BAYESIAN NONPARAMETRIC MODELING OF TEMPORAL COHERENCE FOR ENTITY-DRIVEN VIDEO ANALYTICS

A Thesis

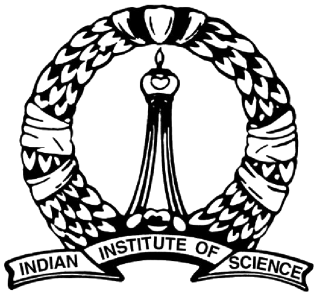
Submitted for the Degree of

Doctor of Philosophy

in the **Faculty of Engineering**

by

ADWAY MITRA



Computer Science and Automation
Indian Institute of Science
Bangalore – 560 012 (INDIA)

JULY 2015

TO

India,
The World
and
Humanity

Acknowledgements

First of all, I will thank my parents- Dr Parthasarathi Mitra and Mrs Ranjana Mitra, imparting in me the desire to pursue academics. Also I must thank them for the encouragement they provided me in joining a PhD program, and the moral support they provided during the darkest hours and deepest nadirs of my research life. It is because of such support that I survived those times and finally reached the goal.

During my research, important roles were played by my advisor, labmates, peers and collaborators. Discussions with Dr. Chiranjib Bhattacharyya and Dr Soma Biswas of EE Department helped me to improve my skill of writing research papers. With Anoop and Ujwal of CVAI Lab, EE Department, I worked on my first research problems in IISc, and survived numerous disappointments and hardships before being able to complete the problems. In my own lab, I had the chance to work with Mrinal Das, from whom I picked up many intuitions on Bayesian Nonparametrics. With Ranganath I worked for a long time on the most intellectually stimulation problem of my thesis, under Dr. Indrajit Bhattacharya. Academic discussions with Himabindu, Goutham, Lavanya, Trapit and Raman also helped in development of my understanding of Machine Learning.

I have been fortunate enough to be advised and mentored by a number of capable researchers of Machine Learning. I will forever cherish the superb intellectually stimulating technical discussions I had with Dr. Indrajit Bhattacharya, which greatly helped in the development of my maturity in research, and particularly in Bayesian Nonparametrics. During my internships in Yahoo! (2011) and in IBM-IRL (2012)-I was fortunate to be mentored by Dr. Rajeev Rastogi and Dr. Srujana Merugu respectively. The Yahoo! internship was of particular significance to me, as it was during this period that I had my maiden first-hand experience of cutting-edge research by top scientists. This exposure helped me in defining the idea for my thesis, and considerably improving my experimental skills. I was also fortunate to interact with Dr. Prateek Jain of Microsoft Research on a problem.

The best asset I had during my PhD life were my friends- both in IISc and outside. Through Gmail and Facebook, as well as on the dining tables of B and D messes, we discussed anything

from intricate philosophical matters through sports to gossips. I would like to convey my love and heartfelt thanks to Swagato Sanyal, Sayan Bhattacharya, Anshuman Dutt, Prerna Priya, Dripto Bakshi, Himabindu Lakkaraju, Souri Datta, Neeraja Yadwadkar, Shweta Jain, Arijita Dutta, Mayuresh Kunjir, Mangesh Pujari, Priyanka Singla, Akanksha Agrawal, Anirban Laha, Nisha Singhania, Priyanka Tyagi, Amita Malik, Amrita Panja, Anasua Deb, Pradeep Bansal, Rashmi Balasubramanyam, Ankit Jauhari, Jagdeep Pani, Swapnil Soni, Snehal Mandlik, Pankaj Jain, Rutuja Chitra-Tarak and many many others.

Finally, I will like to thank the IISc campus itself- with all its wonderful people and its amazing natural beauty, that helped to shape me as a responsible and idealist human being. Also it helped me to interact with people from all over India, and thus grow beyond my regional outlook to acquire a pan-Indian one.

Abstract

In recent times there has been an explosion of online user-generated video content. This has generated significant research interest in video analytics. Human users understand videos based on high-level semantic concepts. However, most of the current research in video analytics are driven by low-level features and descriptors, which often lack semantic interpretation. Existing attempts in semantic video analytics are specialized and require additional resources like movie scripts, which are not available for most user-generated videos. There are no general purpose approaches to understanding videos through semantic concepts.

In this thesis we attempt to bridge this gap. We view videos as collections of *entities* which are semantic visual concepts like the persons in a movie, or cars in a F1 race video. We focus on two fundamental tasks in Video Understanding, namely summarization and scene-discovery. Entity-driven Video Summarization and Entity-driven Scene discovery are important open problems. They are challenging due to the spatio-temporal nature of videos, and also due to lack of apriori information about entities. We use Bayesian nonparametric methods to solve these problems. In the absence of external resources like scripts we utilize fundamental structural properties like *temporal coherence* in videos- which means that adjacent frames should contain the same set of entities and have similar visual features. There have been no focussed attempts to model this important property. This thesis makes several contributions in Computer Vision and Bayesian nonparametrics by addressing Entity-driven Video Understanding through temporal coherence modeling.

Temporal Coherence in videos is observed across its frames at the level of features/descriptors, as also at semantic level. We start with an attempt to model TC at the level of features/descriptors. A tracklet is a spatio-temporal fragment of a video- a set of spatial regions in a short sequence (5-20) of consecutive frames, each of which enclose a particular entity. We attempt to find a representation of tracklets to aid tracking of entities. We explore region descriptors like Covariance Matrices of spatial features in individual frames. Due to temporal coherence, such matrices from corresponding spatial regions in successive frames have nearly identical eigenvectors. We utilize this property to model a tracklet using a covariance matrix, and use it for region-based

entity tracking. We propose a new method to estimate such a matrix. Our method is found to be much more efficient and effective than alternative covariance-based methods for entity tracking.

Next, we move to modeling temporal coherence at a semantic level, with special emphasis on videos of movies and TV-series episodes. Each tracklet is associated with an entity (say a particular person). Spatio-temporally close but non-overlapping tracklets are likely to belong to the same entity, while tracklets that overlap in time can never belong to the same entity. Our aim is to cluster the tracklets based on the entities associated with them, with the goal of discovering the entities in a video along with all their occurrences. We argue that Bayesian Nonparametrics is the most convenient way for this task. We propose a temporally coherent version of Chinese Restaurant Process (TC-CRP) that can encode such constraints easily, and results in discovery of pure clusters of tracklets, and also filter out tracklets resulting from false detections. TC-CRP shows excellent performance on person discovery from TV-series videos. We also discuss semantic video summarization, based on entity discovery.

Next, we consider entity-driven temporal segmentation of a video into scenes, where each scene is characterized by the entities present in it. This is a novel appplication, as existing work on temporal segmentation have focussed on low-level features of frames, rather than entities. We propose EntScene: a generative model for videos based on entities and scenes, and propose an inference algorithm based on Blocked Gibbs Sampling, for simultaneous entity discovery and scene discovery. We compare it to alternative inference algorithms, and show significant improvements in terms of segmentation and scene discovery.

Video representation by low-rank matrix has gained popularity recently, and has been used for various tasks in Computer Vision. In such a representation, each column corresponds to a frame or a single detection. Such matrices are likely to have contiguous sets of identical columns due to temporal coherence, and hence they should be low-rank. However, we discover that none of the existing low-rank matrix recovery algorithms are able to preserve such structures. We study regularizers to encourage these structures for low-rank matrix recovery through convex optimization, but note that TC-CRP-like Bayesian modeling is better for enforcing them.

We then focus our attention on modeling temporal coherence in hierarchically grouped sequential data, such as word-tokens grouped into sentences, paragraphs, documents etc in a text corpus. We attempt Bayesian modeling for such data, with application to multi-layer segmentation. We first make a detailed study of existing models for such data. We present a taxonomy for such models called Degree-of-Sharing (DoS), based on how various mixture components are shared by the groups of data in these models. We come up with Layered Dirichlet Process which generalizes Hierarchical Dirichlet Process to multiple layers, and can

also handle sequential information easily through Markovian approach. This is applied to hierarchical co-segmentation of a set of news transcripts- into broad categories (like politics, sports etc) and individual stories. We also propose a explicit-duration (semi-Markov) approach for this purpose, and provide an efficient inference algorithm for this. We also discuss generative processes for distribution matrices, where each column is a probability distribution. For this we discuss an application: to infer the correct answers to questions on online answering forums from opinions provided by different users.

Publications from the Thesis

1. Adway Mita, Anoop K.R., Ujwal Bonde, Chiranjib Bhattacharyya, K. R. Ramakrishnan; **Eigenprofiles of Spatio-Temporal Fragments for adaptive Region-based Tracking**; *International Conference of Accoustics, Speech and Signal Processing (ICASSP), 2012*

2. Adway Mitra, Soma Biswas, Chiranjib Bhattacharyya; **Temporally Coherent Chinese Restaurant Process for Tracklet Clustering with Application to Person Discovery in Videos**; *SIAM Conference on Data Mining (SDM), 2015*

3. Adway Mitra, Soma Biswas, Chiranjib Bhattacharyya; **Bayesian Modeling of Temporal Coherence in Videos for Entity Discovery and Summarization**; *IEEE Transactions on Pattern Analysis and Machine Intelligence (2016)*

4. Adway Mitra, Chiranjib Bhattacharyya, Soma Biswas; **EntScene: Nonparametric Bayesian Temporal Segmentation of Videos aimed at Entity-driven Scene Detection**; *International Joint Conference on Artificial Intelligence (IJCAI), 2015*

5. Adway Mitra, Ranganath B.N., Indrajit Bhattacharya; **A Layered Dirichlet Process for Hierarchical Segmentation of Sequential Grouped Data**; *European Conference on Machine Learning (ECML-PKDD), 2013*

6. Adway Mitra; **Exploring Bayesian Models for Multi-level Clustering of Hierarchically Grouped Sequential Data**; *CoRR abs/1504.04850 (2015)*

7. Adway Mitra, Srujana Merugu; **Reconciliation of Categorical Opinions from Multiple Sources**; *Conference on Information and Knowledge Management (CIKM), 2013*

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The current age is popularly called the Information Age, due to the progress of digital technology which can collect, store and process vast amount of information to create a knowledge-rich society. Advent of the internet, and especially social media has caused an explosion of data, including user-generated content. Such data are mainly of three different types: text, image/video and audio. There is also multimedia data, which combines two or more of these types. Advancement in the technology of database, hardware and networking has enabled the efficient storage of such humongous amount of data over networked systems of servers, as well as fast retrieval in response to queries. However, though a large volume of data can be expected to contain a large amount of information/knowledge, finding these can be like looking for a proverbial needle in a haystack unless the data is well-organized and well-represented. For example, when an user searches a video-sharing website like *Youtube* with a textual query phrase, (s)he is returned a large number of videos. Many of these videos may be totally irrelevant to the query, while many others may be quite long, with the desired part hidden somewhere in it. Hence, the ready availability of data does not mean ready availability of knowledge/information. To this end, we need Data Mining and Machine Learning which can analyze the data, filter out junk, and represent the rest in a semantically concise way, so that users can easily find the information that they want from the ocean of data.

For efficient semantic analysis and concise representation of data, it is important to exploit *structures* in the data. Some of the data in the internet is *structured*, having well-known and well-defined patterns, and sometimes also having associated annotations or meta-data. But much of the data, especially user-generated data is *unstructured*, where the patterns may not be clear or well-known, and annotations are usually missing. For example, some production

companies upload their movie videos along with textual scripts, which contain detailed information about the characters, and also shot and scene changes. But much more often users shoot or record movie videos directly from the television and upload these videos on the internet, without scripts or any other meta-data. To analyze such data, we need to make use of all the implicit structural information that we can think of.

The broad aim of this thesis is to tap into the implicit structural information for user-generated data on the internet, and use them for semantic analysis of data. We focus on videos and text documents. We attempt to model semantic concepts and structural information through Bayesian methods. We demonstrate experimentally that our proposed models are useful in several applications which may allow users to efficiently browse online content and access the information they want.

1.2 Entities and Temporal Coherence

Raw data like a video or a text document is an ordered collection of pixels or word-tokens respectively. But these contain some latent semantic concepts, which are useful to human beings in making sense of the data. For example, a video may contain people, objects or actions by which the viewers understand the video. These are *semantic concepts* unlike the pixels. In this thesis, we refer to the semantic visual concepts in videos as *entities*.

Entities can be of various types- such as persons, all kinds of objects say aeroplane, car or cat, or even actions such as walking or running. One video may contain entities of different types, though an application may focus on any one type. Again, a video can contain several entities of a single type. For example, a movie involves several persons, each of whom is an entity of the type "person".

To a viewer, a video is a collection of entities of different types, rather than a dense collection of pixels. Similarly in text documents, under the collection of words there are *topics*. In this thesis, we assign importance to such semantic concepts, and explore novel applications that focus on them. However, there is no concrete definition of these, and so it may be hard to represent them mathematically. Once this is done, the challenge is to discover or learn these semantic concepts from the raw data, and represent and analyze the observed data in terms of them.

Videos, audio streams, text documents etc are *sequential* in nature. A video is a sequence of frames (or images), a text document is a sequence of word-tokens and an audio file is a sequence of phonemes. Such data is a sequence of *data-points* each of which is associated with a time-stamp. Each data-point can be represented with various features (depending on the type). Sequential data has one important property: *temporal coherence* (TC)- which means

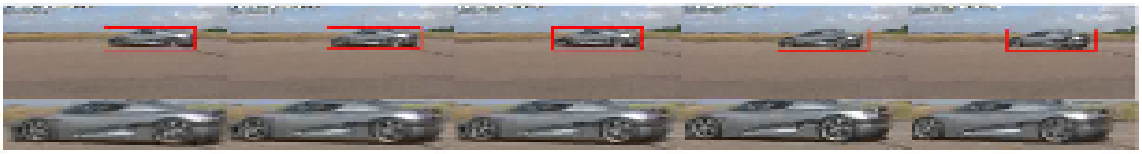


Figure 1.1: Temporal Coherence: Above, 5 successive frames of a video, all having same near-identical visual and semantic contents. Below, the entity (car) has near-identical visual features in all frames

that temporally close datapoints are *similar*, and the number of datapoints where sharp changes occur compared to neighbors are relatively few. Such similarity is usually at a semantic level, but it may also be at a feature-level. For example, in a video successive frames are often visually similar, especially in a movie or a TV-series where the camera tends to focus on a person while (s)he speaks his/her dialogue, and changes in visual content can occur only when the dialogue ends. Such similarity of visual content in successive frames is an example of feature-level temporal coherence. Semantically, the temporal coherence comes from the fact that successive frames (except changepoints) contain the same set of *entities*, like characters, objects or actions. Such semantic-level temporal coherence is also observed in other forms of data. For example, in a text document successive word-tokens are usually associated with a particular topic, except at a few changepoints, which generally coincide with change of sentences or paragraphs.

The TC property can be used to greatly simplify automated analysis of such sequential data. In a temporally coherent sequence, if the changepoints are known it is not necessary to individually analyze all datapoints independently, as most of them will be similar to their neighbors. Even if the changepoints are not known, the analysis of each datapoint can be more robust to noise, as it can be reinforced with the analyses of its temporal neighbors. For example in case of face recognition in video, even if the face in a particular frame is not well-posed and difficult to recognize independently, we will know that it is probably the same as that in the neighboring frames, due to TC.

Utilization of TC also greatly helps in concise representation of sequential data. Due to TC at feature-level, sequential data contains a degree of redundancy, which can be discarded using a more concise representation based on the appropriate features. For example, if we have 5 successive image frames that are visually similar, then the matrices of RGB values from these images will be quite similar (entry-wise), and hence they can be represented by the mean RGB matrices, from those of the individual images.

However, semantic-level TC is harder to model, as we must find an appropriate mathematical representation of the semantic concept. This concept is an entity in case of a video which may

be represented by an image, and in case of text documents the concept is a topic which can be represented as a probability distribution over the vocabulary. However, the values of these are not observed, and need to be learnt from the data. The utilization of TC can considerably improve this learning, and the question is how to make the utilization. In this thesis we study Bayesian generative models for the data, and we argue that TC can be easily modeled this way. The semantic concepts are then learnt by Bayesian inference, which may be challenging due to the complex nature of the models. In this thesis, we also propose appropriate inference algorithms to this end.

1.3 Bayesian Modeling of Sequential Data

Various machine learning techniques have been considered for various applications that involve data with some structural/sequential properties. The most prominent ones are Bayesian generative models. Such models are based on a process by which the data are assumed to have been generated. They are quite intuitive, and so this is an attractive way of modeling data with various structural properties, such as temporal coherence.

Bayesian models represent the latent semantic concepts by probability distributions, called *mixture components* $\{\phi_k\}$. For example, the semantic concept called "topic" for text documents is modeled as a discrete disctribution over the vocabulary. This is based on the intuition that, a "topic" is usually characterized by some prominent words, which can also differentiate topics. For example, a probability distribution which gives maximum weightage to words like "Obama", "Senate", "Democrat" can be interpreted as related to the semantic topic "United States Politics", while a distribution with maximum weightage on "iceberg", "warming", "carbon" etc are probably related to "Climate change".

In a generative model, the individual data-points $\{Y_i\}$ are considered to be drawn from such components $Y_i \sim \phi_{Z_i}$, where Z_i is the index of the component assigned to the i-th datapoint. The exact forms of the mixture components depend on the application and the type of data being used. The most common ones are Gaussian and Dirichlet.

The interesting questions are 1) how to assign Z_i to any datapoint i? 2) how many mixture components should be used? The first question needs to be answered in the light of the structural/sequential properties of the data. For example, if the data is grouped (like word-tokens grouped into documents) then for each group we can have a group-specific multinomial distribution, from which the Z-variables are drawn. If the data is sequential, then the distribution of Z_i should depend on the Z-values assigned to the predecessors of i. Most existing models are Markovian, i.e. Z_i depends only on Z_{i-1}. Temporal Coherence can be modeled by adding extra probability of Z_i being same as Z_{i-1}. Additional structural information need to be handled

appropriately. The Z-variables take discrete values, and hence also cause a clustering of the datapoints, within and across the groups (for example, clustering of words belonging to the same topic). We may consider several levels of clustering in case of hierarchically grouped data (for example, if we want to cluster the documents themselves).

An important issue is the number of components to be used. Most existing Bayesian methods fix it to some empirical value. These models are called *parametric*. But generally it is not possible to decide or fix the number of components for an unknown dataset found from the web. It is also not feasible to try out different values and choose the best one based on data likelihood, because of the sheer scale of the problem. We need to learn the number of topics from the data itself. To aid this, we use the field of *Bayesian Nonparametrics*, which assume potentially infinite number of components, and figure out the true number from the data.

Once the Bayesian model has been designed, the next step is to find the values of the latent variables (like $\{Z\}$ and $\{\phi\}$), using which we can write the joint distributions. While direct inference is intractable because the joint distributions are too complex, it is possible to use approximate inference like Gibbs Sampling, where we initialize the variables and sample one variable at a time, keeping the rest unchanged. It is difficult to sample continuous variables like $\{\phi\}$, so these are usually marginalized out, and estimated later (from the $\{Z\}$-values).

1.4 Applications and Challenges of Video and Text Data

Videos are very common in the web domain, due to popular video sharing sites like Youtube, Dailymotion etc, as well as social media and various sports and news sites. Videos are collections of frames, each of which is a static image. However, the successive frames usually have a lot in common, and it is unwise to treat the frames independently. Given a video, lots of questions may be asked: What is the video about? Who are the people appearing in it? What are the major objects occurring in it? What are the activities shown in it? *Computer Vision* is the area of Computer Science which aims to make sense of images and videos, and help to answer such questions.

The standard approach of Computer Vision is to represent each image or video with visual features. An image is composed of pixels which have intensity values (RGB). These values, as well as other *low-level features* (complex functions of these values) can be used to represent images and videos. These features include image gradients, SIFT points, Gabor filter outputs, Haar features etc for images, and optical flows, 3D interest points etc for videos. Appropriate features need to be chosen for every application. Some of the standard problems of Computer Vision, which can be considered solved to a reasonable extent, are Face Detection (discovering and localizing human faces in images/videos), Face Recognition (classifying human faces), de-

Figure 1.2: Tracking: A person being tracked (with red box) across video frames

tection and recognition of various objects etc. These tasks are done by making use of various low-level features as mentioned above. Machine Learning algorithms like Boosting, Principal Component Analysis, SVM, Multiple Kernel Learning etc are used on these features for learning predictive models. Again, the outputs of these tasks can be used as *high-level features* for various other tasks. For example, a face detector like [80] can be run on each frame of a video, and these detected faces can be clustered/linked together to make a list of the persons present in the video.

In this thesis, we consider three tasks regarding videos. The first one is about tracking an entity across a video, i.e. marking/localizing its position in every frame. The most common application is in surveillance. We consider particularly challenging scenarios with respect to illumination- the field of view can be dark, there may be places which are more dark than others, or there can be sudden changes in lighting (example: power-cut), and the challenge is to continue the tracking despite these distractions. The second and third tasks are related to TV-series or movie videos involving several entities, like persons. One task is to discover the entities (of a particular kind, say persons) appearing in the video, along with all frames of their occurrences. The entity discovery results can be used to summarize the video in a semantically meaningful way. The third is about temporally segmenting the video into scenes and shots. We define a scene as a subset of the video entities. In our work, we represent persons by their faces, and use Face Detectors like [80].

The most common form of data in the internet is *textual*. This includes various documents and articles in collections or archives, called *corpus* in text mining jargon. Such collections of documents usually cover a limited number of *topics*. For example, the set of all papers published in a Machine Learning conference can be considered as a corpus. They will generally talk about a limited number of topics such as classification, clustering, deep learning, probabilistic models, convex optimization etc. Again, the set of stories in a news website will cover a set of topics- each related to some contemporary incident. A lot of text mining tasks are based on topics. For example, we can classify/cluster documents into groups, each of which represents a set of topics. A long document can also be segmented into parts, where each part is about a particular

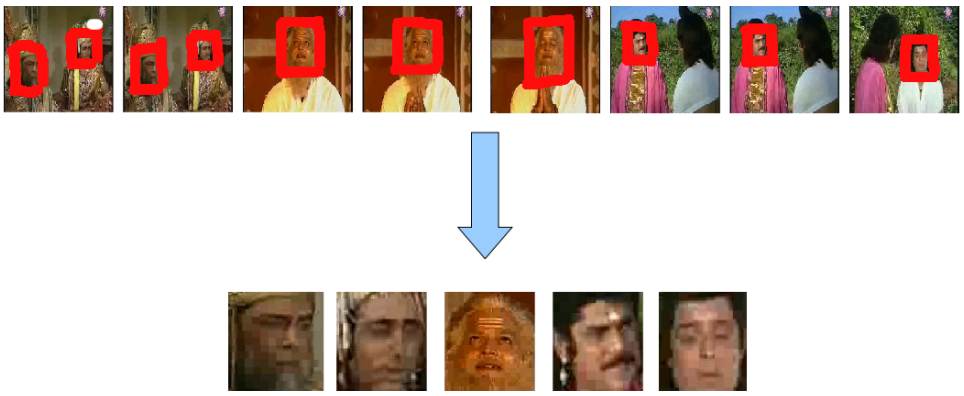


Figure 1.3: Person Discovery from videos: Video frames from a TV-series episode (top), and the corresponding persons (represented by faces) (bottom)

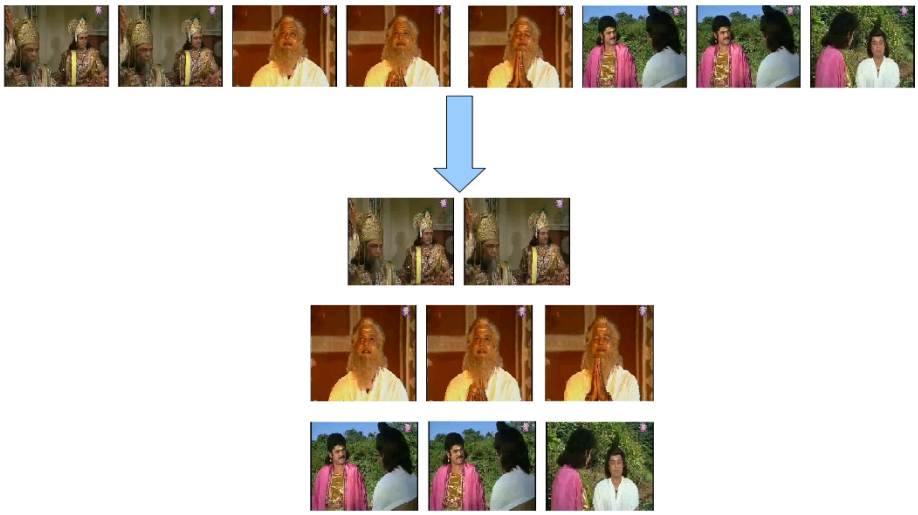


Figure 1.4: Scene Discovery from videos: Video frames from a TV-series episode (top), and the temporal segments representing scenes) (bottom)

topic. Finally, if a topic can be suitably represented, then discovering the prominent topics in a corpus can be a way of summarizing documents and/or the full corpus. Bayesian models, called *Topic Models* are very popular for this type of applications. They consider a large but finite vocabulary of words and represent each topic as a probability distribution over this vocabulary. The idea is that each topic is characterized by a few important words, which should have higher probability in the associated distribution. Various topic models are proposed to capture certain specific features of the data for specific applications- like the Focused Topic Model [85] which is designed for the situation where each document has a small set of topic which occur prominently in that document.

In this thesis we consider textual transcripts of news broadcast on TV, radio etc. They have a specific structure: broad news categories like politics, national affairs, sports etc come in a fixed order of importance, and within each broad category there are individual news stories. Our aim is to segment the transcripts into major categories and individual stories. This task also requires learning topics.

1.5 Research Gaps

Now, we discuss how current research in Computer Vision, Text Mining and Machine Learning are not adequate for our purpose.

1.5.1 Lack of focus on entities in Video Analytics

Video Analytics include many applications related to videos. Many of them are aimed at providing concise and comprehensive understanding of videos. These include, among others, video summarization, video segmentation and scene discovery, detection of face, object or actions in videos, generating textual descriptions of videos and so on. However, many of these tasks fail to recognize that people understand videos using semantic concepts like *entities*.

Video summarization aims at generating a concise and representative visualization of the video, so that people can form an idea about its content without watching it fully. But existing methods (say [18]) provide a few keyframes, chosen based on low-level visual features, as the summary. But these keyframes need not carry any significant information about the entities. Some recent methods like [69] which try to bridge this gap make use of additional information like movie scripts, which are generally unavailable for user-uploaded videos on the internet. The same criticisms also hold good about Scene Discovery of videos. Scene Discovery proceeds by temporally segmenting the video into shots, and trying to cluster these shots to form scenes [21]. But such segmentation and clustering are usually on the basis of low-level features of frames and shots. The few methods which attempt to do these semantically, based on the persons

present, again depend on scripts or other external sources of information.

Recently, there has been significant progress in object detection [27]. It is possible to detect various types of entities throughout a video by deploying an array of entity-specific detectors, and report the types of entities found [2]. Very recently, there have been attempts to generate textual descriptions of videos based on these detections.But that cannot differentiate between different entities of the same type. A face detector can locate faces, but cannot say which sets of faces belong to the same person. We need to do further analysis like clustering using these detections for that purpose.

1.5.2 Lack of models for Temporal Coherence in Videos

Temporal Coherence is a very important aspect of videos. This property can provide a lot of advantage to video-based tasks compared to image-based ones. It is manifested in many ways- at feature-level as well as at semantic level. Its potential as an implicit supervisory signal was first recognized in [65]. Almost all existing research on videos utilizes the property at feature-level, in some way or the other. For example, tracking is a well-studied problem in computer vision. Tracking methods create online appearance models for the target based on the processed frames, and try to match it in the new frame to locate the target. Thus, it exploits the property that the target's appearance will remain reasonably same in the new frame.

