babacan, S.D., Luessi, M., Molina, R. & Katsaggelos, A.K. (2012) Sparse Bayesian Methods for Low-Rank Matrix Estimation, IEEE Transactions on Signal Processing
[8] Nie, F., Wang, H., Cai, X., Huang, H. & Ding, C. (2012) Robust Matrix Completion via Joint Schatten $p$-Norm and $\ell_p$-Norm Minimization, Intl. Conf. on Data Mining (ICDM)
[9] Todeschini, A., Caron, F. & Chavent, M. (2013) Probabilistic Low-Rank Matrix Completion with Adaptive Spectral Regularization Algorithms, Advances in Neural Information Processing Systems (NIPS)
[10] Liu, G., Lin, Z. & Yu, Y. (2010) Robust Subspace Segmentation by Low-Rank Representation, International Conference on Machine Learning (ICML)
[11] Elhamifar, E. & Vidal, R. (2009) Sparse Subspace Clustering, IEEE Intl. Conf on Computer Vision and Pattern Recognition (CVPR)
[12] Liu, G., Lin, Z., Yan, S., Sun, J., Yu, Y. & Ma, Y. (2013) Robust recovery of subspace structures by low-rank representation, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), Vol 35(1)
[13] Shi, J. & Malik, J. (2000) Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), Vol 22(8)
[14] Tierney, S., Gao, J. & Yi, Y. (2014) Subspace Clustering for Sequential Data, IEEE Intl. Conf on Computer Vision and Pattern Recognition (CVPR)
[15] Wang, X., Qian, B. & Davidson, L. (2014) On Constrained Spectral Clustering and its Applications, Data Mining and Knowledge Discovery, Vol 28(1)
[16] Kawale, J. & Boley, D. (2013) Constrained Spectral Clustering using $L_1$ Regularization, SIAM Conference on Data Mining (SDM)
T. S. Ferguson (1973), *A Bayesian Analysis of some Nonparametric Problems*, The Annals of Statistics.

Sethuraman, J. (1991) *A Constructive Definition of Dirichlet Priors*, The Annals of Statistics.

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

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

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)

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

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)

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)

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

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)

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

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)

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)

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)

Peng, Y., Ganesh, A., Wright, J., Xu, W. & Ma, Y. (2010) *RASL: Robust Batch Alignment of Images by Sparse and Low-Rank Decomposition*, IEEE Intl Conf Computer Vision and Pattern Recognition (CVPR)

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

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)

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)

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

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)

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

http://www.cs.albany.edu/~davidson/Publications/KDDSlides.pdf

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

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