However, when it comes to modeling TC at semantic level, there have been few attempts to do it. Recent works on face clustering or face recognition in videos rarely incorporate the property in their generative models, and instead utilize it through pre-processing or post-processing. For example, HMRF-based face clustering [89] first detects and tracks faces, chooses a few representative detections from each track, carry out clustering based on appearance, finds the most frequent cluster assignment among the representatives of each track, and assigns the whole track to that cluster. Various generative models for different kinds of videos such as [35] assign variables to frames independent of the assignments to its neighboring frames. While there are some works where the models consider semantic-level TC in some way or other, such efforts are quite disparate.

Low-rank matrix methods have recently become popular for videos. They have found applications in background subtraction [5][14], video denoising [39] and face recognition. Each column in the matrix represents a video frame, or part of a frame (such as detection). The rationale is that, since video frames are similar, the matrix can be modeled as low-rank. Accordingly, various low-rank matrix recovery methods have been proposed. However, such methods do not consider feature-level or semantic-level TC. While they do give decent performance on the tasks they have been tested on, we find that they fail quite miserably in capturing TC.

1.5.3 Lack of models for Temporal Coherence in Documents

The situation is even worse in case of text documents. There has been a tremendous amount of research about document modeling, mostly using Bayesian models popularly called *topic models*. These have been used to model various kinds of effects in text document and corpora, such as rare topics that occur promimently in a few documents (Focused Topic Models) [85], non-prominent topics [19], a hierarchy of topics and many more. However, a very large fraction of these models considers a document to be a bag-of-words, where topic assignment to each word-token (or sentence in some cases) is done independent of the assignments to its neighbors. Very few models recognize that a document is sequentially generated, and the ordering of words are absolutely important.

One rare work related to text that does consider topic coherence is the CFACTS model [44] for modeling of customer reviews about products. [25] is another work aimed at segmenting a document based on topics. But these works have not really been followed up by the topic modeling community, and the trend of using bag-of-words representation for documents has hardly changed. Moreover, these models are parametric, and use a fixed number of topics. This is generally not possible for an unknown set of documents on the web, and we need to learn the number of topics from the data itself using Bayesian nonparametrics. A Bayesian nonparametric model called sticky HDP-HMM [30] models temporal coherence for sequential data, but it has not been used prominently in document modeling.

1.5.4 Lack of qualitative analysis of Bayesian models

Finally, a large number of Bayesian models- both parametric and nonparametric- have been developed over the last decade. Most of them have a general structure: they use a finite or infinite number of mixture components, and consider that each data-point is generated from one mixture component. There are mixture distributions to assign components to data-points. Many of the models consider data-points that are grouped at many levels [76, 91], say into documents, paragraphs and sentences. Apart from that, there are design choices for the models. The mixture distributions and mixture components may or may not be shared by all the groups. The number of components may be finite or infinite. Temporal coherence may or may not be modeled. Specific design choices about these points can make a model suitable for some applications but unsuitable for others. A large number of models have been proposed for various applications, and some of them have been compared experimentally on these applications. However, there has been no proper attempt to put these models into perspective, study the design choices made by them, and explore the choices hitherto unexplored. Nor has there been

any attempt to generalize the models.

1.6 Contributions of this Thesis

This thesis looks into several problems related to modeling of temporal coherence in sequential data having various structural characteristics. It also considers a variety of applications in two domains of data: videos and text. The thesis can broadly be divided into two parts. The first part focuses on modeling of videos for a number of applications. The second part focuses on Bayesian models for grouped sequential data (text documents), including ways to model TC, and algorithms for segmentation with application to news transcripts. An additional part focuses on Bayesian modeling of short questions and categorical answers. There is also some overlap between the parts.

1.6.1 Video modeling and analysis

The most important contribution of this thesis is in the domain of videos. The major contributions are two-fold:

1. We use entity-based representation and analysis of videos, unlike most existing works which represent and analyze videos using low-level features. Our end-goals are also defined in terms of entities, and we use appropriate measures. We lay special emphasis on entity-centric videos like movies/TV-series episodes, which primarily focus on some entities (like persons) who are few in number compared to the number of frames where they occur.

2. We emphasize the Temporal Coherence property of videos. We exploit feature-level TC to find concise representation of the data, and model semantic-level TC to improve performance on various entity-driven tasks.

The more specific contributions of this thesis regarding videos are as follows:

Tracklet-based Video Representation: A *tracklet* is a collection of detections of a particular entity (a rectangular region around the entity) in a short temporal segment of a video, usually 5-20 frames. Tracklets have been used in video literature for the past 7 years, usually in context of tracking. An innovative aspect of our work is that we use tracklets as our lowest unit of representation for entities in videos. In much of our work (Chapters 4,5), we represent the video as a sequence of tracklets, each associated with an entity. The logic is that due to feature-level TC, the detections of the same entity in successive frames are visually almost identical, so it makes sense to club such detections together as tracklets and work with them rather than the individual detections.

Robust tracklet representation for Entity Tracking: For tracklet-based video modeling, we need a representation for tracklets. One simple way, used in much of this work, is to represent each individual detection as a vector of pixel intensity values and a tracklet with the mean of such vectors. But this is not enough for some applications like continuous entity tracking (Chapter 3) under challenging scenarios, such as when there are abrupt changes in illumination, which cannot be handled by pixel intensity. For this purpose we use a more robust descriptor for individual detections- namely covariance matrix of Gabor features. We find that such matrices from successive frames have near-identical eigenvectors, which we approximate with a shared basis (Eigenprofile), and use it to build a Covariance Matrix descriptor for the tracklet.

Generative model for entities and scenes in movie/TV-series videos Most movies and TV-series episodes are *person-centric*, i.e. involves a few persons who appear repeatedly. They consist of temporal segments called *scenes*, each of which involves a subset of these persons. In Chapter 5 we consider a video as a sequence of tracklets, each of which is associated with a entity (person) and a scene. We propose a generative approach which models entities as mixture components, tracklets as draws from these components, and scenes as sparse distributions over components.

Bayesian nonparametric modeling of TC In videos, semantic-level TC dictates that temporally close tracklets are more likely to have the same associated entity and scene. We handle this in our generative model, using a Markovian approach, in Chapters 4,5.

Entity Discovery in Videos In Chapter 4, we consider the task of discovering the entities (say persons) who appear in the videos, along with all frames where they occur. Unlike existing methods for this task, we do not make use of meta-data like scripts, or other training videos where the entities have been marked. This is achieved by clustering the tracklets, where each cluster is associated with an entity. Since the number of entities is not known, the Bayesian nonparametric approach helps to find out the suitable number of clusters to be formed. We approach the task in three settings- once where we disregard temporal structures like scenes and shots, once when we consider shots with known boundaries, and once when we simultaneously discover the scene boundaries. We also consider this task on streaming videos. Further, we show that our approach can also filter out tracklets created by false entity detections.

Entity-driven Video Summarization Video summarization methods return a few frames (keyframes) or short segments that are considered to be *representative* of the whole video. However, the representativeness is usually considered in terms of low-level features. In this work (Chapter 4), we take an entity-driven approach to summarization, based on entity discovery (mentioned above). Also, all video summaries are supposed to be concise and representative,

but well-defined measures for these are absent. We define these quantities in terms of the entities and their associated tracklets.

Entity-driven Temporal Segmentation (Scene Discovery) Videos have a hierarchical temporal structure, as they are organized into long, heterogeneous scenes and short homogeneous shots. Shot changes can be detected easily by change in visual features of frames, but scene changes are more difficult to define and detect. In Chapter 5 we define scenes in terms of entities, and model each scene as a sparse distribution over entities. We propose algorithms for temporal segmentation of the data where each segment should correspond to a scene.

Low-rank Matrix representation for videos Matrix-based video representations are quite common, where each column corresponds to a frame (or part of it). Due to TC, it is known that adjacent columns are similar, and hence the matrix is modeled as low-rank. However, all existing approaches to low-rank matrix-based modeling try to minimize the rank through the singular values, which does not at all capture the TC property in the columns, as we find in Chapter 6. We try to enforce this within the existing convex optimization framework, by adding suitable regularizers, and see a slight improvement in performance. But we see that Bayesian non-parametric modeling of such low-rank matrices using a distribution over the columns gives a better performance, and also considerable computational efficiency, since it avoids the expensive step of singular value decomposition used in the convex optimization methods.

1.6.2 Bayesian models for grouped sequential data

In the first part of the thesis , we represented a video as a sequence of tracklets. The sequence is temporally coherent, and contains additional structure: there are subsequences (scenes) each of which has an associated distribution over the mixture components. The aim of segmentation is to discover these subsequences. It can also be looked upon as *linear clustering* of the data. In the second part of the thesis (Chapter 7), we look at a generalized version of the problem: Bayesian modeling for *Hierarchically Grouped* sequential data, where a *group* is a (usually predefined) contiguous block of data. Our aim is multi-level clustering of the data-points as well as the groups themselves. The main contributions are as follows:

A comparative study of Bayesian Models for Grouped Data Lots of Bayesian models (parametric/non-parametric) have been proposed for grouped data, primarily for text applications. These include the *topic models*. However, these models differ from each other on issues like how many levels of grouping they consider, to what extent they share the mixture components (i.e. to what extent they allow clustering of the groups at different levels) and so on. In this work we introduce a taxonomy called Degree-of-Sharing (DoS) for this kind of data, using

which we theoretically compare various existing Bayesian models.

Markovian modeling of TC in hierarchically grouped data Most of the existing models (which we compare above) are suitable for *completely/partially exchangeable data*. But we focus on sequential data, which exhibit temporal coherence, perhaps at multiple levels of grouping. To model TC, we attempt to tweak the existing models through a Markovian approach, which encourages each datapoint/data-group to be assigned the same mixture components/distributions as their predecessors. This process is generalized to any number of "layers". We call this model as the *Layered Dirichlet Process*, which can also be specialized to various existing models by adjusting settings.

Semi-Markovian modeling of TC in hierarchically grouped data Markov models are limited in the sense that they work only *locally* but cannot influence the global structure of the data. But in certain types of sequential data, a global structure is known, which must be captured through the model. *Semi-Markovian/Explicit-duration* models help in some cases like this, which we explore.

Inference Algorithms for Sequence Segmentation The above discussion on grouped sequential data deals with the case where the groupings are known. But in case they are not, we have a task of linear clustering/segmentation, which has already been studied in the previous part in context of temporal segmentation of videos. Here, we consider a more generalized setting, and study inference algorithms for the Markovian and semi-Markovian models of sequential data that we discussed above.

Hierarchical Segmentation of news transcripts News transcripts have a hierarchical structure: there are broad categories like politics, sports etc which appear in a fixed order, and individual news stories within them. We want to segment them simultaneously at both levels- i.e. into categories and individual stories. This task perfectly fits into the framework for hierarchically grouped sequential data with unknown grouping. We compare both the Markovian and Semi-Markovian approaches discussed above, and find the latter to be more appropriate.

1.7 Organization of this Thesis

Finally, we describe the structure of this thesis. We define some of the terms we have used extensively, and also the contents of the chapters.

1.7.1 Definitions

Some jargon which we have used in this work:

- **Entity** A semantic concept having a visual representation. Persons, various kinds of

objects and actions are all entities. In a particular application, we may focus on only one type of entities, such as persons. In a movie video, each person (character) is considered an entity. Again, in a F1-racing video, each car is an entity.

- **Spatial Fragment** A (usually) rectangular region in a video frame. Each spatial fragment has a spatio-temporal location in a video, where the spatial location is its coordinates within the frame, and its temporal location is the frame index.
- **Temporal Fragment** A collection of adjacent frames in a video
- **Spatio-temporal Fragment** A collection of spatio-temporally close spatial fragments. They are usually from adjacent frames.
- **Detection** A spatial fragment in a video frame that encloses an entity. It can be found by running an entity-specific detector such as [80, 27].
- **Tracklet** A spatio-temporal fragment associated with an entity. Specifically, in a tracklet each spatial fragment encloses the entity.
- **Shot** A temporal fragment of a video where each frame contains the same set of entities
- **Scene** A temporal fragment of a video that is associated with a set of entities, which is different from the set of entities in the neighboring frames or shots outside the TF. It is usually a sequence of contiguous shots.
- **Appearance Model** A mathematical description for an entity
- **Tracking** Locating an entity in the frames of a video. In each new frame, it is located by making use of its location and appearance model in the previous frames
- **Entity Discovery** Finding appearance models for all entities in a video, and also find all locations of each of them
- **Video Summarization** Finding a concise and representative visual description of the video. It is usually a collection of frames (called keyframes) or temporal fragments
- **Entity-driven Video Summarization** Finding a concise and representative visual description of the video which provides information about the entities
- **Temporal Segmentation** Partitioning the video into non-overlapping temporal fragments

- **Scene Discovery** Temporal segmentation of a video where each segment coincides with a scene
- **Grouped data** Data where each data-point Y_i is associated with observed group variables $(G_i^1, \ldots, G_i^L)$. These variables are associated with levels or layers $1, 2, \ldots, L$.
- **Hierarchically grouped data** Grouped data where $G_i^l = G_j^l \implies G_i^{l'} = G_j^{l'}$ where $l' \leq l$, for every pair of datapoints (i, j). In other words, groups in a higher layer entirely cover groups at a lower layer.
- **Hierarchical clustering** Associating cluster indices $(Z_i^1, \ldots, Z_i^L)$ to data-points, such that $Z_i^l = Z_j^l \implies Z_i^{l'} = Z_j^{l'}$ where $l' \leq l$, for every pair of datapoints (i, j). In other words, clusters in a higher layer entirely cover clusters at a lower layer.
- **Segmentation** Associating variables S_i to each data-points such that, for each datapoint i, $S_i = S_{i-1}$ or $S_i = S_{i-1}+1$. Informally it is partitioning sequential data into non-overlapping sets of contiguous data-points
- **Hierarchical Segmentation** Associating segmentation variables S_i^l to each datapoint at each layer, such that $S_i^l = S_j^l \implies S_i^{l'} = S_j^{l'}$ where $l' \leq l$, for every pair of datapoints (i, j). In other words, segments in a higher layer entirely cover segments at a lower layer.
- **News story** An event of contemporary interest, which can be modeled by a probability distribution over the vocabulary
- **News category** A high-level concept such as politics or sports, with which several stories may be assoiated
- **Temporal Coherence at Semantic level**: The property that at any layer l, $Z_i^l = Z_{prev(i)}^l$ with high probability, where $prev(i)$ is the predecessor of datapoint i defined in some way. In various applications considered here, this may mean any of the following depending on the concept:
    1. Successive frames contain same set of entities
    2. Successive tracklets associated with same set of entities
    3. Successive frames associated with same shot
    4. Successive shots associated with same scene
    5. Successive word-tokens (or sentences) associated with same story

6. Successive word-tokens (or sentences) associated with same news category

- **Temporal Coherence at Feature level**: The property that $Y_i \approx Y_{prev(i)}$ with high probability

1.7.2 Organization of the Chapters

Chapter 2 we briefly describe the relevant literature. We discuss prior work related to image and video processing, various video analytics applications, Bayesian nonparametrics and low-rank matrix recovery. In Chapter 3 we describe our tracklet representation, which is used for tracking entities from videos. Chapter 4 discusses temporal coherence modeling through Bayesian nonparametric approach, and its application to entity discovery and entity-driven summarization. Chapter 5 presents a Bayesian nonparametric model and inference algorithms for entity-driven scene discovery. Modeling temporal coherence in low-rank matrix based video representations is briefly explored in Chapter 6. After that, in Chapter 7 we make a detailed study of models for grouped sequential data, along with an application to news transcript segmentation. Finally, in Chapter 8 we describe a Bayesian model for user expertise to elicit correct answers from opinions provided by users.

Chapter 2

Related Works

In this chapter, we will review some of the background material that is relevant to this thesis. There are broadly two parts: Computer Vision and Machine Learning. In the first part we will first look into traditional problems in videos, such as Face Detection and Representation, Tracking, Tracklets etc. Next, we will look into modern applications of videos, that require semantic analysis of the videos, using the tools discussed earlier. The second part of the chapter looks at the basics of Bayesian Nonparametrics and a few introductory models, followed by a discussion of Bayesian approach to sequence segmentation. Finally we discuss Low-rank Matrix Recovery and Constrained Clustering.

2.1 Video Processing and Representation

A video is a collection of frames, where each frame consists of persons/objects/actions. To understand videos we need a way of automatically detect and identify them. The big problem is that, most of these do not have any well-defined template, by matching which they can be detected or identified in the images. This is because of their complex appearances, and also because of their variability. For example, if we are interested in detecting cars in images, we are faced with the problem that cars can be of various sizes, shapes and colours. Moreover, the same car appears different from different directions and angles, and it can also be partly occluded. Such uncertainties make the task of detection and identification of people/objects so difficult. Computer Vision makes use of sophisticated image features and machine learning algorithms to learn robust models from many training examples.

2.1.1 Face/Object Detection and Representation

Face detection is the task of locating human faces in images. This is one of the most well-researched problems in Computer Vision, and has been solved to a large extent. Today, even

mobile phones and cameras are equipped with Face Detection software. One of the most popular face detection algorithms is the Viola-Jones Face Detector [80], which represents image regions using Haar-like Features, and uses a Machine Learning technique called Boosting. Given any new image, Viola-Jones face detector examines all rectangular windows of different sizes in the image, and classifies them as face or non-face. The final output of face detection is a set of rectangular regions in the image, each of which is believed by the algorithm to enclose faces. Recently detectors have been built for various other objects such as cars, aeroplanes, trees etc. These are based on Part-based object models proposed by Felzenwalb et al [27], learnt using Latent Variable Support Vector Machine.

The same technique for face/object detection can be applied to videos also, on a per-frame basis. However, domain adaptation is an issue for every face/object detector, because the set of images on which it has been trained is usually significantly different from the ones on which it is expected to perform the detections after deployment. It has also been argued that detectors trained on static images often give suboptimal performance on videos. Domain adaptation methods update trained models with a few training samples, both positive and negative, from the new domain. Domain Adaptation for detectors to videos has been studied recently in [71] [74].

The detections from images/videos are often used as input for other tasks. For example, detected faces may be used for face classification or clustering for person identification. For further processing it is necessary to *represent* the detections. One of the simplest, and yet quite effective representation technique is to convert the rectangular sub-images to greyscale, scale them down to fixed dimensions and vectorize them. Thus, each detection is represented with a vector of greyscale pixel values. In case colour information is important and should be preserved, the images should not be converted to greyscale, but the vectorization should be done separately for the three colour channels, and then these three vectors should be joined to form a single long vector. This vectorization technique has been used successfully in several recent papers related to face recognition, like [59] [87]. In case of videos, these vectors are juxtaposed to form a matrix. The main virtue of this representation is its simplicity, while its main drawback is that it is susceptible to illumination changes. A more robust representation is the Covariance Matrix [79] of suitably illumination-invariant features, like Gabor filters [58]. Both pixel intensity vectors and Covariance Matrices are general enough to be used for faces as well as different objects. Some specialized face representations like STASM [52] are also available, which locate facial feature points, and compute feature vectors like SIFT [47] around them.

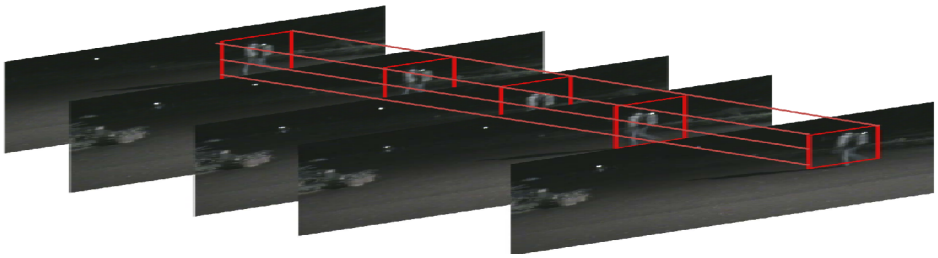


Figure 2.1: Corresponding Spatial fragments from 5 successive frames, each showing a person walking, linked together to form an STF

2.1.2 Tracking and Tracklets in Videos

Tracking is the task of locating one or more objects in the successive frames of a video. Research in tracking has aimed to build efficient appearance models for objects that are robust to real-world challenges, like illumination changes, variations in pose, scale, shape etc. Efficient and concise space-time representation of objects being tracked is thus a challenging task. In tracking literature, three main approaches have been proposed for target modeling. They are low-level (pixel-based), mid-level (region-based) and high-level (parts, shape or pose based). Low-level approaches include tracking interest points such as SIFT [46] and high-level approaches involve building more sophisticated models, like the individual body parts of a human, and tracking then simultaneously [3]. But in videos shot under insufficient illumination, the individual parts are often not visible and the frames are noisy and grainy, rendering interest Points very unreliable. Rather, the target appears like a blob or patch, which suggests a region-based (mid-level) approach. Region-based tracking requires efficient region descriptors,like Colour Histograms [17]. But a more powerful and efficient method is the Covariance Descriptor [79], which is used for Region tracking [61]. Covariance Descriptor of a region in an image is the sample covariance matrix of the feature vectors at locations within that particular region. This descriptor has been shown to be robust to noise and scale.

We use the term **spatial fragment**(SF) to indicate a region within an image, and **temporal fragment**(TF) as a collection of frames within a time-window. The cube formed by stacking the corresponding **Spatial-Fragments (SF)**, which are the spatial regions containing object within a frame, within a **Time-Fragment (TF)** as the **Spatio-Temporal-Fragment (STF)**.

Covariance Tracker [61] models the target by a statistic of target SFs collected over a TF. This statistic is the *Geodesic Mean* of the SF Covariance Matrices from these frames. Wu et al propose [90] for learning another statistic for STFs, which is equivalent to pooling the features

from target SFs in different frames together and estimating the Covariance Matrix.

During tracking, it is important to build an *appearance model* for the entity being tracked, and continuously update it (eg. [68]). Building such appearance models not only helps in the tracking process itself, but also in grouping the created tracks themselves, for further processing. As an example, [73] tries to create appearance models for each object from multiple views (which may never simultaneously appear in any frame), to improve the tracking and also for object-level grouping of tracks.

In tracking literature, a recent paradigm is the fragment-based approach [1], where multiple image patches (SFs) are used to build a template for an object within a single frame. Increasing the number of SFs improve the tracking performance, but also require larger model size. Another recent approach [34] uses multiple images patches and their relative positions for tracking using a discriminative learning framework - namely structured support vector machines.

Object Detection has recently seen significant developments (like [27]), and it is often possible to have object-specific detectors. If such detectors are available for the target being tracked, it is possible to locate the target in some frames using such a detector, and the link the detections based on spatio-temporal locality. This approach is called *Tracking by Detection*([3]). Tracking targets over long videos 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** [36] and then linking these tracklets from successive windows to form tracks. Unlike normal tracking which is done online, tracking-by-detection is usually done offline, because detectors cannot run at real-time video speed. A compromise is to run the detector on only a subset of the frames.

A tracklet is an STF, that has an entity associated with it. Any set of spatio-temporally close SFs from a short segment of frames, each of which encloses an entity, can be clubbed together to form a tracklet.

2.1.3 Temporal Coherence in Videos

TC is a fundamental property of videos. As already discussed, TC can occur at both feature level and semantic level. Most papers related to videos exploit feature-level TC in some way or other. For example, all approaches to *tracking* make assumptions such as the target entities are present in successive frames, and have similar position and appearance. Again, some papers like [65] make use of TC as a supervisory signal. In tasks aimed at *treatment of raw videos*, such as video denoising [39], temporal up-sampling [50], resizing [83], retargetting [95] etc, the aim is to maintain temporal coherence- the similarity of successive frames- with the aim of making the video visually aesthetic.

However, relatively few papers attempt to *model semantic-level TC*. [55] defines Temporal Coherence at semantic level: *successive frames should contain the same objects*, and enforces it using deep belief networks. Some other papers have independently tried to utilize semantic-level TC in their models. For example, in case of label propagation [6], label assignment is done to pixels or superpixels in a frame based on the label assignments to corresponding pixels/superpixels in the previous frame. Again, in the video object model [16], the assignment of object category label to each superpixel is influenced by the assignments to its spatial as well as temporal neighbors. However, these are all disparate, and there has been no conscious attempt at modeling this important phenomena.

2.2 Video Analytics

Person Discovery in Videos is a task which has recently received attention in Computer Vision. Cast Listing [4] 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 [98] or labeled training videos having the same characters [75]. Scene segmentation and person discovery are done simultaneously using a generative model in [45], but once again with the help of scripts. An unsupervised version of this task is considered in [89], which performs **face clustering** in presence of spatio-temporal constraints as already discussed. For this purpose they use a Markov Random Field, and encode the constraints as clique potentials. Another recent approach to face clustering is [92] which incorporates some spatio-temporal constraints into subspace clustering.

Tracklet Association *Tracking* is a core topic in computer vision, in which a target object is located in each frame based on appearance similarity and spatio- temporal locality. A more advanced task is *multi-target tracking* [94], in which several targets are present per frame. A tracking paradigm that is particularly helpful in multi-target tracking is *tracking by detection* [3], where object-specific detectors like [27] 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* [36] which attempts to do the linking hierarchically. This requires pairwise similarity measures between tracklets. Multi-target tracking via tracklets is usually cast as Bipartite Matching, which is solved using Hungarian Algorithm. Tracklet association and face clustering are done simultaneously in [88] using HMRF.

Finally, **Video Summarization** has been studied for a few years in the Computer Vision community. The aim is to provide a short but comprehensive summary of videos. This summary is usually in the form of a few *keyframes*, and sometimes as a short segment of the video around these keyframes. A recent example is [18] which models a video as a matrix, each frame

as a column, and each keyframe as a *basis vector*, in terms of which the other columns are expressed. A more recent work [62] considers a kernel matrix to encode similarities between pairs of frames, uses it for *Temporal Segmentation* of the video, assigns an importance label to each of these segments using an SVM (trained from segmented and labeled videos), and creates the summary with the important segments. However, such summaries are in terms of low-level visual features, rather than high-level semantic features which humans use. An attempt to bridge this gap was made in [69], which defined movie scenes and summaries in terms of characters. This work used face detections along with *movie scripts* for semantic segmentation into shots and scenes, which were used for summarization. Another attempt at semantic summary in terms of action was attempted in [63], however they have been tried only for short sequences, and the non-chronological nature of the summaries make them confusing.

Video Scene Discovery is another task related to temporal segmentation of videos. A video of a movie or TV-series generally consists of many *shots*- short temporal segments in which the camera is fixed and all frames show the same set of entities (say persons). A *scene* is a bigger temporal segment, covering a contiguous set of shots which are semantically related. Shot discovery and scene discovery are both *temporal segmentation* tasks, but the former is easy and the latter difficult. Shot discovery can be done by linearly clustering frames based on temporal features. Scene discovery is usually done by further clustering the shots. Various methods have been described in [21]. Most of them cluster and link shots based on low-level feature similarity. A few methods like [69] do attempt scene discovery in terms of characters, but these again require movie scripts or labeled training datasets which allow them to do face recognition of the characters.

2.3 Bayesian Nonparametrics

DP Mixture Model and Complete Exchangeability: A random measure G on Θ is said to be distributed according to a Dirichlet Process (DP) [28] ($G \sim DP(\alpha, H)$) with base distribution H and concentration parameter α if, for every finite partition $\{\Theta_1, \Theta_2, \ldots, \Theta_k\}$ of Θ, $(G(\Theta_1), G(\Theta_2), \ldots, G(\Theta_k)) \sim Dir(\alpha H(\Theta_1), \alpha H(\Theta_2), \ldots, \alpha H(\Theta_k))$. The stick-breaking representation shows the discreteness of draws G from a DP:

$$\begin{aligned} \phi_k \sim H; \beta_k = \hat{\beta}_k \prod_{i=1}^{k-1} (1 - \hat{\beta}_i) \\ \hat{\beta}_i \sim Beta(1, \alpha); G = \sum_k \beta_k \delta_{\phi_k} \end{aligned} \tag{2.1}$$

We write $\beta_k \sim GEM(\alpha)$.

Given N independent draws $\{\theta_i\}_{i=1}^N$ from G as above, the predictive distribution of the next draw, on integrating out G, is given by $p(\theta_{N+1}|\theta_1 \dots \theta_N) \propto \sum_{k=1}^K N_k \delta_{\phi_k} + \alpha H$, where $\{\phi_k\}_{k=1}^K$ be the K unique values taken by $\{\theta_i\}_{i=1}^N$ with corresponding counts $\{N_k\}_{k=1}^K$. This shows the clustering nature of the DP. Using the DP as a prior results in an 'infinite mixture model' for data $\{Y_i\}_{i=1}^N$ with the following generative process:

$$G \sim DP(\alpha, H); \quad \theta_i \overset{iid}{\sim} G; \; Y_i \overset{iid}{\sim} F(\theta_i), \quad i = 1 \dots N \tag{2.2}$$

where F is a measure defined over Θ. This is called the DP mixture model [28]. This can alternatively be represented using the stick-breaking construction and integer latent variables Z_i as follows:

$$\beta \sim GEM(\alpha); \; \phi_k \sim H, \; k = 1 \dots \infty \; ; \quad Z_i \sim \beta; \; Y_i \sim F(\phi_{Z_i}), \; i = 1 \dots N \tag{2.3}$$

An important notion for hierarchical Bayesian modeling is that of exchangeability [9, 22]. Given any assignment $z = \{\bar{z}_1, \bar{z}_2, \dots, \bar{z}_N\} \in \mathcal{S}$ to a sequence of random variables $\{Z_N\}$, where $\mathcal{S}$ is a space of sequences, exchangeability (under joint distribution P on $\mathcal{S}$) defines which permutations $z\pi = \{\bar{z}_{\pi(1)}, \bar{z}_{\pi(2)}, \dots, \bar{z}_{\pi(N)}\}$ of the assignment have the same probability (under P). In general, any notion of exchangeability E is defined using a statistic, which we call Exchangeability Statistic $S_E(Z)$. A model, defining a joint distribution P, is said to satisfy exchangeability E if $S_E(z) = S_E(z\pi)$ implies $P(z) = P(z\pi)$, for all permutations π on $\{1, \dots, N\}$.

Given a sequence $Z \in \mathcal{S}$, define $S_C(Z) = \{N_i\}_{i=1}^K$ as the vector of counts of the K unique values occurring in it, where N_i is the count of the i^{th} unique value. Using $S_C(Z)$ as the exchangeability statistic leads to the definition of Complete Exchangeability (CE), under which all permutations are equiprobable.

De Finetti's Theorem [20] shows that if an infinite sequence of random variables z is infinitely exchangeable (meaning that every finite subset is completely exchangeable) under a joint distribution $P(Z)$, then the joint distribution can be equivalently represented as a Bayesian hierarchy:

$$P(Z) = \int_\theta P(\theta) \prod_i P(Z_i|\theta) d\theta \tag{2.4}$$

It can be shown that a sequence drawn from a DP mixture model, using a similar hierarchical generation process, satisfies Complete Exchangeability.

HDP Mixture Model and Group Exchangeability: Now consider grouped data of the form $\{Y_i, D_i\}_{i=1}^N$, where $D_i \in \{1, D\}$ indicates the group to which Y_i belongs. The Hierarchical Dirichlet Process (HDP) [76] allows sharing of mixture components $\{\phi_k\}$ across groups using two levels of DPs:

$$\begin{aligned} \phi_k \sim H,\ k = 1 \ldots \infty; \quad \beta \sim GEM(\gamma),\ \theta_j \sim DP(\alpha, \beta),\ j = 1 \ldots D \\ Z_i \sim \theta_{D_i};\ Y_i \sim F(\phi_{Z_i}),\ i = 1 \ldots N \end{aligned} \tag{2.5}$$

This generative procedure for the data is called the HDP mixture model. We have modified the representation to make the group variable explicit, which we can build upon for our work. Note that the HDP can also be represented directly using measures instead of indices. The HDP mixture model can be shown to satisfy a notion of partial exchangeability called Group Exchangeability. For grouped data of the form $\{Z_i, D_i\}_{i=1}^N$, where the Z_i and D_i variables take K and D unique values respectively, define $S_G(z, g) = \{\{N_{j,k}\}_{k=1}^K\}_{j=1}^D$, where $N_{j,k} = \sum_{i=1}^N \delta(Z_i, k)\delta(D_i, j)$. Group Exchangeability (GE) is characterized by the exchangeability statistic $S_G(Z, D)$. For GE models, all intra-group permutations are equiprobable, but probability changes with exchange of values across groups.

Other Group Exchangeable Nonparametric Models: For grouped data $\{Y_i, D_i\}_{i=1}^N$, the Nested Dirichlet Process (NDP) [67] proposes the following generative model with two layers of latent variables (Z^2, Z^1) for each data item:

$$\begin{aligned} \phi_{k,l} \sim H,\ k, l = 1 \ldots \infty; \theta_k^1 \sim GEM(\beta),\ k = 1 \ldots \infty; \theta^2 \sim GEM(\alpha); \\ Z_g^2 \sim \theta^2,\ g = 1 \ldots G; Z_i^1 \sim \theta_{Z_{D_i}^2}^1; Y_i \sim \phi_{Z_{D_i}^2, Z_i^1},\ i = 1 \ldots N \end{aligned} \tag{2.6}$$

This is *hierarchical clustering*, i.e. the clusters of datapoints defined by Z^1 are themselves clustered by Z^2. The indices g represent such second-level clusters.

Unlike the HDP, only some groups share mixture components. Additionally, unlike the HDP they also share distributions over these components.

The MLC-HDP [91] models *hierarchically grouped data* of the form $\{Y_i, D_i^1, D_i^2, D_i^3\}_{i=1}^N$, which is grouped at 3 different levels, and proposes the following generative process:

$$\begin{aligned} \phi_k \sim H,\ k = 1 \ldots \infty; \beta^3 \sim GEM(\gamma^3),\ \beta^2 \sim GEM(\gamma^2),\ \beta^1 \sim GEM(\gamma^1); \\ \theta^3 \sim DP(\alpha^3, \beta^3),\ \theta_k^2 \sim DP(\alpha^2, \beta^2),\ \theta_l^1 \sim DP(\alpha^1, \beta^1),\ k, l = 1 \ldots \infty; \\ Z_a^3 \sim \theta^3\ \forall a;\ Z_{ab}^2 \sim \theta_{Z_a^3}^2 \forall a \forall b;\ Z_{abc}^1 \sim \theta_{Z_{ab}^2}^1 \forall a \forall b \forall c;\ Y_i \sim \phi_{Z_{D_i^3, D_i^2, D_i^1}^1},\ i = 1 \ldots N \end{aligned}$$

Unlike NDP where the clusters of one level are themselves clustered, here the pre-defined groups are clustered. Since these groups are hierarchical, the clustering is also hierarchical, but restricted by the grouping. The indices a, b, c denote clusters at different levels of the grouping hierarchy. Unlike NDP, here the mixture components can be shared by all groups, and two groups can have identical distributions over these components with non-zero probability.

2.4 Bayesian Sequence Segmentation

Now we come to the segmentation problem for a sequence $\{Y_i, Z_i\}$ where the the variables Y_i are observed while $Z_i \in \{1, 2 \ldots\}$ are latent, with distribution $P(Y, Z) = P(Z)P(Y|Z)$. Given any assignment to the $\{Z_i\}$ variables, segments are defined as maximal sub-sequences (s, e) such that $Z_e = Z_s = Z_i$ for $s \leq i \leq e$. Since $\{Z_i\}$ variables are random, a natural definition for the segmentation problem is to first perform inference to find the optimal assignment to $\{Z_i\}$ according to the posterior distribution $P(Z|Y)$, and then identifying segments for this assignment. Instances of this problem include segmentation according to topics for textual documents, and according to speaker in conversational audio. Naturally, distinguishing between different permutations is critical for segmentation of grouped (un-grouped) data, and GE (CE) assumptions for $P(Z)$ are not appropriate, since all permutations are equiprobable. Therefore, HDP (DP) mixture models are not suitable for such segmentation tasks. These call for more discerning models that satisfy other notions of exchangeability that distinguish between different segmentations of $\{Y_i, Z_i\}$ represented by different assignments to $\{Z_i\}$.

Markov Models To model (ungrouped) data $\{Y_i\}$ with such properties, the Hierarchical Dirichlet Process- Hidden Markov Model(HDP-HMM) [30] considers the mixture components Z_i as states of an HMM with infinite state-space. This is done by identifying the groups as well as the mixture components in the HDP with the HMM states. Now $\theta_j \sim DP(\alpha, \beta)$ is considered as transition distribution for the j^{th} state, and is used to generate the next state:

$$\theta_j \sim DP(\alpha, \beta),\ j = 1 \ldots \infty; \quad Z_i \sim \theta_{Z_{i-1}};\ Y_i \sim \phi_{Z_i},\ i = 1 \ldots N \tag{2.7}$$

A special case of this is the Sticky HDP-HMM (sHDP-HMM) [30], which increases the probability of self-transition as $\theta_j \sim DP(\alpha + \kappa, \frac{\alpha\beta + \kappa\delta_j}{\alpha + \kappa})$, to enforce sequential continuity of mixture components which occur naturally in speech (where a mixture component represents a speaker) and text (where a mixture component represents a topic). This is the Bayesian approach to modeling *Temporal Coherence*. Though originally developed for single sequences, the HDP-HMM and sHDP-HMM models can also be extended for grouped data.

Consider the following statistic: $S_M(Z) = (\{N_{ij}\}_{i=1,j=1}^{K,K}, s)$, where N_{ij} is the number of

transitions from the i^{th} unique value to the j^{th} unique value in the sequence Z, and $Z_1 = s$. Using S_M as the exchangeability statistic leads to the definition of Markov Exchangeability (ME) [22]. Intuitively, this means that two different sequences are equiprobable under the joint distribution, if they begin with the same value and preserve the transition counts between unique values. Representation theorems, similar to De Finetti's theorem, exist for Markov Exchangeability as well [22]. It can be shown that the HDP-HMM and sticky HDP-HMM mixture models satisfy Markov Exchangeability.

Semi-Markov Models Markovian approaches like sHDP-HMM model temporal coherence by providing additional weightage κ to the latent variable Z_{i-1} for each datapoint W_i. Depending on κ, long runs of Z-value are expected before a change, i.e. $Z_i \neq Z_{i-1}$. This way, the Markovian approach can help to model segments, and an expected length of each segment can also be computed. But it may be argued that such a modeling is local and not global. These models do not define segments explicitly and the segmentation is achieved as a by-product of the assignment of mixture components. Also in cases that where the number of segments is fixed and known, this approach cannot ensure that the required number of segments will be formed. To answer such criticisms there are Hidden Semi-Markov Models (HSMM) [97], also called *explicit duration models*. It defines a distribution over the duration of every possible value of Z. At point i a tuple (z, d) is sampled, which means that $Z_i = Z_{i+1} = \cdots = Z_{i+d-1} = z$. If this sampling is done at point i, (z, d) may be drawn conditioned on Z_{i-1}. This can also be extended to the case where the number of states, i.e. set of possible values of Z is unknown, where another Bayesian nonparametric model can be employed along the lines of HDP- namely HDP-HSMM [40]. Explicit-duration models for some more complex structures in the data are discussed in [37].

Hierarchical Models Neither sHDP-HMM nor HDP-HSMM are hierarchical model, i.e. they associate only one latent variable Z_i to each datapoint Y_i. Topic Segmentation Model [24] is a two-level hierarchical model. It represents each segment with a mixture distribution, and each point has an associated binary variable to decide whether or not a new segment begins from it. Each segment uses a separate distribution over the mixture components. However, it uses a fixed number of mixture components, and does not take into account the assignment to predecessors while assigning mixture components to the datapoints.

The main challenge for the Bayesian approach lies in the **inference**. Exact inference algorithms have been considered for Product Partition Models [26][93] by considering a prior on lengths of segments. However, the complex hierarchical models require approximate inference. sHDP-HMM [29] performs inference by Gibbs Sampling, where the latent variable assignments

to each datapoint are sampled conditioned on the assignments to its neighboring datapoints. For Topic Segmentation Model [24] a split-merge inference algorithm is considered, where each initial changepoint is provided a binary variable which indicates whether or not it is a true changepoint. The inference by Gibbs Sampling involves sampling this variable along with the mixture component assignments.

A Bayesian model for **co-segmentation** of sequences is the Beta Process Hidden Markov Model (BP-HMM) [86] that considers mixture components to be shared among several sequences. It has been used for modeling human actions in videos. However, it is neither hierarchical nor does it model TC. Spatial Co-segmentation of videos using Bayesian nonparametric framework has been studied recently [16], using Distance-dependent Chinese Restaurant Process [8] to model spatial coherence. LaDP [53] has also been used for co-segmentation of news transcripts.

2.5 Low-rank Matrix Recovery

A problem of great current interest in Machine Learning is low-rank matrix recovery. It finds applications in mainly in Computer Vision, Collaborative Filtering and Recommender Systems. The input to the problem is a matrix M, and the task is to find a low-rank matrix X that is *close* to the given matrix. Generally two settings are considered:

1. **Low-rank Matrix Completion**: Here, the given matrix is incomplete, with entries missing uniformly at random (in some works the uniformity assumption is relaxed). The original task is to find a low-rank matrix that agrees with the observed entries Ω of given matrix, i.e. $X_\Omega = M_\Omega$.

$$\min_X rank(X) \textbf{ such that } X_\Omega = M_\Omega \tag{2.8}$$

   The most common application of this problem is in collaborative filtering and recommender systems, like movie rating predictions [51]. The matrix is considered to be a movie rating matrix, where X_{ij} is the rating provided by user i to movie j. Only a small number of movies have been rated by users, and the target is to predict the remaining. The matrix is considered to be low-rank because it is expected that the ratings given by users are partly correlated. The problem also finds application in Computer Vision, like denoising of images and videos [39]. The matrix entries are pixel intensity values, and the noisy pixels are considered to be missing. Such matrices are considered to be low-rank because of spatio-temporal coherence of images and videos.

2. **Low-rank Matrix Extraction**: Here the given matrix is supposed to be the low-rank matrix superposed with a sparse noise matrix, and the aim is to extract the low-rank matrix. This problem is written as

$$\min_X rank(X) + ||S||_0 \textbf{ such that } M = X + S \tag{2.9}$$

$||S||_0$ is the number of non-zero elements of S, and trying to minimize it is equivalent to making S sparse, as desired.

The most important application of this problem is in *background subtraction* [14] in surveillance videos. Each column corresponds to a frame. The low-rank matrix captures the background which is still, and hence almost all the columns of this matrix should be identical. The sparse noise matrix captures the foreground.

Other versions of this problem include cases where some columns are arbitrarily corrupted, in addition to missing entries. The task is to detect the corrupted columns, and fill up the missing entries in the remaining columns.

2.5.1 Convex Optimization Approach

Most common approaches to the above problems make use of convex optimization. Since rank and sparsity are both non-convex functions, some convex relaxations are needed. It is noted that the rank of a matrix is equal to the number of non-zero singular values, i.e. the ℓ_0 norm of the singular value matrix. It is known that ℓ_1 norm is a good convex relaxation of the ℓ_0 norm, hence the ℓ_1 norm of the singular values (sum of absolute values), also called the nuclear norm ($||X||_*$) is used as the convex relaxation of rank in almost all the papers.

Robust Principal Component Analysis(RPCA) [14] is a procedure for *low-rank matrix extraction*. It considers that a fully observed matrix M is the sum of a low-rank matrix X and a sparse matrix S, and attempts to recover both (X, S) from M. For sparsity, the non-convex ℓ_0 norm is replaced by the convex ℓ_1 norm. The problem is posed as follows:

$$\min_{X,S} ||X||_* + \|S\|_1 \quad \text{where} \quad X + S = Y \tag{2.10}$$

where $||X||_*$ denotes the *nuclear norm*. The formulation has proven to be successful in applications like Background subtraction and face recognition [59].

The seminal papers [13] [12] about low-rank matrix completion solved this problem using Semi-definite programming. They also provided some theoretical results, where they proved that some matrices that satisfy certain properties can be exactly recovered with high probability,

when a sufficient fraction of its entries are observed. Later approaches to low-rank matrix completion approaches usually relax the equality constraint on the observed entries. Instead they use it in the objective function, to minimize the Frobenius norm difference on the observed entries. The program is reduced to:

$$\min_X ||X||_* + \gamma||X - M||_\Omega^2 \tag{2.11}$$

where γ balances the two parts. This problem is often solved by proximal algorithms, which include thresholding the singular values [11]. OPTSPACE [42] is aimed for low-rank matrix completion from an incomplete noisy matrix Y. This proceeds by repeatedly pruning extra entries, projecting the pruned matrix into a low-rank space and minimizing the objective function 2.12, where U, V are orthogonal matrices.

$$F(U, V) = \min_S \sum_{(i,j)\in\Omega} (Y_{ij} - (USV^T)_{ij})^2 \tag{2.12}$$

The paper [15] considers the case where some of the columns are corrupted, i.e. the noise matrix is *column-sparse*. So instead of the sparsity-inducing ℓ_1 norm used for RPCA, they use the column-sparsity inducing $\ell_{2,1}$ norm, which is a mixed norm. It attempts to induce sparsity in the vector whose entries are the ℓ_2 norms of the columns, in effect trying to set most of the columns to zero. The program is given by

$$||X||_* + \lambda||S||_{2,1} \textbf{ such that } M_\Omega = X_\Omega + S_\Omega \tag{2.13}$$

The problem is solved using Augmented Lagrange Multiplier (ALM) method.

A completely different approach is considered in [51], where the matrix X is written as $X = AB^T$, where A,B are *thin* matrices, so that X has low rank. This method proceeds by estimating A and B by solving a series of linear equations, depending on the observed entries. Apart from the standard nuclear norm for rank and ℓ_1 norm for sparsity, other kinds of norms are explored in [57].

2.5.2 Bayesian Approach

These methods take a probabilistic view of Equation 2.10 and try to impose the low rank constraints as priors. In Bayesian Robust PCA (BRPCA) [23] X is modeled as $X = U(DZ)W$ where U,V are orthogonal matrices of singular vectors, D is the diagonal matrix of singular values, and Z is a binary matrix with Beta-Bernoulli prior, to sparsify the singular values of

X and reduce its rank. The same Beta-Bernoulli prior is also used for S. This paper discusses handling Markovian dependency between columns of S. Various priors for the singular values are explored in [78].

Sparse Bayesian Matrix Recovery (SBMR) [5] employs Variational methods, and models the desired matrix as the product of two matrices, $X = MN^T$, thus limiting the rank. Gaussian priors are put on the columns of M and N independently. This approach can handle both the problems of matrix extraction and completion. This method is significantly faster than BRPCA.

2.5.3 Constrained Clustering

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[1]. The constraints can be hard [81] or soft/probabilistic [48]. Constrained Spectral Clustering has also been studied recently [82] [41], which allow constrained clustering of datapoints based on arbitrary similarity measures.

[1]A detailed survey is found in *http://www.cs.albany.edu/davidson/Publications/ KDDSlides.pdf*

Chapter 3

Concise Tracklet Representation for Entity Tracking in Videos

3.1 About this Chapter

In this chapter, we discuss concise representation of video tracklets by exploiting feature-level temporal coherence in videos. For this purpose, we have used the Covariance Matrix descriptor- a robust and concise region descriptor for videos. The main application we considered is tracking of entities in videos, which is robust to illumination conditions. This is particularly relevant in the context of surveillance. Also, the proposed representation for video tracklets can be useful in many other video applications.

Publications: This work has been published in IEEE International Conference on Accoustics, Speech and Signal Processing (ICASSP), 2012 held in Kyoto, Japan.

1. Adway Mitra, Anoop K.R., Ujwal Bonde, Chiranjib Bhattcharyya, K.R. Ramakrishnan. *Eigen-Profiles of Spatio-temporal Fragments for Adaptive Region-based Tracking*, ICASSP 2012

3.2 Introduction

Tracking of entities in video is important in many applications, especially surveillance. It is especially important for tracking methods to be robust to abrupt illumination changes and poor illumination condition. This can be achieved only by using suitable features and descriptors for appearance modeling of the target. At the same time, the appearance model for the target must also be *adaptive*, i.e. should be progressively updated to account for gradual changes in appearance due to changing camera angle, pose and lighting condition. This requires a

STF/tracklet- collection of the most recent detections of the target within a short temporal window, based on which the updated models can be computed. These detections are used to build the updated appearance model for the target. The update is done either in every frame, or after short intervals.

It is a challenge to build an effective appearance model using robust features which can also be updated very efficiently. It is known that invariance to illumination can be achieved by representing images with filter outputs as features [84].Region descriptors like Covariance matrices of Gabor features are known to be robust to illumination, but need to be estimated properly for effective update of the model. Also, in a new frame, several candidate target SFs are sampled based on the target's movement model, and we need an effective measure to compare each candidate SF with the target appearance model. Existing tracking methods based on Covariance Matrices like Covariance Tracker [61] and Incremental Covariance Tensor Learning [90] keep updating the entity appearance models in each frame based on STFs, i.e. the target detections in the past few frames. Covariance Tracker computes the Intrinsic Mean of the covariance matrices from the individual target SFs in the most recent frames (in a TF), and uses this as the tracklet covariance matrix. Candidate locations are compared to this model using the distance between Covariance Matrices on the Riemannian manifold. The intrinsic mean and the distances on the manifold are not efficient to compute, and the model cannot adapt well to abrupt changes in illumination. ICTL is equivalent to pooling together the features from all the target SFs in the TF, and estimating a covariance matrix for the tracklet, which gives away the temporal information. This is also found to be incapable of adapting to abrupt changes, though it can be computed more efficiently than intrinsic mean.

In this chapter, we propose *Eigenprofile*: a novel descriptor for Spatio-Temporal Fragments (STF)/tracklets. Estimation of EP is equivalent to joint diagonalization of covariance matrices from SFs in the individual frames, and EP is a set of orthonormal vectors. We incrementally build models for the target using EP, making use of *Feature-level Temporal Coherence* property. The second-order statistics of these STFs form our target model, which is estimated using the EP. The tracking proceeds by continuously adapting STF models from target SFs over sliding TFs, and matching candidate SFs in each new frames to the current STF model, by KL-divergence. This model is easier to compute than intrinsic mean, and is also seen to adapt well to abrupt illumination changes. The main contribution of this work is to propose an illumination-invariant STF representation which can be efficiently used for tracking, but can also be used as a robust representation for tracklets for various other applications.

3.3 Eigenprofiles

A SF is a rectangular region in a frame, and encloses several pixels. At all or some of these pixels, p-dimensional feature vectors can be computed. These features can be pixel intensity values, pixel coordinates, image gradients, filter outputs and various others, or combinations of several types. The *sample covariance matrix* of these p-dimensional vectors measured within the SF forms the Covariance Matrix descriptor for the SF.

Consider a TF of K frames, where we have Covariance Matrices $C_{t+1}, C_{t+2}, \ldots, C_{t+K}$ for corresponding SFs. Due to feature-level temporal coherence, p-dimensional feature vectors in corresponding SFs in the individual frames within a TF are likely to be similar. We observe empirically that they have *almost identical principal components*. These principal components are nothing but the eigenvectors of the corresponding covariance matrices, ordered with respect to the eigenvalues. Hence, we propose to approximate the eigenbases of the $\{C_k\}$ matrices with a common eigenbasis which we call the $Eigenprofile$ of that STF.

3.3.1 Estimation of the Eigenprofile

Each SF covariance matrix within a TF can be expressed completely with its eigenvectors and eigenvalues as $C_k = \sum_j \delta_{kj} e_{kj} {e_{kj}}^T$. Under our hypothesis, it can be approximated by shared eigenvectors as

$$C_k \approx \sum_{j=1}^{p} \delta_{kj} \beta_j {\beta_j}^T \tag{3.1}$$

Here the β vectors form the EP for the STF obtained by stacking these K SFs. Estimation of EP is nothing but *Approximate Joint Diagonalization* of the $\{C_k\}$ matrices. There is a family of Approximate Joint Diagonalization algorithms, of which one is by Pham [60]. Given the C_k matrices, this algorithm attempts to find a single matrix V to minimize the following function

$$\sum_k (log(det(diag(V^T C_k V))) - log(det(V^T C_k V))) \tag{3.2}$$

However these algorithms do not make use of similarity of eigenvectors of the input matrices in any way. Here, we propose to use this additional information to make an improved estimate. We formulate the optimization problem as

$$\min_{\beta} \sum_{k=t+1}^{t+K} \|C_k - \sum_{j=1}^{p} \delta_{kj} \beta_j \beta_j^\top\|_F^2 \textbf{ such that} \tag{3.3}$$
$$\beta_j^T \beta_j = 1 \forall j \textbf{ and } \beta_j^T \beta_i = 0 \forall i \neq j$$

(3.4)

Writing the Lagrangian dual and solving it with respect to β, we have

$$\beta_j{}^T D_j \beta_j = \alpha_j \tag{3.5}$$

where $D_j = \sum_{k=t+1}^{t+K} 2\psi_{kj} C_k$, which is a symmetric matrix. The program is not convex, but a local solution is obtained when we have β_j as an eigenvector of D_j, for every j. Then we require the D_j matrices for estimating the Eigenprofile. But, we would like to have an estimate of EP from the eigenvectors of the $\{C_k\}$ matrices directly so that we do not need to store the entire matrices. To solve this, we use the observation that the corresponding eigenvectors of the $\{C_k\}$ matrices are quite identical to each other, i.e. $e_{(t+1)j} \approx e_{(t+2)j} \approx \cdots \approx e_{(t+K)j}$. Hence, we solve the following optimization problem

$$\min \sum_{k=t+1}^{t+K} ||u_j - e_{kj}||^2 \quad \textbf{subject to} \tag{3.6}$$

$$u_j^T u_j = 1 \tag{3.7}$$

The solution to this is an estimate of the i-th eigenvector of D_j, and is given by

$$u_j = \frac{\sum_k e_{kj}}{\sqrt{(\sum_k e_{kj})^T (\sum_k e_{kj})}} \tag{3.8}$$

It is to be noted that the estimates u_j of β_j thus obtained do not satisfy the orthogonality criteria, as required by the definition of Eigenprofile. So, we orthonormalize them by Gram-Schmidt procedure, to obtain orthonormal $\{\beta_j\}$.

3.4 Estimation of STF Covariance Matrix

For the tracking application, we build the Covariance Matrix C of the STF as the target model. We posit that C will have the Eigenprofile β as eigenvectors. Hence C is given by $C = \sum_j \sigma_j \beta_j \beta_j{}^T$. So we are now left with the estimation of its *eigenvalues* σ_j to learn it completely.

3.4.1 Maximum Likelihood Estimate: EP-ML

We estimate the STF Covariance Matrix C using Maximum-Likelihood Estimate (**EP-ML**). Within a temporal fragment, the feature vectors in corresponding spatial fragments of individual frames should follow the same distribution. It is known that sample Covariance Matrices of sample populations drawn from the same distribution follow the Wishart Distribution. Assuming these sample covariance matrices C_k are I.I.D., the probability of this set is given by

$$p(\{C_k\}|C) = T\frac{\prod_k |C_k|^{\frac{-p}{2}} e^{(-\frac{1}{2}(trace(C^{-1}\sum_k C_k))}}{|C|^{\frac{K}{2}}} \tag{3.9}$$

By differentiating with respect to σ_j and equating to 0, the M.L.E. of the eigenvalues σ_j from Equation 3.9 is given by

$$\sigma_j = \frac{\sum_k {\beta_j}^T C_k \beta_j}{K} \approx \frac{\sum_k \delta_{kj}}{K} \tag{3.10}$$

3.4.2 Low-Rank Approximation of STF Covariance Matrix

For many features, including Gabor Features which we have used in our experiments,it is observed that the leading eigenvalues of Covariance Matrices of the SFs are considerably larger compared to the rest, which rapidly trail off towards zero. The equation 3.10 shows that the same has to hold for the eigenvalues of the STF Covariance Matrix, and so it is possible to approximate the STF Covariance Matrix with only its R leading eigenpairs, as

$$C_{low} = \sum_{1 \le j \le R} \sigma_j \beta_j {\beta_j}^T \tag{3.11}$$

Thus, for p-dimensional features, STF model now consists of the STF Mean Vector μ, R EP-vectors β of dimension p, and R eigenvalues σ. Moreover we need not store the ST matrices C_k from the frames, but only the R leading eigenvalues δ_k, the corresponding eigenvectors e_k and the mean vector μ_k of the SF. The mean vector μ for STF can be easily obtained from the SF mean vectors μ_k in the individual frames of the TF, as $\mu = \frac{\sum_k n_k \mu_k}{\sum_k n_k}$, n_k being number of feature vectors in the SF in k-th frame. Thus such an approximation of the matrix results in some storage efficiency, especially when R is considerably lower than p.

3.5 Tracking

We now proceed to describe the framework of tracking we used in the experiments. As the main aim of the work is to build a model and not a tracker, we restrict ourself to a simple but effective tracking framework.

3.5.1 Spatio-Temporal Fragments

As mentioned, in our tracking experiments we use multiple spatially overlapping fragments to model the target. We build **9 STF models**. If in a particular frame the entity is known to be located inside a tight rectangle centered at (x, y) with length and breadth (dx, dy), the mean vector and SF Covariance Matrix of features from this rectangular SF are used to build the **Central Model**, and 8 **Peripheral Models** are obtained from the Mean Vectors and SF Covariance Matrices of the rectangular SFs centered at $(x+\delta_x dx/2, y+\delta_y dy/2)$ with dimensions (dx, dy), where $\delta_x, \delta_y \in \{1, 0, -1\}$.

3.5.2 Comparison of Region Models

During tracking, given any new frame, we need to compare the SFs at the candidate locations against the target model(s), and report the location where the matching is the best. This requires a measure to compare the STF model(s) to the candidate SF model(s). In case of our EP-based method, a STF model consists of STF Mean Vector and STF Covariance Matrix . We use the KL-Divergence as the measure of dissimilarity. In case of Covariance Tracker, [61], the measure is the **Geodesic Distance (GD)** between Covariance Matrices. The ICTL method [90] also uses GD between Covariances. For ICTL, we have used both GD and KL-Divergance (ICTL2) in our experiments.For Pham's Algorithm of Joint Diagonalization [60], the measure in Equation 3.2 is used. In this case, the STFs are represented by the V matrix outputted by the algorithm. Since we have 9 STF models $R_1, R_2, \ldots, R_9$ as mentioned above, at each candidate location (x, y) we get 9 candidate SF models $C_1, C_2, \ldots, C_9$. We compare the candidate models to the corresponding STF models to get a final score $f(x, y) = \sum_{i=1}^{9} KL(R_i, C_i)$. In cases where GD or Equation 3.2 is used, the function f is modified suitably.

3.5.3 Tracking Algorithm

We have chosen a simple random walk model. Suppose at any instant i, the location of the entity is given by $Z = (x, y)$. The target is localized using a rectangular box centered at Z. Then the candidate locations for the next frame are sampled from the distribution

$$p(Z_{i+1}|Z_i) = N(Z_{i+1}; Z_i, \Sigma) \tag{3.12}$$

The algorithm is described in details in the adjacent box.

Algorithm 1 Tracking Algorithm

Initialize the locations X_1, X_2, X_3, X_4, X_5 of the target and its size (δ_1, δ_2) in the first 5 frames.

Crop out 9 rectangular SFs around X_i and calculate their mean and SF Matrices for $1 \leq i \leq 5$.

Estimate the 9 STF models and save them.
for $i = FirstFrame : LastFrame$
choose N candidate locations
for $i = 1 : N$
Crop out 9 rectangular SFs corresponding to central and peripheral models around candidate location X_i
Build the candidate SF models $C_1, C_2, \ldots, C_9$ from these.
Calculate $f(X_i)$ with the respective STF models $R_1, R_2, \ldots, R_9$
end for
Set the location to (X^*) where $f(X^*)$ is minimum among all candidate locations
Re-estimate the 9 STF models by replacing the oldest frame in the TF with the current one
end for

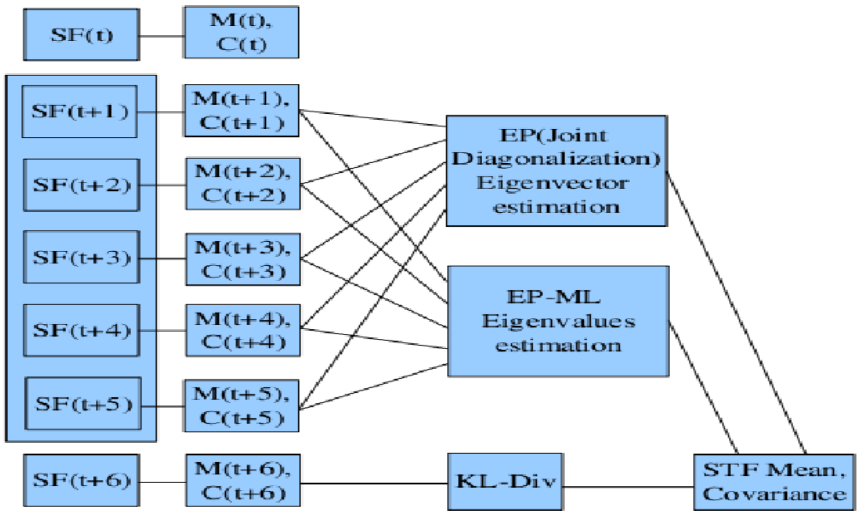


Figure 3.1: schematic diagram of the tracking. We use SF models from a TF of 5 successive frames to be build STF model, and compare candidate SFs from the next frame with it.

3.6 Experimental Evaluation

3.6.1 Datasets and Features

We have carried out experiments on 9 datasets. Of these 2 are standard and 7 captured by us. We have used one sequence (SEQ1) from PETS2000. SEQ2 is the publicly available Toni dataset [1] which involves tracking the face of a person in an obscure room. The person also turns his head and there is a sudden illumination change. SEQ3 and SEQ4 are indoor videos of a person walking on a long corridor with a single light. In SEQ3 the person walks into an obscure area with sharp illumination gradient and in SEQ4 initially the light is off, then it comes on and finally goes off again. The background also changes considerably. SEQ5-SEQ9 are all outdoor videos captured at night. In all the cases there is minimal lighting, and it is difficult to distinguish the target from the background clearly. In SEQ6, in the beginning the person moves before an unevenly lit background. Moreover, there is a sudden illumination change of the target due to the lights of a passing vehicle. The videos are of varying length with as short as 39 frames (SEQ1) to as long as 600 (SEQ2). We used 12-dimensional Gabor Features (3scales, 4orientations) for the sequences. In the eigenprofile-based method, only the leading 3 eigenvectors are used for low-rank approximation of the STF Covariance Matrix.

3.6.2 Benchmark Methods and Results

Since the proposed approach is region-based, and uses Covariance Matrices to model regions, we compare with related approaches like Covariance Tracker and ICTL. We also compare against a variant of ICTL (which we call ICTL2) that uses KL-Divergance rather than Geodesic Distance as the dissimilarity measure. Again, as EP is obtained by Joint Diagonalization, we compare with an alternative JD algorithm ([60]). **All these experiments were performed under the same basic framework of features and Tracking Algorithm, with only the STF model differing across the methods.** Moreover, we also quote results on IVT as in [68] which is not covariance-based, but well-known.

In the Ground Truth, the target's locations are specified by a tight rectangle around it. During tracking also, the method marks the inferred region with a rectangle. The number of frames in which the overlap of these two rectangles is 0 is provided in Table 3.1. Finally the drift in terms of number of pixels obtained by the methods on the 9 datasets is provided in Figure 3.2.

It can be seen that our method EP-ML achieve the best performance in all the 9 sequences.

[1] http://www.svcl.ucsd.edu/projects/tracking/results.html

SEQ	EP-ML	IVT	COV	ICTL	ICTL2	Pham
1	**0**	**0**	0.37	0.02	**0**	0.63
2	**0.07**	0.68	0.70	0.70	0.16	0.38
3	**0**	0.50	0.50	**0**	**0**	0.50
4	**0**	0.72	0.70	**0**	**0**	0.70
5	**0**	0.65	**0**	**0**	**0**	0.08
6	**0**	0.42	0.93	0.93	0.93	0.93
7	**0**	**0**	0.67	0.72	**0**	**0**
8	**0**	0.24	0.41	0.49	0.36	0.33
9	**0**	0.72	0.86	0.50	0.50	0.78

Table 3.1: Fraction of frames in the videos where the output's overlap with Ground Truth is 0 pixels.

SEQ	EP-ML	IVT	COV	ICTL	ICTL2	Pham
1	0.94	2.18	138	1.6	5.0	375
2	4.8	56.1	38.85	41.15	**3.09**	7.35
3	**0.03**	1.46	6.09	0.1	0.1	1.5
4	**0.35**	22.17	31.96	1.8	0.39	30.86
5	0.43	44.25	0.33	**0.1**	0.25	1.50
6	**0.94**	4.96	191	193	148.5	149
7	0.44	X	12.36	14.96	0.43	0.95
8	**0.29**	2.75	3.11	3.07	1.37	0.3
9	**0.41**	25.5	32.69	3.42	4.03	23.94

Table 3.2: Average deviation of target location inferred by tracking from the ground truth, normalized by target size. If target size is (X, Y) and deviation at the n-th frame is $\delta(x_n), \delta(y_n)$, the measure is $\sum_n((\frac{\delta(x_n)}{X}^2) + (\frac{\delta(y_n)}{Y}^2))$

The success of our methods over ICTL2 indicates that our representation as a whole, rather than the measure is the main cause of success of our methods. On the other hand, ICTL2 does perform better than the Geodesic-distance based methods (COV and ICTL1), which shows that K-L Divergence is a better measure. Of course, on SEQ5 the other covariance-based methods also succeed, while in SEQ7 IVT and ACM(K-L Divergance) also succeed. On SEQ4 and SEQ6 most methods fail at a position of background change, and in SEQ3 they fail at the point the person moves into the dark portion of the corridor. In SEQ6 only EP-ML succeeds, while the rest fail at the unevenly lit background. SEQ8 is the toughest sequence as the person is hardly distinguishabe from the background, and the benchmarks lose track at some point or other, unlike the proposed methods. The code and data are available in *http* : *//clweb.csa.iisc.ernet.in/adway/tracking/*.

Figure 3.2: SEQ1: EP-ML,ICTL and Cov. Tracker from top to bottom. There is a sudden background change, EP-ML, ICTL succeed unlike Covtrack

Figure 3.3: SEQ3: The results shown are for EP-ML,ICTL and Covariance Tracker from top to bottom. COV losse track at the illumination gradient in the middle

Figure 3.4: SEQ4: EP-ML,ICTL and Cov. Tracker from top to bottom. There is a sudden illumination change, EP-ML, ICTL succeed unlike Covtrack

Figure 3.5: SEQ6: EP-ML,ICTL and Cov. Tracker from top to bottom. The video is dark and blurred, EP-ML succeeds unlike the rest

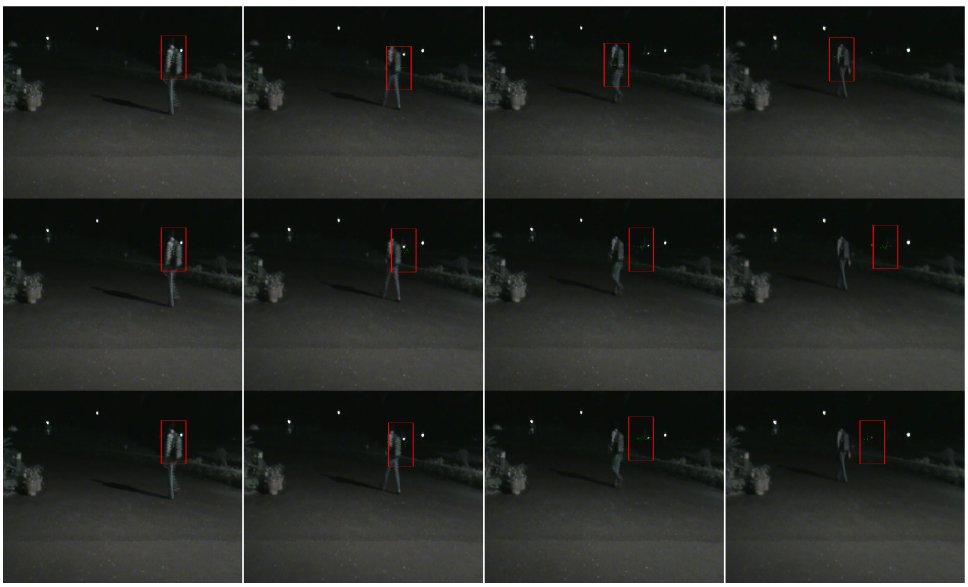


Figure 3.6: SEQ7: EP-ML,ICTL and Cov. Tracker from top to bottom, for frames 6, 31, 55 and 70 of total 112 frames. EP-ML succeeds unlike the rest

3.7 Implementation Details

Data Collection Tracking is a well-studied problem, and there are several well-known datasets that have been frequently used. Different datasets are useful for various tracking scenarios. For example, PETS and CAVIAR datasets are suitable for surveillance scenarios, while the "Dudek" and "Toni" are more suitable for close-range face tracking. In this work, we tested our method, and the alternatives against all these, though the proposed method is not tailor-made for any particular scenario like surveillance or face-tracking. The most prominent aspect of the proposed method is that it is invariant to illumination, and is also capable of working in relative darkness. So, in the absence of any dataset that is suitable for this task, we shot appropriate videos ourselves. They were shot at various locations within the Indian Institute of Science campus, and one of the participants of this project was the target of tracking. In all cases, the camera was mounted on a stationery tripod stand, while the target moved around. In some videos like SEQ3, the target moved across a corridor from a well-lit portion to a relatively dark portion. In SEQ4, the target moved along a corridor while the light was turned on and off repeatedly. The aim was to test if the tracking algorithm was hampered by these illumination changes. SEQ5, SEQ6, SEQ7 and SEQ8 were shot in relatively dark areas in the evening.

Features We used the same set of features (Gabor features with 3 scales and 4 orientations) for EP-ML, ICTL1, ICTL2, Pham and Covariance Tracker. The covariance matrices obtained were thus 12×12. The number of scales and orientations were fixed by tuning over two test sequences. It was seen that using too few scales and orientations reduced accuracy, while using too many slowed down the computation without a proportionate improvement in performance. In case of IVT, the publicly available code was used. This is not a covariance-based methods, and uses the pixel values rather than Gabor features. In scenarios which do not involve illumination issues (like CAVIAR or DUDEK) we also tried using image gradients and pixel values (greyscale and RGB) instead of Gabor features, and continued to get decent results. However, the results reported in the tables were all obtained using Gabor features.

Implementation The system was implemented in MATLAB. We used publicly available code for computing Gabor features, and wrote our own code for target model computation, candidate region evaluation as well as model updation. It was found that computation of Gabor features is time-consuming, and it slowed down the performance of the tracker to about 1 frame per second, though the eigenprofile computation and matching took just about 0.1 seconds per frame. In comparison IVT was quicker, at about 4-5 frames per second, but its performance was much lower, especially on the sequences involving illumination change which require Gabor features. It is hoped that implementation of the system in OpenCV will enable

close to real-time performance. Modern cameras have frame rates of more than 25 frames per second, making it increasingly difficult to process all frames in real time. However, the *temporal coherence* feature of videos comes to our aid. Due to this property, in successive frames the target is likely to be in very close locations, especially if the frames are taken at very short time-difference (which happens in case of high frame rate). Thus it is not really necessary to process each and every frame, we can process frames at regular intervals and interpolate the target's locations in the intermediate frames.

3.8 Applications, Limitations and Extensions

The proposed system can have many applications- most prominently in surveillance settings. It is particularly important in outdoor surveillance of streets in evenings and nights. However, as already established, it can work well in any tracking environment. The main computational bottleneck of the system is the Gabor feature extraction. But if we use simpler features like pixel values and image gradients the system has very little computation and memory requirements, and can be run on simple computation devices like cameras and mobile phones.

One issue with the current system is *initialization*. Like several research papers on tracking this work assumes that the location of the target is known in the first 5 frames. In our implementation this is done manually- whenever the code is run with a test video, the first 5 frames are displayed successively, and the user is prompted to mark the target's locations using the *imcrop* tool of Matlab. Obviously, this cannot be done if the system is to be deployed for surveillance. The various initialization schemes are as follows:

1. Maintain a background model, with which each new frame can be compared. If there is a target to be tracked, its location can be found automatically by frame differencing. But this either assumes that the background will remain unchanged all the time (which is infeasible in most applications, most notably due to varying illumination conditions across day and night), or we need multiple background models.

2. Much work has been done on *foreground-background separation* which define foreground as entities that move and background as those that remain stationery. Movement within the field of view of the camera can be detected using *optical flow*, and the target can thus be localized.

3. In some tracking works [49], *visual saliency* is used for automatic tracker initialization. This is another alternative.

Extensions: The most recent papers on tracking mostly address *multi-target tracking*, where there are multiple targets per frame, to be tracked simultaneously. In the experiments presented here, we have always considered a single target, but multi-target tracking can also be handled using the proposed technique, in a straightforward way- we simply need to maintain appearance models (eigenprofiles and STF covariance matrices) for each target.

Chapter 4

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

4.1 About this Chapter

In this chapter, we discuss our work related to Bayesian nonparametric models for modeling semantic-level temporal coherence in videos. The main application we have considered is entity discovery from TV series/movie videos downloaded from Youtube, that have no additional information such as dialogue scripts. We have also discussed entity-driven summarization of the videos. These are both novel applications, with particular relevance to concise, user-friendly representation of videos in video-sharing websites. On the theoretical side, we have proposed two generative models: Temporally-Coherent Chinese Restaurant Process (TC-CRP) and Temporally-Coherent Chinese Restaurant Franchise (TC-CRF).

Publications: The proposed TC-CRP model, and much of the experimental results presented here have been published in SIAM Data Mining Conference (SDM), 2015 held in Vancouver, Canada. The TC-CRF model, and the rest of the experimental results have been included in its journal version which has been accepted in IEEE Transactions on Pattern Analysis and Machine Intelligence.

1. Adway Mitra, Soma Biswas, Chiranjib Bhattacharyya. *Temporally Coherent CRP: A Bayesian Non-Parametric Approach for Clustering Tracklets with applications to Person Discovery in Videos*, SDM 2015

2. Adway Mitra, Soma Biswas, Chiranjib Bhattacharyya. *Bayesian Modeling of Temporal*

Coherence in Videos for Entity Discovery and Summarization, IEEE Transactions on Pattern Analysis and Machine Intelligence (2016)

4.2 Introduction

Most videos uploaded on the internet are centered around a few *entities* of a particular type, which appear repeatedly. For example, a movie or TV-series episode is centered around a few persons, each of whom is an entity. Similarly, a video about a Formula-one race is centered around a few cars, each of which is an entity. *Entity Discovery* is the task of describing each entity by some appropriate representation, and finding all the locations in the video where it appears. The input to this problem is the set of detections of the entities in all the frames of the video, using a suitable detector.

One specialization of this problem which has recently received a lot of attention is *automated discovery of persons from videos of movies or TV-series episodes*. Each person is represented by his/her *face*, so here the input is the set of all face detections in the video. Various relaxations have been made to simplify the task. Automatic Cast Listing [4] aims at providing a representative subset of the detections, that is expected to include all the persons. However it does not provide the list of all occurrences of the persons. Another line of work [98][75] makes use of *textual movie scripts* which provides close to frame-by-frame details of the dialogues, including the character names. These works aim to align the scripts with the frames, and the detections with character names. The problem with this approach is that, such scripts or any similar meta-data are not available for most videos uploaded by users on sites like Youtube. The task of Face/Track Clustering [89] in videos can avoid this issue, but even then we need to know in advance the number of clusters to be formed, which is generally not possible. Moreover, none of these methods are capable of performing the task *online*, which is required in case of *streaming videos.*

In this chapter, we pose the entity discovery problem as *tracklet clustering*, as done in [88]. Our goal is to design algorithms for tracklet clustering which can work on long videos. Tracklets[36] 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. 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 a *Bayesian non-parametric* clustering approach to Tracklet clustering and study its effectiveness in automated discovery of entities with all their occurrences in long videos. The main challenges are in modeling the spatio-temporal properties. To the best of our knowledge this problem has not been studied either in Machine Learning or in Computer

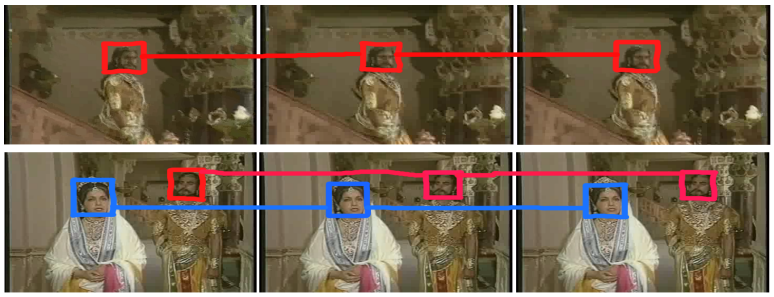


Figure 4.1: Top: a window consisting of frames 20000,20001,20002, Bottom: another window- with frames 21000,21001,21002. The detections are linked on spatio-temporal basis to form tracklets. One person (marked with red) occurs in both windows, the other character (marked with blue) occurs only in the second. The two red tracklets should be associated though they are from non-contiguous windows

Vision community.

4.3 Problem Definition

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* [36] and then linking the tracklets from successive windows to form tracks. The *short-range association of tracklets* to form tracks is known as *tracking*. But in a TV series video, the same person may appear in different (non-contiguous) parts of the video, and so we need to associate tracklets on a *long-range* basis also (see Figure 4.1). 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. 4.3). Additionally, *overlapping tracklets (that span the same frames), cannot belong to the same entity.* A tracklet can be easily

Figure 4.2: 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

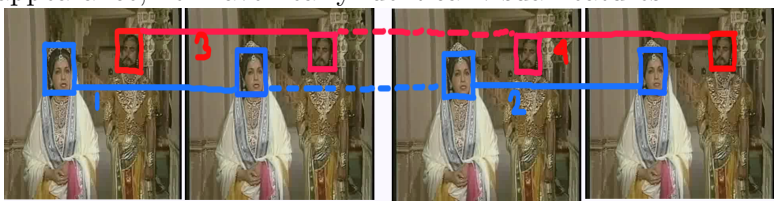


Figure 4.3: TC at Tracklet level: Blue tracklets 1,2 are spatio-temporally close (connected by broken lines), and belong to the same person. Similarly red tracklets 3 and 4.

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.

4.3.1 Notation

In this work, given a video, we fix beforehand the *type of entity* (eg. person/face, cars, planes, trees) we are interested in, and choose an appropriate detector like [80] [27], which is run on every frame of the input video. The detections in successive frames are then linked based on spatial locality, to obtain tracklets. At most R detections from R contiguous frames are linked like this. The tracklets of length less than R are discarded, hence all tracklets consist of R detections. We restrict the length of tracklets so that the appearance of the detections remain almost unchanged (due to detection-level TC), which facilitates tracklet representation. At $R = 1$ we work with the individual detections.

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

The tracklets can be sorted topologically based on their starting and ending frame indices. 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 $CF(i)$ which span frame(s) that overlap with the frames spanned by i. Each detection (and tracklet) is associated with an entity, which are unknown in number, but presumably much less than the number of detections (and tracklets).

These entities also are represented by vectors, say $\phi_1, \phi_2, \ldots, \phi_K$. Each tracklet i is associated with an entity indexed by Z_i, i.e. $Z_i \in \{1, 2, \ldots, K\}$.

4.3.2 Entity Discovery

Let each video be represented as a sequence of d-dimensional vectors $\{Y_1, \ldots, Y_N\}$ along with the set $\{prev(i), next(i), CF(i)\}_{i=1}^N$. We aim to learn the vectors $\{\phi_1, \phi_2, \ldots, \}$ and the assignment variables $\{Z_i\}_{i=1}^N$. In addition, we have *constraints* arising out of *temporal coherence* and other properties of videos. Each tracklet i is likely to be associated with the entity that its predecessor or successor is associated with, except at shot/scene changepoints. Moreover, a tracklet i cannot share an entity with its conflicting tracklets $CF(i)$, as the same entity cannot occur twice in the same frame. This notion is considered in relevant literature [92] [89]. Mathematically, the constraints are:

$$\begin{aligned} Z_{prev(i)} = Z_i = Z_{next(i)} \forall i \in \{1, \ldots, N\} \ \ w.h.p. \\ Z_i \notin \{Z_j : j \in CF(i)\} \forall i \in \{1, \ldots, N\} \end{aligned} \tag{4.1}$$

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

Finally, the video frames can be grouped into short segments, based on the starting frame numbers $F(1), F(2), \ldots, F(N)$ of the N tracklets. Consider two successive tracklets i and $(i+1)$, with starting frames $F(i)$ and $F(i+1)$. If the gap between frames $F(i)$ and $F(i+1)$ is larger than some threshold, then we consider a new temporal segment of the video starting from $F(i+1)$, and add $i+1$ to a list of *changepoints* (CP). The beginning of a new temporal segment does not necessarily mean a scene change, the large gap between frames $F(i)$ and $F(i+1)$ may be caused by failure of detection or tracklet creation. The segment index of each tracklet i is denoted by $S(i)$.

4.4 Generative Process for Tracklets

We now explain our Bayesian Nonparametric model TC-CRP to handle the spatio-temporal constraints (Eq 4.1) for tracklet clustering, and describe a generative process for videos based on tracklets.

4.4.1 Bayesian Nonparametric modeling

In Section 4.3, we discussed the vectors $\phi_1, \phi_2, \ldots$ each of which represent an entity. In this chapter we consider a Bayesian approach with Gaussian Mixture components $\mathcal{N}(\phi_k, \Sigma_1)$ to account for the variations in visual features of the detections, say face detections of a person. As already mentioned, number of components K is not known beforehand, and must be discovered from the data. That is why we consider nonparametric Bayesian modeling. Also, as we shall see, this route allows us to elegantly model the temporal coherence constraints. In this approach, we shall represent entities as mixture components and tracklets as draws from such mixture components.

Dirichlet Process [28] has become an important clustering tool in recent years. Its greatest strength is that unlike K-means, it is able to discover the correct number of clusters. Dirichlet Process is a distribution over distributions over a measurable space. A discrete distribution P is said to be distributed as $DP(\alpha, H)$ over space A if for every finite partition of A as $\{A_1, A_2, \ldots, A_K\}$, the quantity $\{P(A_1), \ldots, P(A_K)\}$ is distributed as $Dirichlet(\alpha H(A_1), \ldots, \alpha H(A_K))$, where α is a scalar called *concentration parameter*, and H is a distribution over A called Base Distribution. A distribution $P \sim DP(\alpha, H)$ is a discrete distribution, with infinite support set $\{\phi_k\}$, which are draws from H, called *atoms*.

4.4.2 Modeling Tracklets by Dirichlet Process

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

$$P \sim DP(\alpha, H); X_i \sim P, Y_i \sim \mathcal{N}(X_i, \Sigma_1) \forall i \in [1, N] \tag{4.2}$$

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

Using the constructive definition of Dirichlet Process, called the Stick-Breaking Process [70], the above process can also be written equivalently as

$$\begin{aligned} \hat{\pi}_k \sim Beta(1, \alpha), \pi_k = \hat{\pi}_k \prod_{i=1}^{k-1} (1 - \hat{\pi}_{i-1}), \phi_k \sim H \ \forall k \in [1, \infty) \\ Z_i \sim \pi, Y_i \sim \mathcal{N}(\phi_{Z_i}, \Sigma_1) \forall i \in [1, N] \end{aligned} \tag{4.3}$$

Here π is a distribution over integers, and Z_i is an integer that indexes the component corresponding to the tracklet i. Our aim is to discover the values ϕ_k, which will give us the entities, and also to find the values $\{Z_i\}$, which define a clustering of the tracklets. For this

purpose we use collapsed Gibbs Sampling, where we integrate out the P in Equation 4.2 or G in Equation 4.3. The Gibbs Sampling Equations $p(Z_i|Z_{-i}, \{\phi_k\}, Y)$ and $p(\phi_k|\phi_{-k}, Z, Y)$ are given in [32]. For Z_i,

$$p(Z_i = k|Z_{-i}, \phi_k, Y_i) \propto p(Z_i = k|Z_{-i})p(Y_i|Z_i = k, \phi) \tag{4.4}$$

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

4.4.3 Temporally Coherent Chinese Restaurant Process

In the generative process (Equation 4.3) all the Z_i are drawn IID conditioned on π. Such models are called *Completely Exchangeable*. This is, however, often not a good idea for sequential data such as videos. In Markovian Models like sticky HDP-HMM, Z_i is drawn conditioned on π and Z_{i-1}. In case of DP, the independence among Z_i-s is lost on integrating out π. After integration the generative process of Eq 4.3 can be redefined as

$$\phi_k \sim H \forall k; Z_i|Z_1, \ldots, Z_{i-1} \sim CRP(\alpha); Y_i \sim \mathcal{N}(\phi_{Z_i}, \Sigma_1) \tag{4.5}$$

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 (Eq 4.1) mentioned in the previous section. 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 [53]. Here, with each Z_i we associate a *binary change variable* C_i. If $C_i = 0$ then $Z_i = Z_{prev(i)}$, i.e the tracklet identity is maintained. But if $C_i = 1$, a new value of Z_i is sampled. Note that every tracklet i has a temporal predecessor $prev(i)$. However, if this predecessor is spatio-temporally close, then it is more likely to have the same label. So, the probability distribution of change variable C_i should depend on this closeness. In TC-CRP, we use two values (κ_1 and κ_2) for the Bernoulli parameter for the change variables. We put a threshold on the spatio-temporal distance between i and $prev(i)$, and choose a Bernoulli parameter for C_i based on whether this threshold is exceeded or not. Note that maintaining tracklet identity by setting $C_i = 0$ is equivalent to *tracking*.

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

$$T(Z_i = k|Z_{1:i-1}, C_{1:i-1}, C_i = 1) = 0 \textbf{ if } k \in \{Z_{CF(i)}\} - \{0\}$$
$$\propto \beta \textbf{ if } k = 0$$
$$\propto n_{k1}^{ZC} \textbf{ if } k \in \{Z_1, \ldots, Z_{i-1}\}, k \notin \{Z_{CF(i)}\}$$
$$\propto \alpha \textbf{ otherwise} \quad (4.6)$$

where $Z_{CF(i)}$ is the set of values of Z for the set of tracklets $CF(i)$ that overlap with i, and n_{k1}^{ZC} is the number of points j ($j < i$) where $Z_j = k$ and $C_j = 1$. The first rule ensures that two overlapping tracklets cannot have same value of Z. The second rule accounts for false tracklets. The third and fourth rules define a CRP restricted to the changepoints where $C_j = 1$. The final tracklet generative process is as follows: where T is the PPF for TC-CRP, defined in Eq 4.6.

Algorithm 2 TC-CRP Tracklet Generative Process

$\phi_k \sim \mathcal{N}(\mu, \Sigma_0)\ \forall k \in [1, \infty)$
for $i = 1 : N$ **do**
  if $dist(i, prev(i)) \leq thres$ **then**
    $C_i \sim Ber(\kappa 1)$
  else
    $C_i \sim Ber(\kappa 2)$
  end if
  if $C_i = 1$ **then**
    draw $Z_i \sim T(Z_i|Z_1, \ldots, Z_{i-1}, C_1, \ldots, C_{i-1}, \alpha)$
  else
    $Z_i = Z_{prev(i)}$
  end if
  if $Z_i = 0$ **then**
    $Y_i \sim \mathcal{N}(\mu, \Sigma_2)$
  else
    $Y_i \sim \mathcal{N}(\phi_{Z_i}, \Sigma_1)$
  end if
end for

4.4.4 Inference

Inference in TC-CRP can be performed easily through Gibbs Sampling. We need to infer C_i, Z_i and ϕ_k. As C_i and Z_i are coupled, we sample them in a block for each $i \in [1, N]$ as done in [53]. If $C_{i+1} = 0$ and $Z_{i+1} \neq Z_{i-1}$, then we must have $C_i = 1$ and $Z_i = Z_{i+1}$. If $C_{i+1} = 0$ and $Z_{i+1} = Z_i$, then $Z_i = Z_{i+1}$, and C_i is sampled from $Bernoulli(\kappa)$. In case $C_{i+1} = 1$ and $Z_{i+1} \neq Z_{i-1}$, then $(C_i = a, Z_i = k)$ with probability proportional to $p(C_i = a)p(Z_i|Z_{-i}, C_i = a))p(Y_i|Z_i = k, \phi_k)$. If $a = 0$ then $p(Z_i = k|Z_{-i}, C_i = 1) = 1$ if $Z_{i-1} = k$, and 0 otherwise. If $a = 1$ then $p(Z_i|Z_{-i}, C_i = a))$ is governed by TC-CRP. For sampling ϕ_k, we make

use of the Conjugate Prior formula of Gaussians, to obtain the Gaussian posterior with mean $(n_k\Sigma_1^{-1} + \Sigma^{-1})^{-1}(\Sigma_1^{-1}Y_k + \Sigma^{-1}\mu)$ where $n_k = |\{i : Z_i = k\}|$, and $Y_k = \sum_{i:Z_i=k} Y_i$. Finally, we update the hyperparameters μ and Σ after every iteration, based on the learned values of $\{\phi_k\}$, using Maximum Likelihood estimate. κ_1,κ_2 can also be updated, but in our implementation we set them to 0.001 and 0.1 respectively, based on empirical evaluation on one held-out video. The threshold *thres* was also similarly fixed.

4.5 Generative Process for Video Segments

In the previous section, we considered the entire video as a single block, as the TCCRP PPF for any tracklet i involves (Z, C)-values from all the previously seen tracklets throughout the video. However, this need not be very accurate, as in a particular part of the video some mixture components (entities) may be more common than anywhere else, and for any i, Z_i may depend more heavily on the Z-values in temporally close tracklets than the ones far away. This is because, a TV-series video consists of *temporal segments* like scenes and shots, each characterized by a subset of persons (encoded by binary vector B_S). The tracklets attached to a segment s cannot be associated with persons not listed by B_s. To capture this notion we propose a new model: Temporally Coherent Chinese Restaurant Franchise (TC-CRF) to model a video temporally segmented by S (see Section 4.3).

4.5.1 Temporally Coherent Chinese Restaurant Franchise

Chinese Restaurant Process is the PPF associated with Dirichlet Process. Hierarchical Dirichlet Process (HDP) [76] aimed at modeling *grouped data sharing same mixture components*. It assumes a group-specific distribution π_s for every group s. The generative process is:

$$\hat{p}_k \sim Beta(1, \alpha), p_k = \hat{p}_k \prod_{i=1}^{k-1} (1 - \hat{p}_{i-1}), \phi_k \sim H \ \forall k \in [1, \infty)$$
$$\pi_s \sim p \forall s \in [1, M]; Z_i \sim \pi_{S(i)}, Y_i \sim \mathcal{N}(\phi_{Z_i}, \Sigma_1) \forall i \in [1, N] \tag{4.7}$$

where datapoint i belongs to the group $S(i)$. The PPF corresponding to this process is obtained by marginalizing the distributions p and $\{\pi\}$, and is called the *Chinese Restaurant Franchise* process, elaborated in [76]. In our case, we can modify this PPF once again to incorporate TC, analogously to TC-CRP, to have Temporally Coherent Chinese Restaurant Franchise (TC-CRF) Process. In our case, a group corresponds to a temporal segment, and as already mentioned, we want a binary vector B_s, which indicates the components that are active in segment s. But HDP assumes that all the components are shared by all the groups, i.e. any particular component can be sampled in any of the groups. We can instead try *sparse modeling* by incorporating $\{B_s\}$

into the model, as done in [85] for Focused Topic Models. For this purpose we put an IBP [33] prior on the $\{B_s\}$ variables, where $p(B_{sk} = 1|B_1, \ldots, B_{s-1}) \propto n_k$ where n_k is the number of times component k has been sampled in all scenes before s, and $p(B_{sk_{new}}|B_1, \ldots, B_{s-1}) \propto \gamma$. The *TC-CRF* PPF is then as follows:

$$\begin{aligned} TF(Z_i = k|B_s, Z_{1:i-1}, C_{1:i-1}, C_i = 1) = 0 \textbf{ if } k \in \{Z_{CF(i)}\} - \{0\} \\ = 0 \textbf{ if } B_{sk} = 0 \\ \propto \beta \textbf{ if } k = 0 \\ \propto n_{sk1}^{SZC} \textbf{ if } B_{sk} = 1, k \in \{Z\}_s, k \notin \{Z_{CF(i)}\} \\ \propto \alpha \textbf{ if } B_{sk} = 1, k \notin \{Z\}_s, k \notin \{Z_{CF(i)}\} \end{aligned} \tag{4.8}$$

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

Algorithm 3 TC-CRF Tracklet Generative Process

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

associated with tracklet i.

4.5.2 Inference

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

4.6 Relationship with existing models

TC-CRP draws inspirations from several recently proposed Bayesian nonparametric models, but is different from each of them. It has three main characteristics: 1) Changepoint-variables $\{C\}$ 2) Temporal Coherence and Spatio-temporal cues 3) Separate component for non-face tracklets. The concept of changepoint variable was used in Block-exchangeable Mixture Model [53], 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 [29], but this model does not consider change-point variables. Moreover, it uses a transition distribution P_k for each mixture component k, which increases the model complexity. Like BEMM [53] we avoid this step, and hence our PPF (Eq 4.6) does not involve $Z_{prev(i)}$. DDCRP [8] 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^Z: the number of previous datapoints assigned to component k. Hence our method is significantly different from DDCRP. Finally, the first two rules of TC-CRP PPF are novel.

TC-CRF is inspired by HDP [76]. However, once again the three differences mentioned above hold good. In addition, the PPF of TC-CRF itself is different from Chinese Restaurant Franchise as described in [76]. The original CRF is defined in terms of two concepts: tables and dishes, where tables are local to individual restaurants (data groups) while dishes (mixture components) are global, shared across restaurants (groups). Also individual datapoints are assigned mixture components indirectly, through an intermediate assignment of tables. The

concept of table, which comes due to marginalization of group-specific mixture distributions, results in complex book-keeping, and the PPF for datapoints is difficult to define. Here we avoid this problem, by skipping tables and directly assigning mixture components to datapoints in Eq 4.8. Inspiration of TC-CRF is also drawn from IBP-Compound Dirichlet Process [85]. But the inference process of [85] is complex, since the convolution of the DP-distributed mixture distribution and the sparse binary vector is difficult to marginalize by integration. We avoid this step by directly defining the PPF (Eq 4.8) instead of taking the DP route. This approach of directly defining the PPF was taken for DD-CRP [8] also.

4.7 Experiments on Person Discovery

One particular entity discovery task that has recently received a lot of attention is person discovery from movies/ TV series. We carried out extensive experiments for person discovery on TV series videos of various lengths. We collected three episodes of The Big Bang Theory (Season 1). Each episode is 20-22 minutes long, and has 7-8 characters (occurring in at least 100 frames). We also collected 6 episodes of the famous Indian TV series "The Mahabharata" from Youtube. Each episode of this series is 40-45 minutes long, and have 15-25 prominent characters. So here, each character is an entity. These videos are much longer than those studied in similar works like [88], 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 [89][88], we represent the persons with their faces. We obtained face detections by running the OpenCV Face Detector on each frame separately. As described in Section 4.3 the face detections were all converted to greyscale, scaled down to 30×30, and reshaped to form 900-dimensional vectors. We considered tracklets of size $R = 10$ and discarded smaller ones. The dataset details are given in Table 1.

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 [1]. The objects were detected using the Object-specific detectors [27]. 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.

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

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)

Table 4.1: Details of datasets

Figure 4.4: Face detections (top), and the corresponding atoms (reshaped to square images) found by TC-CRP (bottom)

4.7.1 Alternative Methods

A recent method for face clustering using track information is WBSLRR [92] based on Subspace Clustering. Though in [92] 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 [41]. 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 [14] [39]. 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) [5]. Others are either too slow (BRPCA [23]), or recover matrices with ranks too low (OPTSPACE [42]) or too high (RPCA [14]). Finally, we compare against another well-known sequential BNP method- the sticky HDP-HMM [29].

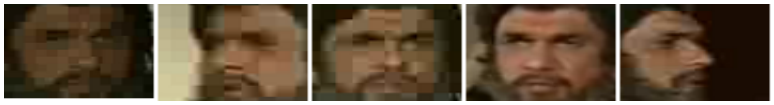


Figure 4.5: Different atoms for different poses of same person

4.7.2 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. 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 person* A(say Sheldon for a BBT video, or Arjuna for a Mahabharata video). We also declare that the cluster k and its corresponding mixture component ϕ_k corresponds to the person A. Also, then A is considered to be *discovered*. The threshold of purity was set to 70% 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 persons (entity) with at least 1 cluster (at least 10 tracklets) corresponding to them*. Next, we measure *Tracklet Coverage: the fraction of tracklets that are assigned to pure clusters*. Effectively, these tracklets are *discovered*, and the ones assigned to impure clusters are lost.

4.7.3 Results

The results on the three measures discussed above are shown in Tables 2,3,4. In terms of the three measures, TC-CRF is usually the most accurate, followed by TC-CRP, and then sHDP-HMM. This demonstrates that BNP methods are more suitable to the task. The constrained spectral clustering-based method is competitive on the purity measure, but fares very poorly in terms of tracklet coverage. This is because, it forms many small pure clusters, and a few very large impure clusters which cover a huge fraction of the tracklets. Thus, a large number of tracklets are lost.

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 (See Figure 4.5). The number of clusters formed per video by the different methods 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-CRF, TC-CRP and sHDP-HMM

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

Table 4.2: Purity results for different methods. The number of significant clusters are written in brackets

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

Table 4.3: Entity Coverage results for different methods

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.

4.7.4 Online Inference

We wanted to explore the case of streaming videos, where the frames appear sequentially and old frames are not stored. This is the online version of the problem, the normal Gibbs Sampling will not be possible. For each tracklet i, we will have to infer C_i and Z_i based on $C_{prev(i)}$, $Z_{prev(i)}$ and the $\{\phi_k\}$-vectors learnt from $\{Y_1, Y_2, \ldots, Y_{i-1}\}$. Once again, (C_i, Z_i) is sampled as a block as above, and the term $p(Z_i|Z_{-i}, C_i = a))$ follows from the TC-CRP PPF (Eq 4.6). The same thing can be done for TC-CRF also. Instead of drawing one sample per data-point, an option is

Dataset	TCCRF	TCCRP	sHDPHMM	SBMR+ ConsClus	WBSLRR
BBTs1e1	**0.82**	0.67	0.79	0.29	0.73
BBTs1e3	0.86	**0.88**	0.68	0.09	0.53
BBTs1e4	**0.92**	0.82	0.78	0.22	0.62
Maha22	**0.90**	**0.90**	0.86	0.43	0.69
Maha64	**0.93**	0.90	0.81	0.39	0.62
Maha65	**0.94**	0.85	0.91	0.40	0.68
Maha66	0.74	**0.80**	0.68	0.43	0.65
Maha81	**0.80**	0.75	0.66	0.46	0.50
Maha82	0.76	**0.81**	0.64	0.37	0.64

Table 4.4: Tracklet Coverage results for different methods

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

Table 4.5: Online (single-pass) analysis on 4 videos

to draw several samples and consider the mode. In the absence of actual streaming datasets we performed the single-pass inference (Sec 4.7.4) 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 4.7.1 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 the best on the Mahabharata videos and TC-CRF on the Big Bang Theory ones. Notably, the figures for TC-CRP and TC-CRF in the online experiment are not significantly lower than those in the offline experiment (except one or two exceptions), unlike sHDP-HMM. This indicates that the proposed methods converge quickly, and so are more efficient offline.

4.7.5 Outlier Detection / Discovery of False Tracklets

Face Detectors such as [80] are trained on static images, and applied on the videos on per-frame basis. This approach itself has its challenges [71], 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 [74].

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

Figure 4.6: 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 4.6: Discovery of non-face tracklets

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*.

4.7.6 Evaluation of TC enforcement

The aim of TC-CRP and TC-CRF is to encourage TC at the semantic level, that spatio-temporally close but non-overlapping tracklets should belong to the same entity. In the Bayesian models like sHDP-HMM, TC-CRP and TC-CRF, these cues are modeled with probability distributions, in WBSLRR with convex regularization and in constrained clustering they are encoded as hard constraints. We now evaluate how well the different methods have been able to enforce these cues. We create ground-truth tracks by linking the tracklets which are spatio-

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

Table 4.7: Fraction of ground truth tracks that are fully linked

Dataset	TCCRP	sHDPHMM	SBMR+ ConsClus	WBSLRR
Cars	0.94 (35)	0.92 (12)	**1.00** (54)	0.24 (21)
Aeroplanes	**0.95** (43)	0.87 (15)	0.84 (44)	0.21 (24)

Dataset	TCCRP	sHDPHMM	SBMR+ ConsClus	WBSLRR
Cars	**5**	**5**	**5**	2
Aeroplanes	**6**	5	**6**	4

Dataset	TCCRP	sHDPHMM	SBMR+ ConsClus	WBSLRR
Cars	0.73	0.69	**1.00**	0.04
Aeroplanes	**0.93**	0.70	0.88	0.09

Table 4.8: Purity, Entity Coverage and Tracklet Coverage results for different methods on Cars and Aeroplanes videos

temporally close to each other (with respect to the chosen threshold *thres* in the generative process), and belong to the same entity. All the tracklets in each ground-truth track should be assigned to the same cluster. This is the task of *tracklet linking*. We measure *what fraction of the these ground-truth tracks have been assigned entirely to single clusters* by the different methods. We do not compare SBMR+ConsClus, since it uses hard constraints. The results are shown in Table 7. We find that TC-CRF is the best once again, followed by TC-CRP and sHDP-HMM. WBSLRR has significantly poorer performance, though it springs a surprise on BBTs1e1.

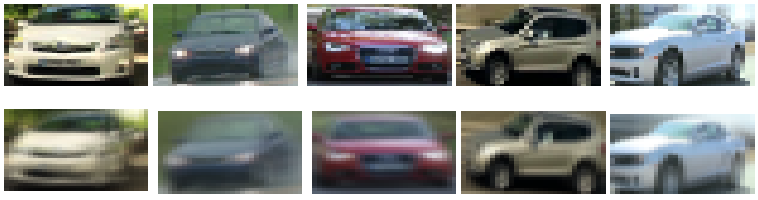


Figure 4.7: Car detections (top), and the corresponding atoms (reshaped to square images) found by TC-CRP (bottom)

4.8 Discovery of Non-person Entities

To emphasize the fact that our methods are not restricted to faces or persons, we used two short videos-one of cars and another of aeroplanes. The cars video consisted of 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 [1]. The objects were detected using the Object-specific detectors [27]. 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. Both videos have 750 frames. The *Cars* video has 694 detections, and the *Aeroplanes* video has 939 detections. The results are shown in Table 8. Once again, TC-CRP does well.

4.9 Semantic Video Summarization

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

4.9.1 Entity-based Summarization

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

Any summary should have two properties: 1) It should be *concise* 2) It should be *representative*. Along these lines, an entity-based summary should have the property that it should cover as many entities as possible, with least number of clusters. On the other hand, the selected clusters should cover a sufficiently large fraction of all the tracklets. In our evaluation of entity discovery (Section 4.7) we have measured *Entity Coverage*, *Tracklet Coverage* and *Number of significant clusters*. These same measures are useful in evaluating the summarization. Entity Coverage and Tracklet Coverage should be high, and number of significant clusters should

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

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

Table 4.9: Conciseness results for different methods for entity-based summarization

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

Table 4.10: Representativeness ($\times 100$) results for different methods for entity-based summarization

be low (See Figures 8,9). To make the evaluation more comprehensive, we define two more measures: 1) *Conciseness:* defined as the ratio of Entity Coverage to the number of significant clusters, and 2) *Representativeness:* defined as the ratio of the Tracklet Coverage to the number of significant clusters.

The results are shown in the Tables 9,10. We find that in terms of Conciseness, TC-CRP turns out to be the best, while the other methods are all comparable when averaged across the videos. In terms of Representativeness, TC-CRP is once again the best by a long way, while TC-CRF and sHDP-HMM are at par. The non-Bayesian methods are way behind.

4.9.2 Shot-based Summarization

Another way of summarization is by a collection of *shots*. [69] follows this approach, and selects a subset of the shots based on the total number of characters (entities), number of prominent characters (entities) etc. A shot [1] is a contiguous sequence of frames that consist of the same set of entities. It is possible to organize the video into temporal segments based on the cluster indices assigned to the tracklets. In a frame f, let $\{Z\}_f$ denote the set of cluster labels assigned to the tracklets that cover frame f. For two successive frames $f1$ and $f2$, if $\{Z\}_{f1} = \{Z\}_{f2}$ we say that they belong to the same temporal segment, i.e. $T(f2) = T(f1)$.

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

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

Table 4.11: Conciseness results for different methods for shot-based summarization

But if $\{Z\}_{f1} \neq \{Z\}_{f2}$, then we start a new temporal segment, i.e. $T(f2) = T(f1) + 1$. By this process, the frames of the video are partitioned into temporal segments. The cluster labels are supposed to correspond to entities, so each temporal segment should correspond to a shot. Each such segment can be easily represented with any one frame, since all the frames in a segment contain the same entities. This provides us a *shot-based summarization* of the video.

As in the case with entities, once again a large number of temporal segments are created by this process, with several adjacent segments corresponding to the same set of entities. This happens because often several clusters are formed for the same entity. Analogous to Entity Coverage, we define *Shot Coverage* as the total number of true shots that have at least one temporal segment lying within it. We then define *significant segments* as those which cover a sufficient number (say 100) of frames. Finally, we define *Frame Coverage* as the fraction of the frames which come under the significant segments.

To evaluate such shot-based summarization, once again we need to consider the two basic properties: conciseness and representativeness. These are measured in exact analogy to the entity-based summarization discussed above (See Figures 10,11). The *Conciseness* of the summary is defined as the ratio of the Shot Coverage to the number of significant segments, while the *Representativeness* of the summary is defined as the ratio of the Frame Coverage to the number of significant segments. The results are shown in Tables 11 and 12. This time we find that in terms of representativeness TC-CRF leads the way, followed by TC-CRP. In terms of conciseness the best performance is given by WBSLRR, which however does poorly in terms of conciseness.

4.10 Implementation Details

Data Collection: The video datasets- the Big Bang Theory episodes and the Mahabharata episodes were downloaded from Youtube. They had been uploaded to Youtube by users. These videos did not have any dialogue scripts, subtitles or any other secondary source of information (except the voices). The episodes to be tested were selected randomly, and not based on any

Figure 4.8: Entity-based summarization of Mahabharata Episode 22 using TC-CRF. Each image is a reshaped cluster mean.

Figure 4.9: Entity-based summarization of Mahabharata Episode 22 using WBSLRR. WBSLRR creates many more clusters than TC-CRF, but both discover the same number of persons (14). Hence the summary by TC-CRF is more concise.

Figure 4.10: Shot-based summarization of Mahabharata Episode 22 using TC-CRF. Each image is a keyframe from a significant segment.

Figure 4.11: Shot-based summarization of Mahabharata Episode 22 using SBMR+ConsClus. SBMR+ConsClus creates more significant segments to cover roughly the same set of true shots as TC-CRF, so TC-CRF summary is more concise

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

Table 4.12: Representativeness ($\times 100$) results for different methods for shot-based summarization

particular criteria. However, it must also be noted that these are longer than videos used earlier for tracklet clustering in [88, 92] etc, and also have more persons. The videos of cars and aeroplanes were obtained from a Youtube-based Objects dataset collected earlier for object detection research.

Features: For the person discovery experiments, we used the face to represent each person. This is a fairly standard practice in related literature, including [4, 92, 88]. The faces are detected frame-by-frame using the openCV face detector, that uses Viola-Jones algorithm [80]. The detection performance is reasonable, though there are some false negatives (misses) and false positives (non-face detections). The number of non-face detections is particularly high on the Mahabharata videos which have more complex backgrounds, and background structures are often mistaken as face. As already discussed, the proposed approach can filter out most of the false detections. It may be possible to reduce the number of missed detections by using a detector that is re-trained on the input video itself rather than pre-trained detectors.

The detections were reshaped into 30×30 images, converted to greyscale and then reshaped to 900-dimensional vectors. This practice is also fairly common in literature related to tracklet clustering [88, 92] and face recognition [59]. Some of the works use RGB colour channels instead of converting to greyscale, in which case we would have a 2700-dimensional vector. We used this in case of the car and aeroplane experiments where colour played a distinguishing role between the entities. However, for faces we observed that this does not give any significant improvement in performance, but increases the computational complexity. In fact, sometimes larger number of clusters were formed. So we used greyscale conversion for the person discovery experiments.

In some recent related works like [75], the full body of the person was used instead of just the face. They even proposed a clothing model for the persons. The advantage of this approach is that, it can handle even those frames where the person's face was not fully visible. In our work too, it should be possible to represent the persons with full bodies, by using a person detector instead of face detector. We can use the part-based detection technique [27] to detect human

bodies, and augment it with the clothing model proposed by [75]. However, we did not explore these because of two reasons: 1) We wanted to keep the approach agnostic to the type of entity, so we did not want features specific to face or human body 2) In the Mahabharata videos, most characters had similar costumes, and we felt that the faces would be more discriminative than the whole body, especially the clothes.

Moreover, instead of using pixel values, it may be possible to use more robust and sophisticated features like Gabor features used in the previous chapter. However, it must be noted that our generative model considers the entity basis vectors to be drawn from a Gaussian distribution $\mathcal{N}(\mu, \Sigma)$, and the individual tracklets/detections are Gaussian perturbations. The representation should be such that these assumptions are reasonable, which is the case in case of pixel values, but less reasonable in case of Gabor features. Other works which have used low-rank matrix representation for face recognition (like [59]) have also stayed away from such features and used simple pixel values for similar reasons.

Implementation: Apart from the face detection which was done using OpenCV, the rest of the system was implemented in Matlab. There were mainly two parts- processing the face detections and converting them into a sequence of tracklet vectors, and the Bayesian inference using these tracklet vectors as input. For the first part, the detections from successive frames with sufficiently close co-ordinates (difference less than 10 pixels in both X and Y directions) were linked to form the tracklets. The threshold distance of 10 pixels was chosen on the basis of empirical judgment, and not a single "impure" tracklet was produced (when detections of different entities are linked). Spatio-temporal distances between successive tracklets were computed as the distance between the last detection of the preceding tracklet and the first detection of the following tracklet. The threshold difference $thres$ between successive tracklets (used in the generative model) was also set by empirical observation. As mentioned, each tracklet was represented as the mean feature vector of its associated detections. Before the inference, all the values were scaled by 255 and thus reduced to the $[0, 1]$ range. This is not necessary in our approach, but some of the baselines (like SBMR) could not work without this step, so for consistency we also used it.

We estimated the base mean μ and covariance matrix Σ from the tracklet vectors. For inference by Gibbs Sampling, we first initialized the latent variables Z by running a forward sampling, using the PPFs of TC-CRP (Eq 4.6) or TC-CRF (Eq 4.8). The mixture components ϕ were estimated along the way. After initialization, the blocked Gibbs Sampling was done using the inference equations already discussed. It was found that after 5-10 iterations the variable values become stabilized for all the test videos, so we stopped the sampling after 25 iterations and collected the final assignments as our inferred values for the variables. So

although Gibbs Sampling is considered slow, in this case only a few iterations were enough to achieve convergence, and the time was not an issue. Among the various competitors, SBMR quickly estimated the low-rank representation, but the constrained clustering step was much slower. The subspace clustering methods took considerable time to even estimate the affinity matrix, following which clustering also took time. WBSLRR also took a lot of time to converge, and its convergence was very sensitive to the initialization. Moreover, it was computationally expensive as it used 6 large matrices.

4.11 Applications, Limitations and Extensions

Applications: The most important application of this system is in analyzing videos uploaded on *public video-sharing sites* like Youtube and Dailymotion. The videos uploaded by users have no detailed information about the contents, and any user may need to watch an entire video to understand if it is relevant to her interest or not. Since there is a humongous amount of video content available on these sites, it is neither possible for users to select the appropriate ones by watching, nor possible for administrators to label them with relevant information in textual/visual form. In such a situation, the only alternative is automatic analysis, as proposed here.

The aim of automatic video analysis in this case would be to produce short but comprehensive summaries of videos, and simplify browsing. Both of these have been addressed in this work. The discovery of entities along with all their occurrences allow the user to watch only those parts of the video that contain an entity she is interested in. On the other hand, the entity-based and shot-based summarizations discussed earlier gives her an idea of the contents of any video without viewing it. Entity discovery and entity-driven summarization are both one-time processes, which can be done as soon as a video is uploaded on to a website. Since it is generally not known beforehand what type of entities are present in a video, the system should have an *ensemble of detectors* for various kinds of entities, and entity-discovery should be carried out separately for all of them.

It may be noted that the proposed method has quite low computational complexity, and we have seen that videos can be processed online (single-pass) without significant performance deterioration. So, it can even be run on *handheld devices and also on surveillance devices like cameras*. These devices nowadays are often equipped with face detectors, which can be used for this task. In surveillance scenarios, it is possible to have applications where the recorded videos would be analyzed on the fly, and the persons who appeared repeatedly would be discovered along with their footages, which would potentially save many hours of inspection in case the authorities are interested in observing the movements of a suspicious person. On the other

hand, if the system is installed on a handheld device such as a mobile phone, it will allow the user to download any video and generate its summary on the fly, without even needing to store the full video.

Limitations: One limitation of the work is that, we are *unable to determine the exact number of entities* as several clusters are formed per entity. This is inevitable due to the fact that the same entity may appear in different poses and views. It may be possible to improve this by using features that are robust to viewpoints, such as SIFT. But once again, for such features the assumption of Gaussian mixture components may not hold good. To find the suitable features, or some other way of regularizing the number of clusters remains an open question.

Extensions: A direction that has not been explored in this work is potential incorporation of supervision. We may consider weak supervision in which a small number of the tracklets will be annotated by an expert. Alternatively, an expert can be shown some pairs of tracklets or clusters and asked if they correspond to the same entity. Such annotation can form a new set of *must-link* constraints, and push the system towards finding the correct number of entities. However, this may be possible to only a limited extent in the application scenarios discussed above, as suitable experts may not be found. One possibility is to ask the uploader herself to make a small number of annotations for the upload to be accepted by the website. The question of which, or which pair of, tracklets or clusters to choose for annotation is a question of *active learning*.

Another extension of this work is to consider temporal entities that are not limited to a single frame but span several frames, such as *actions*. Mining repeated temporal patterns in a sequence is already an area of research. The proposed framework will continue to work if we can find suitable vector/matrix representation of each temporal entity. However, this is likely to be challenging since we do not know the temporal extent of each entity, and different entities may well have different temporal extents. Time warping may have to be used to produce a set of consistent vectors/matrices. A suitable *action detector* is also needed, and this is a research problem by itself which is still relatively new.

Chapter 5

Bayesian Inference for Entity-driven Scene Discovery in Videos

5.1 About this Chapter

In this chapter, we pick up from the previous one and address a related problem: temporal segmentation of videos for entity-driven scene discovery. Once again, we use Bayesian nonparametric modeling. We propose EntScene- a generative model for entities and scenes in videos, along with appropriate inference algorithms. Like the previous chapter, the work done here is also relevant to concise, user-friendly representation of videos in video-sharing websites. On the theoretical side, this is an attempt at segmentation of sequential data that has additional structures.

Publications This work has been published in International Joint Conference on Artificial Intelligence (IJCAI), 2015 to be held in Buenos Aires, Argentina.

1. Adway Mitra, Chiranjib Bhattacharyya, Soma Biswas. *EntScene: Nonparametric Bayesian Temporal Segmentation of Videos aimed at Entity-driven Scene Detection*, International Joint Conference on Artificial Intelligence (IJCAI), 2015

5.2 Temporal Video Segmentation

Naturally occurring sequential data often have an important property- *Temporal Coherence*, i.e. successive datapoints in the sequence are semantically related. It is often important and interesting to *segment* sequential data into semantically coherent subsequences. For example, in a stream of image frames in a video, successive frames in a video show the same objects, except at a few *changepoints*. Detecting such changepoints, i.e. temporally segmenting the

video into coherent subsequences helps in video summarization [62]. Existing approaches to temporal video segmentation are based on similarities of low-level visual features of the frames.

Consider the video of a movie or a TV series episode with a reasonably small but unknown number of persons. A TV serial episode is formed of a few *scenes or acts*- where a small subset of the persons are present. Such a video can be represented by the sequence formed by detecting the faces of the persons in all frames. Each person can be considered an entity. For easier browsing it is interesting to simultaneously discover the persons along with all the frames where they appear, and also segment the sequence into the scenes, and annotate each scene with its persons. This is an entity-driven approach to *scene discovery* of videos.

Temporal segmentation of videos has been studied earlier [62], which segments the video into *shots*. Using the entity-driven approach also it is possible to segment the video into shots, where each shot is associated with an entity. But a video is hierarchically organized [21] and each scene is a collection of several shots. A scene in a TV series episode involves several entities (like people), and successive shots within a scene may alternate between the entities in roughly cyclical patterns. For example during a two-person discourse the camera focuses on one person when she speaks, then on the second person, then back to the first and so on (See Fig 5.1). Existing scene detection methods have been surveyed in [21]. The general approach is to cluster the shots. Much of the existing methods are based on low-level features of frames and shots, rather than entities. A few more recent methods like [69] do attempt character-driven scene discovery in movies, but such methods are heavily dependent on additional information, such as movie scripts. In the absence of such scripts, which is the case for most user-generated videos available online, entity-driven scene discovery is quite unexplored.

In this chapter, we consider *entity-driven temporal segmentation*, where each temporal segment should be associated with one or more entities, like persons, objects or actions. A major challenge is that these entities, or even their number, are not known apriori, and need to be learnt from data. In sequential data each datapoint is associated with one entity. Moreover, the data may have some intricate temporal patterns, like a set of entities may be more frequent in some subsequences than others. modeling these patterns is a second major challenge.

5.3 Problem Definition

Consider the episode of a TV-series, with several entities (say persons). We can run a face detector on each frame, and link spatio-temporally close ones to form tracklets [36]. We consider tracklets spanning r frames. Normally $5 \leq r \leq 20$, and at $r = 1$ we have individual detections. The detections within each tracklet are visually similar due to temporal coherence. It is possible to represent each detection as a feature vector. We represent each tracklet i by the tuple (R_i, Y_i)

Figure 5.1: Keyframes from a TV-series episode. Shot changes occur after frames 2,3,4,6,8, but scene change occurs after frame 6 only

where Y_i is the mean feature vector of the associated detections, and R_i is the set of indices of the frames spanned by i. Note that there can be several face detections per frame, and hence the R-sets of different tracklets can overlap. The tracklets can be ordered sequentially using the indices of their starting frames (ties resolved at random), and for each tracklet i we can define predecessor $pred(i)$ and successor $succ(i)$. If the temporal gap between any tracklet i and $pred(i)$ is too large, we set $pred(i) = -1$ (similarly for $succ(i)$). Let $\{F_j\}_{j=1}^{M}$ be the set of frames with at least one associated tracklet, arranged in ascending order of frame index.

Next, define *latent variables* Z_i as the index of the entity associated with tracklet i and S_j as the index of the scene associated with frame F_j. Temporal coherence property holds at tracklet-level (as in Chapter 4) as well as at frame-level. This means, with high probability

$$Z_{pred(i)} = Z_i = Z_{succ(i)} \tag{5.1}$$

$$\{Z\}_{j-1} = \{Z\}_j = \{Z\}_{j+1} \tag{5.2}$$

$$S_{j-1} = S_j = S_{j+1} \tag{5.3}$$

Here i is a tracklet with neighbors $pred(i)$ and $succ(i)$. Also, $\{Z\}_j = \{Z_i : F(j) \in R_i\}$, i.e. $\{Z\}_j$ is the set of Z-variables corresponding to tracklets covering frame $F(j)$. With slight abuse of notation, $\{Z\}_s$ denotes the set of all Z-variables associated with all frames satisfying $S_j = s$. We call the frames where the Condition 5.2 does not hold as *Level-1 changepoints* and the ones where the Condition 5.3 does not hold as the *Level-2 changepoints*. The *hierarchical segmentation problem* is to find these changepoints. An interval of frames $\{F(j1), \ldots, F(j2)\}$ is a *level-1 segment* if $\{Z\}_{j1} = \{Z\}_{j1+1} = \cdots = \{Z\}_{j2}$, but $\{Z\}_{j1} \neq \{Z\}_{j1-1}$ and $\{Z\}_{j2} \neq \{Z\}_{j2+1}$. In this case, $j1$ and $j2{+}1$ are level-1 changepoints. Similarly, an interval of frames $\{F(j1), \ldots, F(j2)\}$ is a *level-2 segment* if $S_{j1} = \cdots = S_{j2}$, but $S_{j1} \neq S_{j1-1}$ and $S_{j2} \neq S_{j2+1}$. In this case, $j1$ and $j2{+}1$ are level-2 changepoints. $CP1$ and $CP2$ are *Candidate Frames* like shot changepoints which may be Level-1 or Level-2 changepoints respectively, which can be found by shot segmentation

Figure 5.2: Face detections, frames and tracks

methods [1].

The temporal video segmentation setup is illustrated in Figure 5.2. We show 3 successive frames, each of which have two face detections corresponding to two persons. The detections are numbered 1-6 (in green), and linked to each other based on spatio-temporal locality, as shown by the red and blue lines. The frame numbers are indicated in green. Here the tracklets are individual detections, i.e. $R = 1$. So here, $F = \{2000, 2001, 2002\}$; $pred(3) = 1, pred(4) = 2, pred(5) = 3, pred(6) = 4$; $succ(1) = 3, succ(2) = 4, succ(3) = 5, succ(4) = 6$; $Z(1) = Z(3) = Z(5) = 1, Z(2) = Z(4) = Z(6) = 2$; $S(1) = S(2) = S(3) = 1$.

Continuing the line of Bayesian modeling employed in Chapter 4, we model entities as mixture components, and scenes as sparse distributions over these mixture components.

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

5.4 Generative Model and Inference

Having described the notation, we now come to a generative process for videos. One part of this generative process is modeling the observed data (the tracklets). We model the entities as mixture components $\{\phi_k\}$, and the datapoints (tracklets) are drawn from these components.

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

We represent the tracklets as vectors $\{Y_i\}$ of pixel intensity values, as in Chapter 4. Tracklet i is associated with entity Z_i, and according to our model, $Y_i \sim \mathcal{N}(\phi_{Z_i}, \Sigma)$.

5.4.1 EntScene

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

Temporal Coherence at scene level The variables S_j can be Markovian [29] conditioned on its predecessor $S_{fpred(j)}$. Frame j and its associated detections remain in the current scene ($S_{fpred(j)}$) with probability κ, or start a new scene ($S_{fpred(j)} + 1$) with probability $(1 - \kappa)$.

$$S_j \sim \kappa\delta_{S_{j-1}} + (1 - \kappa)\delta_{S_{j-1}+1} \tag{5.4}$$

Modeling of a Scene Each level-2 segment (scene) s has to be modeled as a distribution G_s over mixture components (persons). In case of TV series videos,a person can appear in several scenes. Such sharing of components can be modeled using like Hierarchical Dirichlet Process [76], using H as base distribution (Gaussian) and $\{\alpha_s\}$ as segment-specific concentration parameters.

$$\phi_k \sim H\forall k;\ G \sim GEM(\alpha);\ G_s \sim DP(\alpha_s, G)\forall s \tag{5.5}$$

A sparse modeling can be considered, where each level-2 segment selects a sparse subset of the components using a Beta-Bernoulli process [33][38][85]. Then each segment s has an associated binary vector B_s which indicates which components are active in s.

$$\beta_k \sim Beta(1, \beta)\forall k; B_{sk} \sim Ber(\beta_k)\forall s, k \tag{5.6}$$

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

$$Z_i \sim \rho\delta_{Z_{pred(i)}} + (1 - \rho)(B_s \circ G_s) \tag{5.7}$$

where B_s is the sparse binary vector. As G_s is discrete (Dirichlet Process-distributed), multiple draws from it may result in sampling a component repeatedly in the same segment s. This is desirable in TV series videos, since a particular person is likely to appear repeatedly in a scene.

We can consider an auxiliary variable $C_i \sim Ber(\rho)$ to sample Z_i (similar to TCCRP). Based on all these, the generative process for videos is given as follows:

Algorithm 4 Generative Process for Videos

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

5.4.2 Merge Inference by Blocked Gibbs Sampling

As mentioned earlier, *hierarchical segmentation* is to discover the frames where Equation 5.2 or Equation 5.3 is violated. For this purpose, we need to infer the S_j and Z_i variables. The complete likelihood function in terms of the random variable discussed above can be written as

$$p(Y, Z, S, B, \Phi, \beta, G, G_0) \propto \prod_{k=1} p(\beta_k)p(\phi_k) \times p(G) \times \prod_{j=2}^{M} p(S_j|S_{j-1}) \times \prod_s p(G_s|G) \times \prod_{s,k} p(B_{sk}|\beta_k) \times \prod_{i=1}^{N} p(Z_i|Z_{pred(i)}, S_{F(i)}, B_{S_{F(i)}}, \{G_s\})p(Y_i|Z_i, \Phi) \quad (5.8)$$

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

$$p(Y, Z, S, B) \propto \prod_{j=2}^{M} p(S_j|S_{j-1}) \times \prod_s p(B_s|B_1, \ldots, B_{s-1}) \times \prod_{i=1}^{N} p(Z_i|Z_1, \ldots, Z_{pred(i)}, \{B\}, \{S\})p(Y_i|Z_i, Y_1, \ldots, Y_{i-1}) \quad (5.9)$$

For inference we use Blocked Gibbs Sampling as several variables are usually strongly coupled, and must be sampled together. We form blocks *dynamically* using the S variables. Clearly the scene boundaries occur at frames where the S-variable changes, i.e. where $S_j \neq S_{j-1}$, and each value s of S defines a segment. A block $BL(s)$ is formed as $\{\{B\}_{s-1}, \{B\}_{s+1}, \{B\}_s, \{Z\}_s, \{S\}_s\}$. As a first step, we infer the segment $\{S\}_s$ using Eq 5.4 and the marginal likelihood of the data $\{Y\}_s$. We try to merge each segment s with either segment $(s-1)$ or segment $(s+1)$ (or leave

it alone), so the state-space of $\{S\}_s$ is only $\{s-1, s, s+1\}$. After each iteration, the blocks are re-defined according to the new assignment of $\{S\}$ variables. Since the aim is always to merge each segment with its neighbors, the number of segments should reduce till convergence. We can use $CP1$ and $CP2$ to initialize $\{Z\}$ and $\{S\}$ respectively for Gibbs Sampling, thus getting an initial segmentation. We know that if frames a and b are two successive points in $CP2$, then obviously there is no changepoint between them, i.e. $a < i < i' < b \Rightarrow S_a = S_i = S_{i'}$. This considerably reduces the search space for segments and allows us to keep merging the segments progressively, without considering splits. Once $\{S\}_s$ is sampled, we sample the B and Z variables using Eq 5.9. The process is explained in Algorithm 2.

The various parts of Eq 5.9 can be computed using the inference equations of Indian Buffet Process [33] for $\{B\}_s$ and TC-CRP [54] for $\{Z\}_s$. The convolution of G_s with the sparse binary vector B_s poses a major challenge as it cannot be collapsed by integration, as noted in [85]. We suggest an approximate PPF (modified version of the TC-CRP PPF) for easy inference. In segment s, for a datapoint i where $C_i = 1$, a component ϕ_k may be sampled with $p(B_{sk} = 1, Z_i = k) \propto n_k^s$, which is the number of times ϕ_k has been sampled from $B_s \circ G_s$ within the same segment. If ϕ_k has never been sampled within the segment but has been sampled in other segments, $p(B_{sk} = 1, Z_i = k) \propto \alpha n_k$, where n_k is the number of segments where ϕ_k has been sampled (Corresponding to $p(B_{sk}) = 1$ according to IBP). Finally, a completely new component may be sampled with probability proportional to α_0. Note that $p(B_{sk} = 0, Z_i = k) = 0 \forall k$.

Algorithm 5 Merge Inference Algorithm by Blocked Gibbs Sampling (MI-BGS)

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

5.4.3 Split-Merge Inference

The above algorithm has the property that the number of segments keep decreasing and then converges. This property is desirable as it helps in quick convergence. But two segments can never split after they are merged once, which may come as a disadvantage in case of a wrong merge. The Topic Segmentation Model (TSM) [24] allows for split-merge inference by a Bernoulli random variable U_s with each initial segment s from $CP2$, which indicate whether

or not a new segment should start from s, i.e. if $\{Z\}_s$ and $\{Z\}_{s-1}$ should be modeled with the same distribution. To change U_s from 0 to 1 is to split the segments s and $(s-1)$, and the reverse change is to merge them. The algorithm is explained in Algorithm 3.

Algorithm 6 Split-Merge Inference Algorithm by Blocked Gibbs Sampling (SpMI-BGS)

1: Initialize segments S using CP2; Initialize B,Z;
2: Estimate components $\hat{\phi} \leftarrow E(\phi|B, Z, U, Y)$;
3: **while** Number of segments not converged **do**
4: **for** each segment s **do**
5: Sample $(U_s, \{B\}_s, \{Z\}_s) \propto p(\{B\}_s, \{Z\}_s|\{B\}_{-s}, \{Z\}_{-s}, Y, \{U\}_{-s}, \hat{\phi})$
6: **end for**
7: Update components $\hat{\phi} \leftarrow E(\phi|B, Z, U, Y)$;
8: **end while**

5.4.4 Sweep-Merge Inference

In an alternative inference scheme, we initially consider the sequence to be segmented into the slices defined by $CP2$, and carry out inference of $\{B\}$ and $\{Z\}$ by Gibbs Sampling. In the next step, we make a sweep from left to right, attempting to *merge* the slices. For every slice s, we propose to merge it into the currently running level-2 segment c, using a common binary vector B^{merge} for all datapoints in the proposed merged segment and component assignments $\{Z^{merge}\}$ to the datapoints in slice s. We may accept or reject the merger proposal based on how well $(B^{merge}, \{Z^{merge}\})$ can model the data $Y_{c \cup s}$ in the merged segments (c, s), compared to modeling them as separate segments. The merger probability is enhanced by temporal coherence (Eq 5.4). If we accept it, we will merge slice s into level-2 segment c, and set $S_j = c$ for all frames j occurring within the slice s. If we reject it, we start a new level-2 segment $(c+1)$, and set $S_j = c+1$ for all frames j occurring within the slice s. The process is explained in Algorithm 3.

Algorithm 7 Sweep-Merge Inference Algorithm (SMI)

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

5.5 Experiments on Temporal Segmentation

5.5.1 Datasets and Preprocessing

We use the same dataset of TV-series videos as used in Chapter 4. Once again, we link spatio-temporally close detections to form tracklets, and convert the video to a sequence of tracklets. The hyperparameters like α and β provide some control over the number of components learnt. After tuning them on one episode, we found an optimal setting, where we were able to cover $80 - 85\%$ of the detections with 80-90 components. κ, ρ etc are also fixed by tuning on one episode.

5.5.2 Performance Measures

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

Evaluation of Entity Discovery We use the same evaluation measures for the entities as used in 4. We select only those components that have at least 10 assigned tracklets overall, and reject the rest. This is because we are interested only in persons that have reasonable screen presence. We attribute a selected mixture component to entity A if 70% of the detections

Video	SMI		MI-BGS		SpMI		sHDP-HMM	
	CP	EC	CP	EC	CP	EC	CP	EC
BBTs1e1	**0.84**	**6**	0.78	5	0.80	**6**	**0.84**	5
BBTs1e3	0.91	8	0.94	8	**0.96**	**10**	0.76	6
BBTs1e4	0.89	6	**0.91**	**8**	0.90	**8**	0.83	**8**
Maha22	**0.96**	12	0.89	**14**	0.94	13	0.86	**14**
Maha64	**0.94**	**14**	0.92	13	0.91	12	0.91	**14**
Maha65	0.88	16	0.83	15	**0.90**	**18**	**0.90**	17
Maha66	0.91	14	0.81	**15**	0.89	**15**	**0.95**	13
Maha81	**0.86**	**22**	**0.86**	21	0.85	20	0.84	20
Maha82	**0.93**	19	0.89	19	0.81	**20**	0.86	**20**

Table 5.1: Entity Discovery results for SMI, MI-BGS, SpMI and sHDP-HMM

assigned to that component belong to entity A. This is because, we observe that if a component's associated tracklets are at least 70% pure then the corresponding mean vector μ_k resembles the entity well enough for identification. For large components (200 or more associated tracklets), we observe that 60% purity is enough for identifiability. We measure as *Cluster Purity (CP)*, the fraction of the selected components which can be assigned to an entity We also measure as *Entity Coverage (EC)*, what fraction of the entities with at least 10 tracklets, have been represented by at least one selected component. We compare these with our *baseline: sticky HDP-HMM [29].* This is the state-of-the-art BNP model suited to unsupervised learning of mixture components and segmentation.

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

We can look upon segmentation as a *retrieval problem*, and define the *Precision and Recall of level-2 changepoints (CP-RC2 and CP-PR2).* Let i be the starting point of an inferred segment s, i.e. $S_{i-1} \neq S_i$. Then, if there exists $(i0, s0)$ such that $i0$ is the starting point of a gold-standard segment $s0$ satisfying $|i - i0| < k$ then inferred changepoint (i, s) is *aligned* to gold standard changepoint $(i0, s0)$. Precision, recall of a segmentation are defined as

$$\textbf{Precision} = \frac{\textbf{\#inferred segments aligned to a gold-standard segment}}{\textbf{\#inferred segments}}$$

$$\textbf{Recall} = \frac{\textbf{\#gold-standard segments aligned to an inferred segment}}{\textbf{\#gold-standard segments}}$$

Video	SMI		MI-BGS		SpMI	
	CP-RC2	CP-PR2	CP-RC2	CP-PR2	CP-RC2	CP-PR2
BBTs1e1	**0.78**	0.26	0.78	**0.30**	0.33	0.22
BBTs1e3	0.77	0.24	**0.85**	0.23	0.85	**0.30**
BBTs1e4	0.75	**0.32**	**0.83**	0.26	0.75	0.24
Maha22	0.71	0.21	**0.76**	**0.24**	0.53	0.20
Maha64	**0.88**	0.16	0.82	0.17	0.71	**0.23**
Maha65	0.78	0.20	**0.87**	**0.27**	0.74	0.23
Maha66	0.80	0.13	**0.87**	0.16	0.47	**0.19**
Maha81	0.55	0.19	**0.80**	**0.22**	0.75	0.16
Maha82	0.32	0.15	**0.72**	**0.38**	0.48	0.26

Table 5.2: Recall and Precision of segment boundaries, using alignment threshold to be 20% of the average scene length

Video	SMI			MI-BGS			SpMI		
	S2	NS2	SP2	S2	NS2	SP2	S2	NS2	SP2
BBTs1e1	0.14	51	**0.77**	**0.09**	44	0.67	0.19	**25**	0.61
BBTs1e3	0.10	40	0.74	**0.08**	46	**0.88**	0.10	**30**	0.68
BBTs1e4	0.11	26	0.71	0.12	37	0.79	0.13	35	0.81
Maha22	0.16		0.82	**0.12**	**53**	**0.84**	0.15	73	0.74
Maha64	0.19	94	**0.91**	0.19	81	0.89	**0.18**	**50**	0.77
Maha65	0.18	87	**0.82**	**0.16**	**71**	0.71	0.19	72	**0.82**
Maha66	**0.12**	87	0.82	0.20	79	**0.90**	0.19	**35**	0.78
Maha81	0.23	**56**	**0.88**	**0.15**	68	0.78	0.20	89	0.82
Maha82	0.15	50	**0.77**	**0.07**	**46**	0.71	0.19	69	0.68

Table 5.3: Segmentation error (S2), number of segments formed (NS2) and segment purity (SP2)

5.5.3 Results

The entity discovery results are shown in Table 5.1, and the segmentation results in Tables 5.2,5.3 and 5.4. We see that in terms of entity discovery, none of the methods (including sHDP-HMM) have any significant advantage. Averaged across all the videos, SMI leads in terms of Cluster Purity, while sHDP-HMM is the worst. In terms of Entity Coverage, all methods are almost at par when averaged across the videos. At level-2 (i.e. scenes), we see that MI-BGS clearly performs better than SMI and SpMI on precision and recall of segment boundaries (CP-PR2, CP-RC2) and also fares best on the segmentation error (S2). However, SMI is found to be better in terms of segment purity (SP2), which is understandable since it produces a large number (NS2) of pure but small segments. On the other hand, SpMI is found to produce a small number of segments, but they are often inaccurate, resulting in its poor performance in terms of all the measures. In general the number of segments formed (NS2) is quite high compared to the actual number of scenes, and this affects the precision values for all the methods. This is not a fault of the segmentation algorithms, but the result of formation of a large number of clusters compared to the number of persons. This happens due to significant variations in face poses of the same person across a video.

Here, we give examples of three temporal segments from Maha65 learnt by SpMI, to illustrate our evaluation measures for segmentation. Each segment is visually represented by the set of

Video	SMI		MI-BGS		SpMI	
	CP-RC2	CP-PR2	CP-RC2	CP-PR2	CP-RC2	CP-PR2
BBTs1e1	0.61	0.21	**0.72**	**0.28**	0.22	0.15
BBTs1e3	0.23	0.07	**0.77**	**0.21**	0.54	0.19
BBTs1e4	0.58	**0.25**	**0.67**	0.21	0.50	0.16
Maha22	0.29	0.09	**0.65**	**0.20**	0.24	0.09
Maha64	0.59	0.10	**0.76**	**0.16**	0.41	0.13
Maha65	0.35	0.09	**0.57**	**0.18**	0.48	0.15
Maha66	0.47	0.08	**0.53**	0.10	0.27	**0.11**
Maha81	0.35	0.12	**0.55**	**0.15**	0.50	0.11
Maha82	0.24	0.12	**0.28**	**0.15**	0.16	0.09

Table 5.4: Recall and Precision of segment boundaries, using alignment threshold to be 200 frames (about 8 seconds)

Figure 5.3: A learnt temporal segment which is *pure* as it consists of frames from a single scenes. Note that it also covers several shots, and the two persons appear alternately

the first frames of all tracklets covered by it.

5.6 Co-modeling of Videos

The generative process can be extended to multiple videos, which share the same persons (with similar facial appearances). This may be done by allowing the videos to share the mixture components $\{\phi_k\}$, though the weights may differ. In that case, the inference process (SMI, SpMI or MI-BGS) can proceed on the individual videos by estimating the shared components first. This can be done by considering all the initial segments induced by $CP2$ from all the sequences together, and estimating the shared components while initializing $\{B\}$ and $\{Z\}$ variables accordingly.

We experimentally evaluate such co-modeling on videos which have almost the same set of persons with similar facial appearances, and hence may be modeled using the same mixture components. Modeling with same set of mixture components allow us to easily find out common persons and temporal segments from the videos. In case they are modeled separately, discovery of common persons and segments require matching the sets of mixture components from the different videos, which may not be accurate.

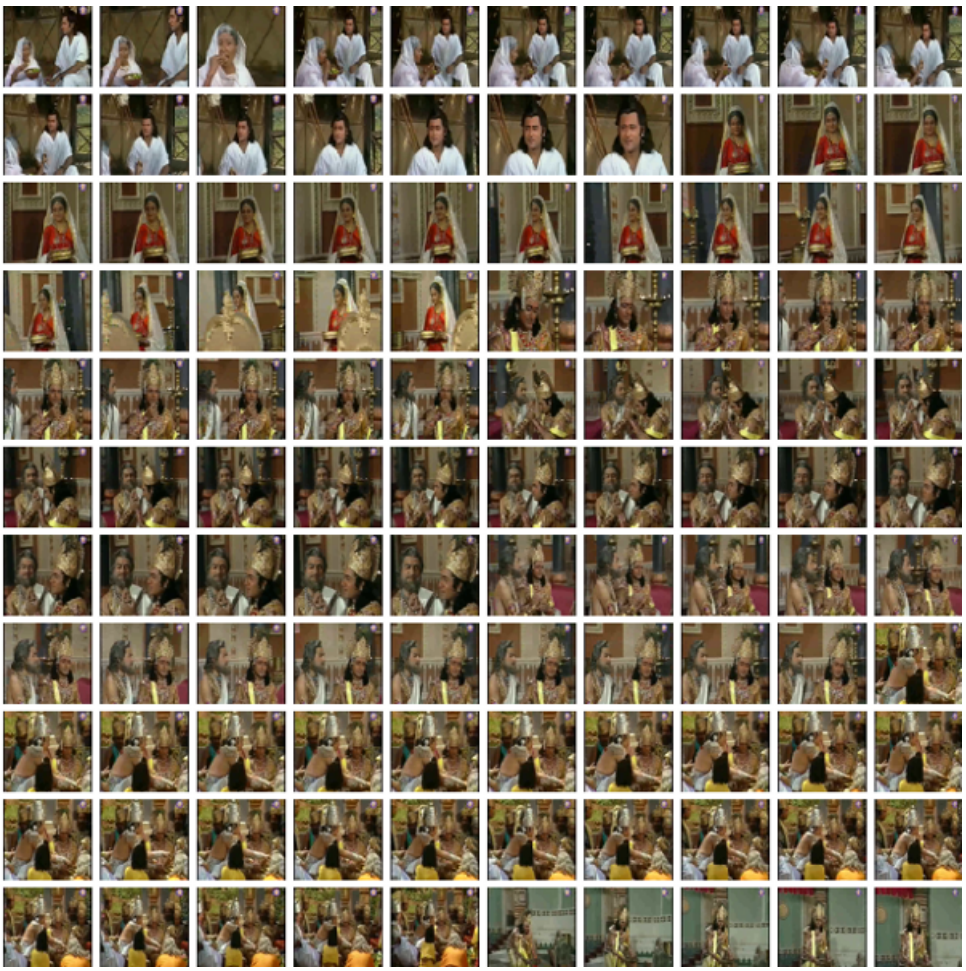


Figure 5.4: A learnt temporal segment which is *impure* as it consists of frames from several scenes

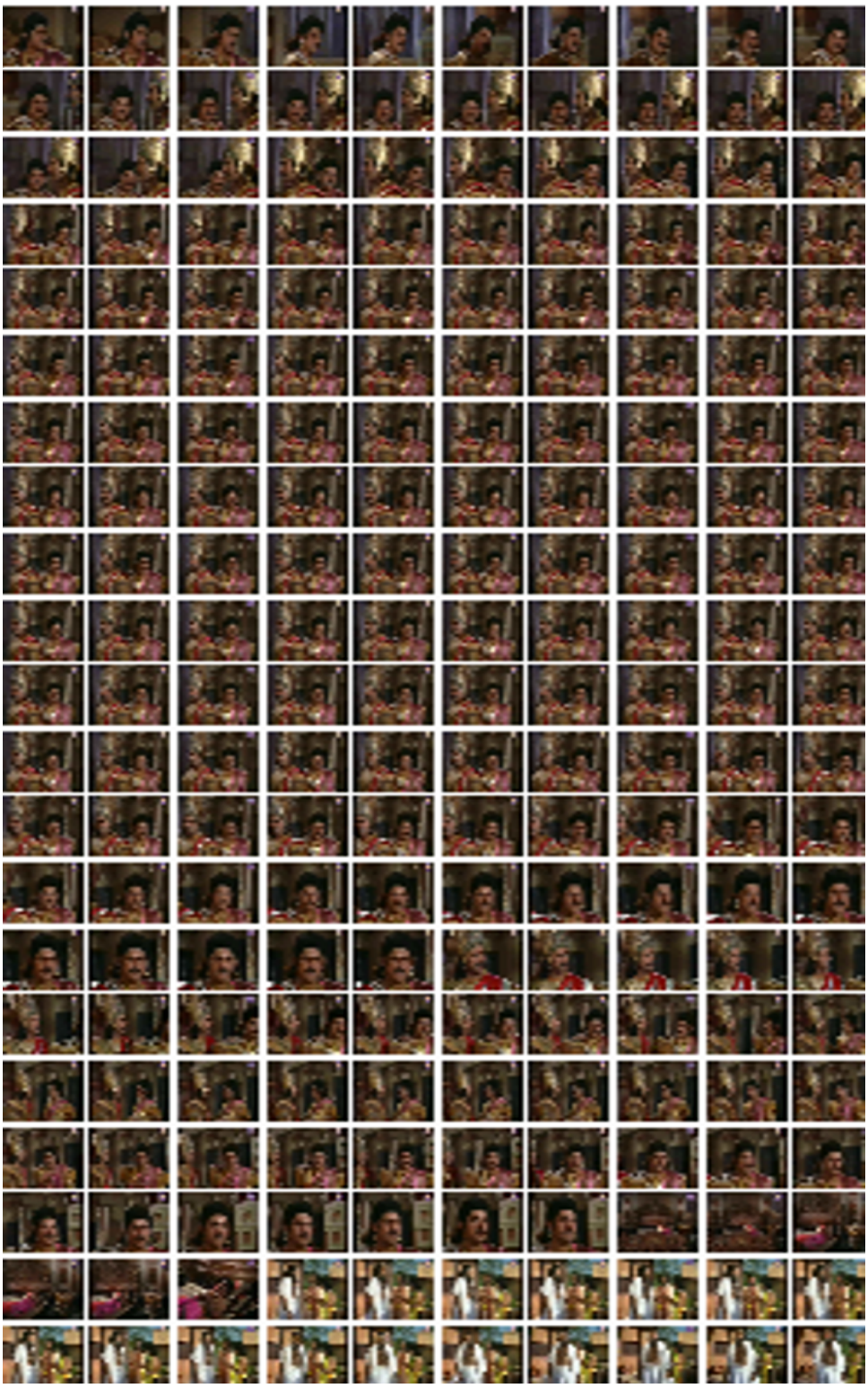


Figure 5.5: A learnt temporal segment which is *impure* as it consists of frames from several scenes, but it misses the true scene changepoint by a relatively short margin, which may be within tolerable limits.

Method	BBTep1	BBTep3	BBTep4
Co-Modeling	**0.72**	0.76	**0.67**
Individual	0.58	**0.77**	0.64

Table 5.5: Segment Matching precision for Co-Modeling and separate modeling of video pairs

For this we collect a set of videos corresponding to 3 episodes of the TV series *The Big Bang Theory*. For each episode, we have a main video (full episode) and a short video showing snippets. Every such pair of videos contain the same persons in same facial appearances, and hence fits our case. We first create initial segmentations of all the videos using their respective shot boundaries ($CP2$). Next, for each pair of videos from same episodes we learn the mixture components together, and use these common components to *identify similar segments (that contain the same persons) across the pairs*. The binary vector B_s learnt for every segment s is used for this purpose. We say that a segment s_i^a from video a and another segment s_j^b from video b are similar based on the Hamming distance of the corresponding B-vectors. Every pair of matched segments can then be classified as *good* or *bad* according to a gold-standard matching of such segments, and the *Matching Precision* (fraction of matches that are *good*) can be measured. As baseline, we repeat this for the case where the two videos in a pair are modeled individually, and then the two sets of learnt mixture components are matched based on ℓ_2-distance of their mean vectors. The results are shown in Table 5.5, which show that co-modeling performs clearly better than individual modeling in this case. It is also possible to evaluate the two approaches (co-modeling and individual modeling) based on the measures CP,PC, and P_k. However, we find that according to these measures, neither approach is clearly better than the other.

5.7 Applications, Limitations and Extensions

Applications: Like the previous work on entity discovery, the main application of this work is in analyzing user-uploaded videos on public video-sharing sites. Once again the main aim is to simplify browsing, and provide the users with a semantic summary. In this case, the semantic summary can be in the form of representative frame(s) from each scene, which will be more concise than the shot-based summarization discussed in the previous chapter (as there are many more shots than scenes). On the other hand, browsing will be simplified since the user will be able to watch only those scenes that she wants rather than the full video.

The co-modeling part can also have interesting and novel applications. This can be, in fact, looked upon as a reverse problem of summarization- given any short summarized video we can try to find its full version in a video repository. So if an user finds a short video clip but is not sure of its source, this method can help her compare it with others, and thus identify it by

finding a match.

Limitations: One important limitation of this work is that the number of temporal segments produced is typically much larger than the number of scenes. As already discussed, there are two main reasons for this 1) Several clusters are formed per entity, so that an entity seen earlier in a scene may be considered a new entity when it re-appears 2) a scene has complex temporal structure, as different entities may appear alternately, and it is very difficult to determine when a new scene starts. It will be very useful if such dynamics can be modeled within the generative framework.

Extensions: This work can be related to the novel tasks of co-summarization and co-segmentation. When several videos of a particular event, say a social ceremony or a sports event are shot and uploaded on video-sharing sites, it is of interest to relate and interlink them so that users can seamlessly switch from one video to another if they want. Since these videos have nearly the same set of entities, they can be co-modeled as discussed above, and this in turn can simplify co-summarization and co-segmentation. Besides, the inference algorithms proposed here are independent of computer vision, and can handle any kind of sequential data, like speech or text. *Discourse analysis* is a text mining task in which these algorithms may find use.

Chapter 6

Modeling Temporal Coherence in Low-rank Matrices for Video Representation

6.1 About this Chapter

In this chapter we discuss a collection of small results related to low-rank matrix representation of videos. Motivated by video representation using low-rank matrices that has been employed frequently over the last few years, we explore how temporal coherence can be incorporated in such representations. We show that the existing low-rank matrix recovery methods cannot capture temporal coherence. We explore models for matrices having a particular structure: namely sets of similar columns. For matrices that are used to represent videos, we show that a Bayesian model along the lines of TC-CRP is able to achieve good results in recovery of such matrices.

6.2 Introduction

In an earlier chapter we have dealt with entities which appear repeatedly in a video, and used discrete predictive function like TC-CRP to model the repetitions. In this chapter, we consider matrices which have similar or near-identical columns, along with temporal coherence. We consider matrices with sets of identical columns, including cases where adjacent columns are likely to be equal.

We note that matrices with sets of identical columns are low-rank, and so matrices with sets of similar columns can be well approximated with low-rank matrices that have sets of identical columns. This connects us to the vast amount of work done about low-rank matrix

recovery, in presence of missing entries or sparse entry-wise corruptions. However, we show that these algorithms perform quite poorly on the matrices we are interested in. The approximating matrices they recover may have low rank, but they do not contain the additional structure which we want (e.g. sets of identical columns). Taking a convex optimization approach with suitable regularizers that encourage such structures also does not greatly help, as they do not strictly enforce the properties. The better option is generative processes which explicitly enforce the structures we are interested in. Even computationally these are more suitable, as the convex optimization approaches require computing Singular Value Decompositions (SVD) of large potentially matrices, which is very expensive.

6.3 Matrices with sets of Identical Columns

Matrices are quite commonly used in computer vision. They have been used for both still images [59] and for videos [14]. 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 both cases, the matrix is expected to have sets of nearly identical columns (such as those representing the face of the same person). In case of videos too, feature-level Temporal Coherence ensures that successive columns are nearly identical except at the shot/track boundaries. The small differences which exist between the columns in such sets is due to noise, camera movements or movements in the scene, and it should be possible to approximate this matrix by one that has sets of identical columns. Such a matrix will clearly have low rank. This representation can help in applications like scene segmentation or clustering, or efficient selection of exemplars. To find this low-rank approximation, we can consider **low-rank matrix recovery** methods.

6.3.1 Low-rank Matrix Recovery

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 ϕ_k,

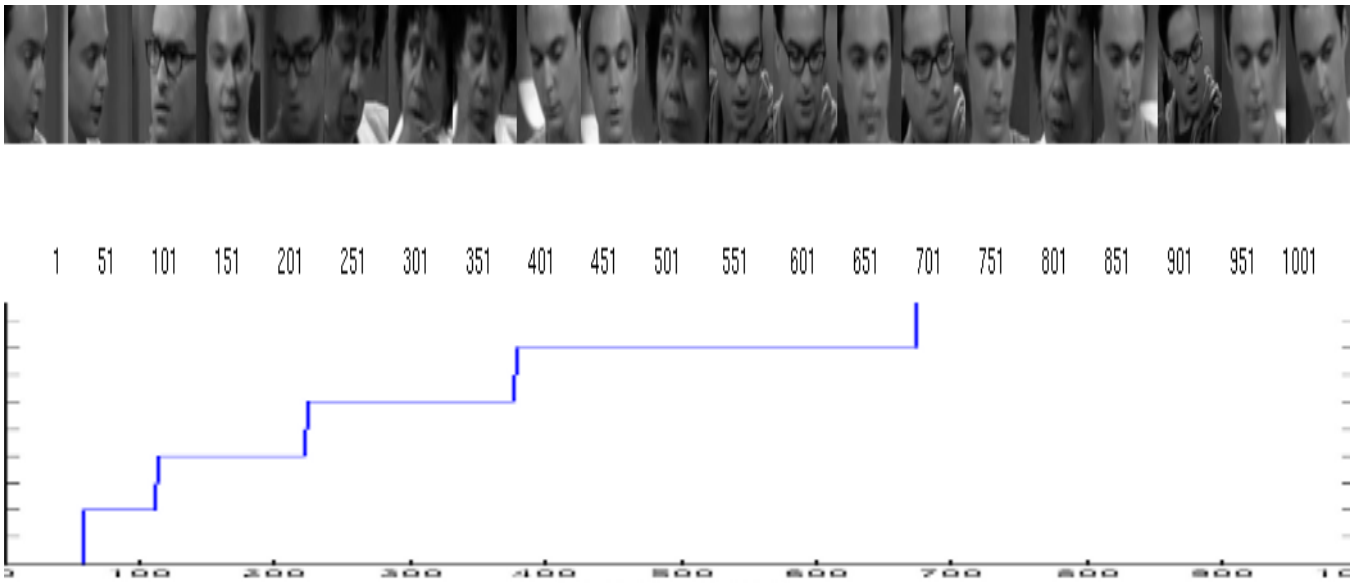


Figure 6.1: Face detections from the test video and the expected rank-column plot of its low-rank matrix representation: a step function which may increase at shot change-points. In case of the low-rank matrices learnt by RPCA, BRPCA and SBMR, this behavior is not observed.

then for column X_{i+1} we sample ϕ_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 (200×1000), 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 6.1. 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 $\tilde{X}_i = rank(X_{1:i})$- the rank of the submatrix formed by the first i columns of X. If X has identical columns between the change-points, this

quantity should remain fixed between these changepoints, and may increase by 1 only at the changepoints. Hence the plot of $\tilde{X}_i$ versus i should be a *step-function*, as shown in Figure 6.1. We call this plot as the *Rank-column plot* of X. We study the *rank-column plot*(Figure 6.2,6.3) of the estimated low-rank matrix X returned by three recent methods for low-rank matrix approximation: Robust PCA [14], Bayesian Robust PCA [23] and Sparse Bayesian Robust PCA [5]. 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-changepoints. Moreover, *if the estimated matrix had rank r, then the submatrix formed by any set of m 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.

6.3.2 Convex Regularizers to encourage Identical Columns

Most of the existing approaches for low-rank matrix recovery use convex optimization. The general form of the optimization problem for matrix completion is

$$\min_X ||X||_* + \gamma||Y - X||_\Omega^2 \tag{6.1}$$

where Ω is the set of observed entries. For matrix extraction, the formulation is

$$\min_X ||X||_* + \gamma||Y - X||_1 \tag{6.2}$$

Here Y is the observed, potentially noisy matrix and X is the low-rank one, which we are trying to recover. Both of them are $M \times N$, where N is the number of datapoints and M is the dimensionality. The nuclear norm $||||_*$ tries to minimize the rank, but as already observed, it does not encourage the columns to be identical. To this end, we must add new regularizers. Define $D_{ij} = ||X_i - X_j||_2$- a measure of difference between any two columns (i, j). D is then a $N \times N$ matrix. D is sparse if and only if a large number of columns of X are identical, and we know that D can be encouraged to be sparse by minimizing its ℓ_1 norm, i.e. the absolute sum of its entries. So $||D||_1 = \sum_{i=1,j=1}^{N,N} ||X_i - X_j||_2$ should be added as a regularizer. The matrix recovery problem is then

$$\min_X ||X||_* + \gamma\mathcal{L}(Y - X) + \alpha \sum_{i=1,j=1}^{N,N} ||X_i - X_j||_2 \tag{6.3}$$

where $\mathcal{L}$ is a suitable regularizer depending on the task- completion or extraction. There is no closed-form solution to this problem, and the gradients also cannot be computed, so we have to solve it using ADMM, as done in [15].

Note that the above approach tries to encourage pairs of columns to be identical, but encodes no further information. But in case of temporally coherent data (like videos), successive columns are more likely to be identical than those far apart. So an alternative to the above regularizer is *Fused Lasso*[77] which encourages adjacent columns to be identical, rather than every pair of columns. Then the matrix recovery problem becomes:

$$\min_X ||X||_* + \gamma\mathcal{L}(Y - X) + \alpha \sum_{i=2}^{N} ||X_i - X_{i-1}||_2 \tag{6.4}$$

This can be made further sophisticated, by finding a common ground between the two formulations: every column should be encouraged to be identical to others within a neighborhood, of size say R. In that case, the formulation is:

$$\min_X ||X||_* + \gamma\mathcal{L}(Y - X) + \alpha \sum_{i=R+1,r=-R}^{N-R,R} ||X_i - X_{i-r}||_2 \tag{6.5}$$

Unfortunately it turns out that these formulations do not work particularly well even on synthetic data. We do get a low-rank X, but usually not a single pair of columns are *exactly* identical. The problem increases with increasing dimensionality M. We observed that in case of many pairs of columns, a large number of elements are identical but a few different ones exist. This means that we do not automatically get the clustering that we want. This happens because the regularizers we used are only convex approximations of the actual regularizers (like ℓ_0 norm) which are non-convex. The convex approximations can only *encourage* sparsity, but not *enforce*. This was observed earlier by [56], who noted that these methods promote *weak sparisty* where entries are minimized, as opposed to *strong sparsity* where entries are forced to be 0.

6.3.3 Bayesian model to enforce Identical Columns

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 earlier in this thesis. 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 κ to 1 (i.e. $C_i = 1 \forall i$). Note that at any column i the rank $\tilde{X}_i$ increases from $\tilde{X}_{i-1}$ if a new vector

(different from $X_1, \ldots, X_{i-1}$) is sampled. The value α in the PPF (Equation 4.6) regulates the probability of sampling of a new vector from the base distribution, so a *smaller value of* α *ensures a lower rank*.

Matrix Completion: In case the observed column vectors have *missing entries*, the inference algorithm for TC-CRP can be easily modified so that these entries are inferred. 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 $Y_{kd} = \sum_{i:Z_i=k,(i,d)\in\Omega} Y_{id}$.

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) $\frac{||X_{recovered} - X_{original}||_F}{||X_{original}||_F}$, the *rank error*(RE) $\frac{|rank(X_{recovered} - X_{original})|}{|rank(X_{original})|}$. 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 [72], which requires a *similarity matrix* between pairs of datapoints. We define pairwise similarity $S(i,j) = exp(-||X_i - X_j||_{\Omega_i \cap \Omega_j})$ (Ω_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 6.2 for synthetic data and Figure 6.3 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 changepoints.

6.4 Applications in Computer Vision

Computer Vision is a domain where this kind of matrices have lots of applications. In the above experiment, we have already discussed representing face detections from successive video frames as columns in a matrix. The tasks in this case can be person discovery from videos, as already explored. In fact, we have compared the proposed method (based on TC-CRP) against some constrained clustering methods, where the number of clusters was determined using a low-rank

missing	TCCRP			NCUT	SVT		OPTSPACE		SBMR	
fraction	FE	RE	RAND	RAND	FE	RE	FE	RE	FE	RE
0.1	**0.002**	**0**	**1.0000**	0.998	**0.031**	0.055	0.138	0.98	0.038	0.24
0.3	**0.008**	**0**	**1.0000**	0.988	0.040	0.03	0.138	0.98	0.049	0.66
0.5	**0.040**	**0.02**	**0.9994**	0.985	0.048	0.09	0.137	0.98	0.068	0.71
0.7	**0.059**	**0.02**	**0.9990**	0.980	0.116	0.40	0.136	0.98	0.103	0.70

Table 6.1: 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

missing	TCCRP			NCUT	SVT		OPTSPACE		SBMR	
fraction	FE	RE	RAND	RAND	FE	RE	FE	RE	FE	RE
0.1	0.008	**0**	**1.0000**	0.998	0.03	0.14	0.169	0.97	**0.007**	0.15
0.3	**0.013**	**0.01**	**1.0000**	0.994	0.03	0.14	0.169	0.98	0.037	0.60
0.5	**0.035**	**0.04**	**0.9999**	0.987	0.038	0.09	0.178	0.98	0.056	0.71
0.7	**0.048**	**0.05**	**0.9996**	0.976	0.105	0.41	0.178	0.98	0.095	0.83

Table 6.2: 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

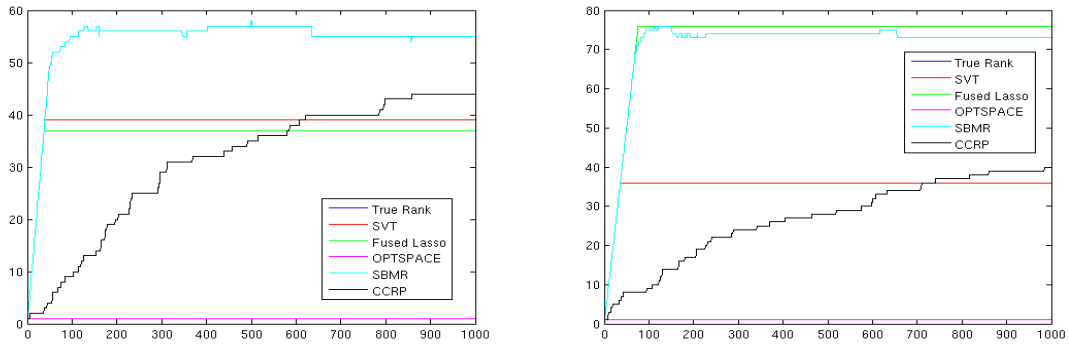


Figure 6.2: Rank-column plots for various methods. Left figure is for a matrix with 10% missing entries, and right figure for 50% missing entries. The Blue Line (True Plot) and the Black Line (proposed method) coincide

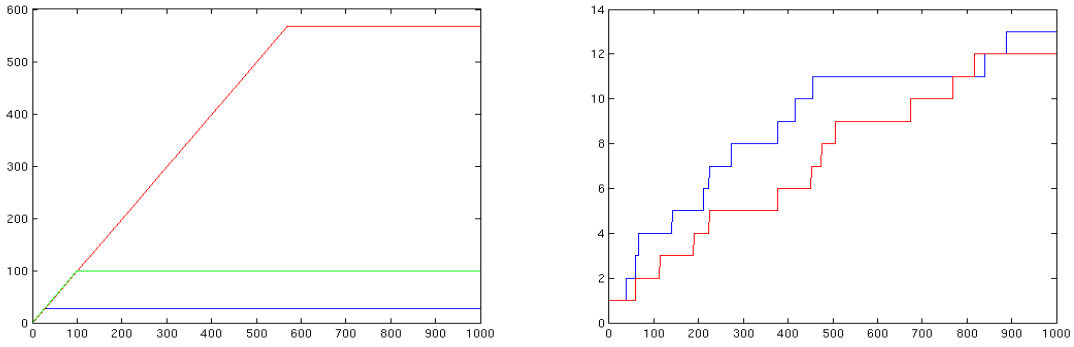


Figure 6.3: Left: Rank-column plots for SBMR(blue), RPCA(red) and BRPCA(green) for the test video. The estimated matrices all have rank much more than the number of shot segments (12), and do not exhibit the expected step function-like behavior. Right: Rank-column plot for TC-CRP(blue), and the shot number(red) which increments at the shot changepoints. The rank is 13, and the steps reasonably match with the shot changepoints

matrix representation. In case we have a low-rank matrix recovery method which can create a matrix with sets of identical columns, it will have already done the clustering.

An additional advantage we get is the ability to deal with *missing pixels* by matrix completion. 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 tested the performance of our method with 20% pixels missing at random. We carried out experiments on the person discovery problem, using the same videos as used earlier, to show that the performance is relatively unaffected by the presence of missing pixels. As benchmark, we consider *Low-rank Matrix Completion* methods like SBMR [5] and OPTSPACE [42]. 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 earlier.

Chapter 7

Bayesian modeling of Temporal Coherence in Hierarchically Grouped Sequential Data

7.1 About this Chapter

In this chapter, we focus our attention on modeling temporal coherence in text documents. We look upon documents as hierarchically grouped sequential data, and consider multi-level clustering and segmentation of such data by modeling temporal coherence. We explore and qualitatively compare various existing Bayesian models which have similar goals. We also propose a nomenclature for classifying such models, and also a Generalized Bayesian model that can subsume all the existing ones. Next, we consider two novel instantiations of this model. These are an attempt to model temporal coherence at multiple levels- one model uses a Markovian approach, and the other uses a Semi-Markovian approach. We have considered a novel application- simultaneous multi-level segmentation of news transcripts into broad news categories and individual stories.

Publications: One of the proposed models and its associated experiments have been published in European Conference for Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD), 2013 held in Prague, Czech Republic. The remaining part of the work is under review.

1. Adway Mitra, Ranganath B.N., Indrajit Bhattacharya. *A Layered Dirichlet Process for Hierarchical Segmentation of Sequential Grouped Data*, ECML-PKDD 2013

2. Adway Mitra. *Exploring Bayesian Models for Multi-level Clustering of Hierarchically*

Grouped Sequential Data, CoRR abs/1504.04850 (2015)

7.2 Introduction

In many applications we come across hierarchically grouped data. For example in a text corpus, data is grouped into documents, paragraphs and sentences. Such data can be clustered at multiple levels, based on the notion of topics. A large number of hierarchical Bayesian models have been proposed for such data, many of whom are quite similar to each other in various aspects. However, to the best of our knowledge, there has not been much research aimed at placing these models in perspective, and making a comparative study of them, except empirical comparisons. This is what we attempt in this chapter. The main aspect of these models which we compare is how they share the mixture components and distributions across the groups at different levels.

The contributions of this chapter are as follows: 1) We introduce a novel classification of Hierarchical Bayesian models for grouped data, based on Degree of Sharing of mixture components and distributions 2) We introduce a generalized Hierarchical Bayesian model and show many existing ones to be special cases of it, and 3) We show how it can be adapted for news transcript segmentation, for which we give an inference algorithm and demonstrate experimental results.

7.3 Notations

Consider N datapoints $Y_1, Y_2, \ldots, Y_N$, of any type (eg. integers, real-valued vectors) based on the application. Each of these are associated with *group membership variables* (positive integers), which specify the grouping of the datapoints. If there are L levels of grouping, each datapoint Y_i is associated with observed variables $\{D_i^1, D_i^2, \ldots, D_i^L\}$. For example, a text corpus consists of a set of documents, each of which consists of word-tokens. We can consider the word-tokens as data-points $\{Y_i\}$, which are tagged with their document memberships using $\{D^2\}$, where $\{D^1\}$ are the token indices, to capture the sequential ordering. This is the standard setting used in most topic models for text documents. In addition, it is possible to consider a 3-level grouping with *sentences* within documents. Then each word-token Y_i is associated with a sentence membership variable D_i^2 and a document membership variable D_i^3. In this chapter, we will overload D^l $(l > 1)$ to indicate the higher-level group-memberships of lower-level-groups. For example if g is the index of a level-2 group, then $D^3(g)$ is the level-3 group that covers all the datapoints under group g, i.e. $D^3(g) = D_i^3$ where $D_i^2 = g$. Please see Fig 7.1 for illustration.

Most topic models consider documents or sentences to be bags of words, and do not consider

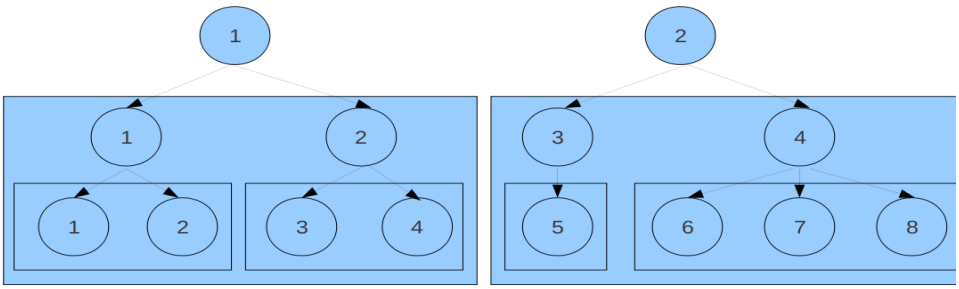


Figure 7.1: Data grouped at 3 levels: $D_i^1 = i \forall i$, $D_1^2 = 1$,$D_3^2 = 2$,$D_5^2 = 3$,$D_6^2 = 4$ etc, $D_i^3 = 1$ for $i = 1 \ldots 4$, $D_i^3 = 2$ for $i = 5 \ldots 8$. Also, $D^3(1) = 1$, $D^3(3) = 2$ etc

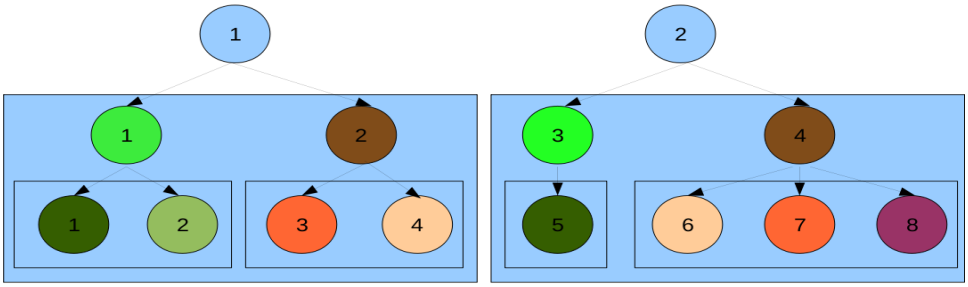


Figure 7.2: Grouped data clustered at 2 levels ($l = 1, 2$). Colours indicate the clustering, like $Z_1^1 = Z_5^1$, $Z^2(2) = Z^2(4)$ etc. Different colours used at the two levels. Note that $Z_3^1 \neq Z_6^1$, but $Z_3^2 = Z_6^2$

the sequential nature of the data. This can be avoided with the current representation, as sequential relations between the word-tokens can be encoded using the indices $\{D^1\}$ which takes integer values. Accordingly for each datapoint i we can define sequential neighbors $prev(i)$ and $next(i)$. Even sequential ordering of the higher-level groups like sentences and documents can be captured by the variables D^2 and D^3 respectively. In case sequential ordering is irrelevant at any level (for example, ordering of documents is usually not relevant unless there are timestamps), the group membership variables at that level act as simple identifiers.

The groups at the different levels may be clustered in some applications, like multi-level clustering. For this, we associate a *cluster variable* with each datapoint: $\{Z^1\}, \{Z^2\}, \ldots, \{Z^L\}$. Again, we can overload Z^l $(l > 1)$ to indicate the higher-level cluster memberships of lower-level groups. If g is the index of a level-2 group, then $Z^3(g)$ is the level-3 cluster that covers all the datapoints under g, i.e. $Z^3(g) = Z_i^3$ where $D^2(i) = g$. This causes hierarchical clustering of the datapoints, specified by the tuple $\{Z_i^1, \ldots, Z_i^L\}$.

A Bayesian modeling involves mixture components and mixture distributions. We will consider K mixture components (topics) $\phi_1, \ldots, \phi_K$, where K may not be known. Also, we need mixture distributions for each level- $\{\theta^1\}, \{\theta^2\}, \ldots, \{\theta^L\}$. These are discrete distributions

over index variables, that are cluster indices of the lower layer. Note that the cluster indices at level 1 are indices of the mixture components. At each level, the distributions may be specific to the group clusters, defined by the group cluster variables Z. For example, if the groups at level l are clustered, the groups in cluster k i.e. $\{j : Z^l(j) = k\}$ will have access to only the distribution θ_k^{l-1} at level $l-1$. The basic inference problem is to learn the cluster assignments $\{Z^l\}$, and estimate the mixture components ϕ.

7.4 Review of Existing Models

In this section we make a short review of several well-known models using the above notation. The models can be classified based on the number of levels of grouping in the data that they consider.

7.4.1 1-level models

The simplest models are the 1-level mixture models, like Gaussian Mixture Model [7]. Here $L = 1$, with $D_i^1 = i$ and the datapoints are not grouped at all. There are K mixture components $\{\phi\}$ which are Gaussian distributions, i.e. $\phi_k = \mathcal{N}(\mu_k, \Sigma_k)$. In general, the mixture components need not be Gaussian. The mixture distribution θ^1 is a K-dimensional multinomial. Each datapoint is assigned to a mixture component Z_i^1, which defines a clustering of the datapoints. This assignment is IID as $Z_i^1 \sim \theta^1$ and sequential structure of the datapoints is not considered.

In GMM, the number of mixture components K is fixed and known. A non-parametric model with $L = 1$ is the Dirichlet Process Mixture Model (DP-MM), which considers infinitely many mixture components, though only a few of them are used for a finite number of datapoints. The mixture distribution θ^1 is an infinite-dimensional multinomial, drawn from a stick-breaking distribution. The parameters of the mixture components are drawn from *base distribution* H.

A one-level nonparametric model which does consider the *sequential structure of the data* is the HDP-HMM [29]. This model considers a set of θ^1-distributions from which one may be chosen conditioned on the previous assignments of Z^1. The Z^1-assignment to each datapoint i is done as $Z_i^1 \sim \theta_j^1$ with $j = Z_{prev(i)}^1$, where $prev(i)$ is the predecessor of the current datapoint in the sequential order encoded by $\{D^1\}$, i.e. $prev(i) = i'$ where $D_i^1 = D_{i'}^1 + 1$.

7.4.2 2-level models

Next, we move into two-level models, i.e. where $L = 2$. This is the standard setting for document modeling, where the word-tokens are grouped into documents (one level of grouping). The document membership of the variables are encoded by D^2. The most standard model of this kind is the Latent Dirichlet Allocation (LDA) [9] which considers K mixture components

(topics) $\{\phi\}$, where K is fixed and known. Each mixture component ϕ_k is a multinomial distribution over the vocabulary of size V. Here the level-2 groups (documents) are not clustered, i.e. Z^2 is distinct for each document. Consequently, θ^2 is not used here, and $\{\theta^1\}$ are group-specific. The Z^1-variables of the datapoints within any group j are assigned as IID draws from θ^1_j. Once again, no sequential structure is considered. Note that the mixture components ϕ are shared by all groups.

$$\phi_k \sim Dir(\beta), k \in [1, K]; \theta^1_j \sim Dir(\alpha), j \in [1, M]$$
$$Z^1_i \sim \theta^1_{D^2_i}, Y_i \sim \phi_{Z^1_i} \tag{7.1}$$

A non-parametric generalization of LDA is the Hierarchical Dirichlet Process (HDP) [76], which is also a 2-level extension of the DP-MM discussed above. Here, the number of components is not fixed or known, so the document-specific $\{\theta^1\}$-distributions are infinite-dimensional, and drawn from a Dirichlet Process/Stick-Breaking Process instead of finite-dimensional Dirichlet.

Another nonparametric 2-level model is the Nested Dirichlet Process (NDP), where the level-2 groups (documents) are clustered using Z^2, which are drawn according to a discrete distribution θ^2. Each cluster induced by Z^2 uses its own θ^1. However, unlike the previous models, here the mixture components themselves are specific to the clusters induced by Z^2.

$$\phi_k \in H \forall k; \theta^1_z \sim GEM(\kappa_1) \forall z; \theta^2 \sim GEM(\kappa_2)$$
$$Z^2(j) \sim \theta^2, j \in [1, M]; Z^1_i \sim \theta^1_{Z^2(D^2_i)}, Y_i \sim \phi_{Z^2(D^2_i), Z^1_i} \tag{7.2}$$

7.4.3 3-level models

Next, we look into some 3-level models. MLC-HDP [91] is an attempted compromise between HDP and NDP, where the groups are clustered (unlike HDP) but mixture components are not cluster-specific (unlike NDP), and moreover the data is grouped into 3 levels by observed group variables $\{(D^3, D^2, D^1)\}$. These groups can be clustered by random variables $\{Z^3\}, \{Z^2\}, \{Z^1\}$, which are drawn from discrete distributions θ^3,$\{\theta^2\}$,$\{\theta^1\}$ respectively.

A three-level model that considers the sequential nature of the data is the Topic Segmentation Model (TSM) [24]. Here within each document the sentences are clustered using $\{Z^2\}$, but analogous to HDP-HMM the distributions θ^2 are specific to values of Z^2. In particular, for any sentence s, θ^2 is a distribution over two values: $Z^2(prev(s))$ and $Z^2(prev(s))+1$ (to induce linear clustering/segmentation). The $\{\theta^1\}$ are specific to the sentence-clusters. The documents themselves are not clustered, so θ^3,Z^3 are not used.

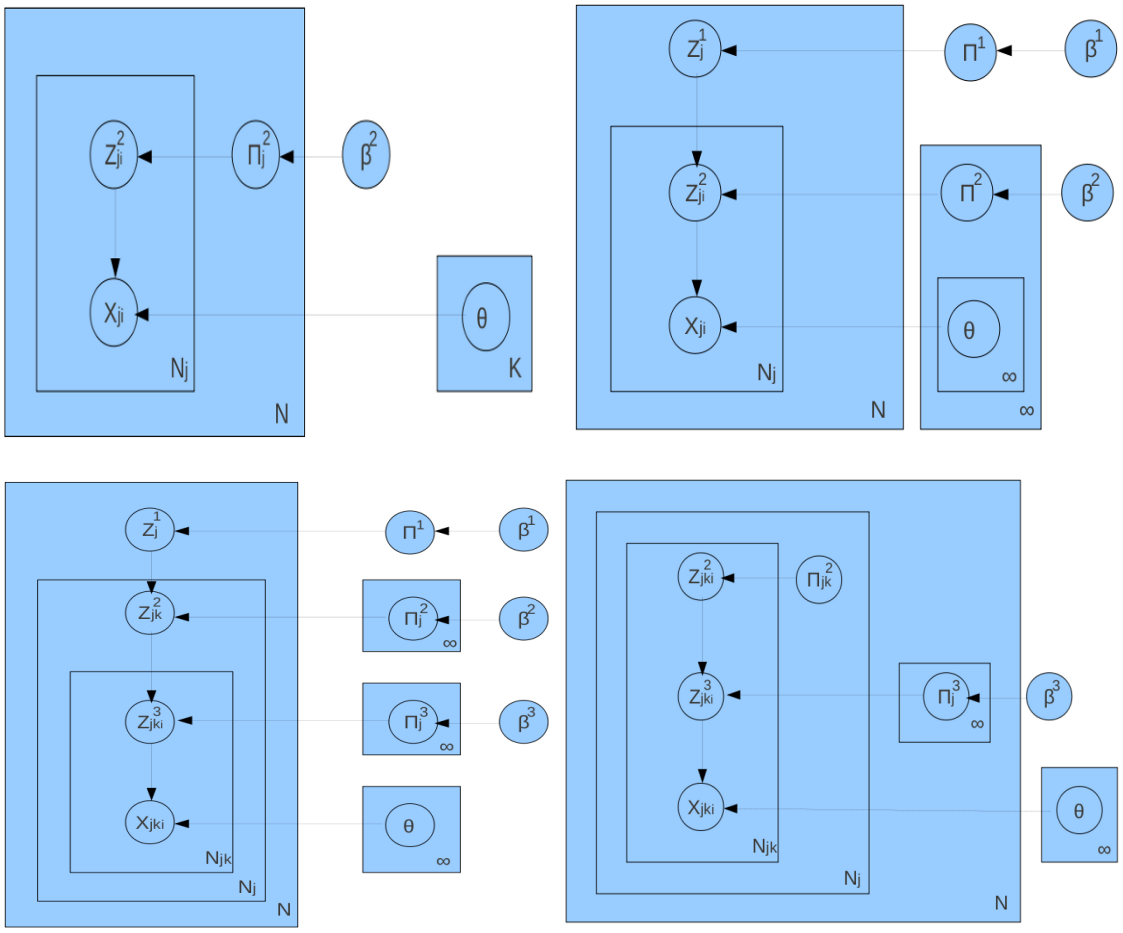


Figure 7.3: Above: HDP and NDP, Below: MLC-HDP and STM. The locations of the mixture components and distributions in the plate diagrams indicate the type of sharing (full/group-specific/cluster-specific)

A somewhat unusual case is Subtle Topic Model (STM) [19] which considers multiple document-specific distributions over the mixture components, and distributions specific to sentences over this set of distributions. Here neither the documents nor the sentences are clustered. Effectively, only $\{\theta^1\}$-distributions are present, which are shared across sentences in the same document, but not across documents. However, the process of assigning Z^1-variables requires other sentence-specific variables in addition to $\{\theta^1\}$.

7.5 DoS-classification of models

In the above discussion, we have focused on 3 major aspects- 1) Number of layers of grouping 2) the way in which the mixture components and mixture distributions are shared 3) Whether sequential structure is considered or not at different layers. Based on these aspects, we propose a nomenclature for the models.

7.5.1 DoS Concept

As already discussed, in all the hierarchical Bayesian models, the mixture components $\{\phi\}$ and the mixture components $\{\theta^1\}, \ldots, \{\theta^L\}$ are shared among the different groups. We have seen three types of sharing

1. *Full sharing (F)*: where components/distributions are shared by all the groups. For example, in HDP, MLC-HDP etc the mixture components are shared by all the level-2 groups.

2. *Group-specific sharing (G)*: where components/distributions are specific to groups, and not accessible outside the groups. For example, in HDP, STM etc the distributions θ^1 are specific to the top-level groups (documents).

3. *Cluster-specific sharing (C)*: where the components/distributions are specific to clusters of groups, but not accessible outside the clusters. For example, in MLC-HDP each θ^1-distribution is accessible to only one cluster of level-2 groups, and each θ^2-distribution is accessible to only one cluster of level-3 groups. In all the models, the mixture components are specific to clusters of level-1 groups (as datapoints are clustered by the assignment of a mixture component through Z^1).

Based on these notions we introduce Degree-of-sharing (DoS). For any given model, we first specify how the mixture components are shared at each of the levels- Full (F), group-specific (G) or cluster-specific (C), and we call this the DoS of $\{\phi\}$. The type of sharing at the different levels are hyphen-separated. Next, regarding the distributions $\{\theta^l\}$ at each level l, we specify

how it is shared by the levels $(l+1)$ upwards, and we call this the DoS of $\{\theta^l\}$. Also, to indicate if sequential structure is considered at the different levels, we add S to the levels where it is considered. Finally, to indicate how groups are clustered at different levels, we add N to the levels where there is no clustering of groups, P to the levels where the number of clusters is fixed, and NP to the levels where the clustering is non-parametric. Note that this indicates the dimensionality of $\{\theta^l\}$- P indicates that it is finite-dimensional, NP indicates it is infinite-dimensional, and N indicates it is not in use.

By combining the DoS of $\{\phi\}, \{\theta^1\}, \ldots, \theta^L$ in that order, we have the DoS-classification of the model. The DoS of the different variables are semicolon-separated. The number of components in any of these models is $(L+1)$, so that the DoS-classification of any model will have $(L+1)$ semicolon-separated parts. Also, the first part (corresponding to ϕ) will consist of L hyphen-separated letters, and the number of these letters will keep decreasing by one for each of the following parts (corresponding to θ^1, θ^2 etc), but followed by the letters specifying dimensionality and sequence structure.

7.5.2 Classification of Models

Let us illustrate the concept of DoS-classification with a case study of all the models discussed in Section 7.4.

Level-1 parametric models like GMM have mixture-components $\{\phi\}$ specific to clusters of datapoints, so that its DoS is C. But the mixture distribution θ^1 is fully shared by all the datapoints, so that its DoS is F. The number of clusters formed at level 1 (i.e. the dimensionality of θ^1) is fixed (i.e. P) and sequential structure is not considered. Hence the DoS-classification of GMM is $C; F-P$. In case of DP-MM, θ^1 is infinite- dimensional (NP), i.e. DoS-classification is $C; F-NP$.

In case of HDP-HMM, the mixture-components $\{\phi\}$ are again specific to clusters of datapoints, so that its DoS is C. Here, the $\{\theta^1\}$ are non-parametric (NP), and the sequential structure is also considered. So, the DoS-classification of HDP-HMM is $C; F-NP-S$. Note that here $\{\theta^1\}$ is a collection of distributions, from which one is chosen for each data-point i, depending on the assignment to $prev(i)$.

Level-2 models: In HDP or LDA, the mixture-components are shared cluster-specific in level-1 and fully at level-2, so the DoS for ϕ is $C-F$. The $\{\theta^1\}$ are specific to level-2 groups, so the DoS for θ^1 is G. Sequential structure is not considered at any level. In case of LDA the number of clusters of datapoints (level-1) is fixed (P), and for HDP it is NP. The level-2 groups are not clustered (N) in either model. So we say the DoS-classification of LDA is $C-F; G-P; N$, and for HDP it is $C-F; G-NP; N$. In case of NDP, the $\{\phi\}$ are cluster-

specific at both levels, so its DoS is $C - C$. θ^1 is specific to clusters of level-2 groups (C), and it is non-parametric (NP). The θ^2 are also non-parametric (NP). So, the DoS-classification is $C - C; C - NP; NP$.

Level-3 models: In MLC-HDP, the mixture-components $\{\phi\}$ are cluster-specific in level-1, but fully at both levels 2 and 3, so that its DoS is $C - F - F$. The $\{\theta^1\}$ are specific to clusters of level-2 groups but fully shared by level-3 groups, and they are nonparametric, so the notation is $C - F - NP$. The $\{\theta^2\}$ are specific to clusters of level-3 groups and nonparametric, and finally θ^3 is nonparametric. So the DoS-classification of MLC-HDP is $C - F - F; C - F - NP; C - NP; NP$.

For Topic-segmentation model, the topics $\{\phi\}$ are shared by all sentences and documents, i.e. its DoS is $C - F - F$. The $\{\theta^1\}$ are specific to clusters of sentences inside individual documents, i.e. the DoS is $C - G$, and they are of fixed dimension (P). The $\{\theta^2\}$ used to cluster sentences is document-specific (G). The number of clusters (segments) of sentences to be formed is not fixed, and sequential structure is also taken into account, so the notation is $G - NP - S$. Finally, the documents themselves are not clustered (N), and the DoS-classification of TSM is $C - F - F; C - G - P; G - NP - S; N$.

Finally we come to Subtle Topic Model (STM), where the topics are shared by all sentences and documents, i.e. the DoS is $C - F - F$. The θ^1 are shared by all sentences in a document but are specific to documents, and they are nonparametric, i.e. the notation is $F - G - NP$. The sentences and documents are not clustered, so the DoS-classification of STM is $C - F - F; F - G - NP; N; N$.

7.6 Generalized Bayesian Model for Grouped Sequential Data

Having discussed the DoS-classification of various existing models, it is clear that despite over a decade of research on topic models, there are several DoS-classifications for which there are no existing models. But instead of trying to point out those classifications individually and propose models following them, we now propose a generalized Bayesian Model for grouped sequential data. We will show that by specific settings of this model, it is possible to recover all the previously discussed models (or their close variants). Other models, not explored so far, can also be obtained from it.

7.6.1 GBM-GSD

We consider sequential data with L-levels of grouping, where the groups are sequential in every level (eg. in document modeling, we will consider the sentences within each document, and

the documents themselves, are sequentially arranged). We consider that clustering happens at all levels, i.e. $\{\theta^1\}, \ldots, \{\theta^L\}$ all exist. To capture the sequential nature, we will assume that at every level (say l), there is a collection of distributions $\{\theta^l\}$ from which one can be chosen for each group, conditioned on the previous assignments (as considered in sHDP-HMM and TSM). We also consider that all the distributions are infinite-dimensional (i.e. NP), i.e. neither the number of mixture components nor the number of clusters formed at each level is known in advance. We also consider that all the mixture components are accessible to all the level-2 groups, but introduce a binary random vector B_i specific to each datapoint. This vector indicates which all mixture components are accessible to each datapoint. We will show that using this vector, we can make the mixture components group-specific or cluster-specific, and also capture other more intricate structures that would not be possible without it. The generative process hierarchically clusters the groups from top to bottom level. At

Algorithm 8 Generalized Bayesian Model for Group Sequential Data (GBM-GSD)

$\phi_k \sim H, \forall k$
for $g = 1 : G_L$ **do**
  $Z^L(g) \sim \theta^L | Z^L(1), \ldots, Z^L(prev(g))$
end for
for $l = L - 1 : 2$ **do**
  for $g = 1 : G_l$ **do**
    $Z^l(g) \sim \theta^l_j | Z^l(1), \ldots, Z^l(prev(g))$ where $j = Z^{l+1}(D^{l+1}(g))$
  end for
end for
for $i = 1 : N$ **do**
  $B_i = f(B_1, \ldots, B_{i-1}, Z^1_1, \ldots, Z^1_{i-1}, Z^2, \ldots, Z^L)$
  $Z^1_i \sim B_i \circ \theta^1_j | Z^1_1, \ldots, Z^1_{i-1}$ where $j = Z^2(D^2_i)$
  $Y_i \sim \phi_k$ where $k = Z^1_i$
end for

every intermediate level l, it assigns $Z^l(g)$ to each group g at level l. But for that it will have access to only those θ^l-distributions, that are specific to the cluster Z^{l+1}_g of group g as a result of the clustering at level $(l+1)$. If group g is part of group $m = D^{l+1}(g)$ at level $(l+1)$ then the θ^l-distributions corresponding to $Z^{l+1}(m)$ must be used. Finally, at level 1, each datapoint i is assigned a binary vector B_i conditioned on the B-vectors corresponding to all previous datapoints. The distribution θ^1 is convoluted with this vector B_i, so that a subset of the components are available to datapoint i.

7.6.2 Recovery of Existing Models

The level-1 models can be recovered easily. By setting B_i as a vector of all 1-s for all the datapoints, and by making θ^1 conditioned only on $Z^1_{prev(g)}$ and GEM-distributed we get back HDP-HMM. In case θ^1 is also independent of the previous assignments, we can have DP-MM, and if it is finite-dimensional it will GMM provided the base distribution H is Gaussian.

When $L = 2$, to recover HDP we need to define θ^L such that $Z^L(g) = g$ for all the groups g, so that groups are not clustered. Then we again set B_i to be the vector of all 1-s, and make $\{\theta^1\}$ independent of all previous assignments of Z^1. The $\{\theta^1\}$ should be drawn from a GEM. If the $\{\theta^1\}$ are finite-dimensional and drawn IID from a Dirichlet, and if all the $\{\phi_k\}$ are also drawn from a Dirichlet, then we have LDA.

NDP involves nonparametric clustering of level-2 groups without sequential ordering, so generation of $Z^L(g)$ should be independent of previous assignments, and θ^L should be drawn from a GEM. NDP also has the special characteristic that the different level-2 clusters do not share the same mixture components. This can be managed by setting B_i through an appropriate function f, which will return a vector with 0 for those mixture components that have been assigned in other level-2 clusters, i.e. $B_{ik} = 0$ if $\exists j$ such that $Z^2_{D^1_j} \neq Z^2_{D^1_i}$ and $Z^1_j = k$.

When $L = 3$, MLC-HDP can be recovered by removing the conditioning on previous assignments in the assignment of Z^3, Z^2 and Z^1, and by setting B_i to be the vector of all 1-s. The $\{\theta^l\}$ should be drawn from GEMs. For TSM, θ^3 should ensure that documents are not clustered, B_i should be the vector of all 1-s, and assignment of Z^1 should be independent of all previous assignments. Regarding Z^2, θ^2 should ensure that for any sentence (level-2 group) g, $Z^2(g)$ should be either $Z^2(prev(g))$ or $Z^2(prev(g)) + 1$.

7.7 Layered Dirichlet Process

We now discuss a particular instantiation of GBM-GSD. We call this as the Layered Dirichlet Process (LaDP). We consider versions of it, depending on whether or not it models sequential structure.

7.7.1 LaDP Generative Process

We define a joint probability distribution over the N sets of cluster variables hierarchical Bayesian approach. For each layer l, $1 \leq l \leq L$, we have a countable set of measures $\{\theta^l_j\}_{j=1}^{\infty}$ defined over positive integers. The cluster variables $\{Z^l_i\}_{i=1}^{N}$ at layer l serve as indexes for these measures. Using this countable property, the atoms of all of these measures at layer l, which are integers, correspond one-to-one with the measures at the next layer $l-1$. This gives us a hierarchy of measures, in the sense that each θ^l_j forms a measure over the measures $\{\theta^{l-1}_{j'}\}_{j'=1}^{\infty}$ at the next layer. Finally, at the lowest layer, each ϕ_k is a measure over the space $\mathcal{Y}$ of the observations $\{Y_i\}$. For discrete text data, these are multinomial distributions over the vocabulary.

Next we need to define the measures $\{\theta^l_j\}_{j=1}^{\infty}$ and the sequential properties at each layer l. In LaDP, we define each of these distributions to be DP-distributed. We begin with the simplest case, which assumes complete exchangeability at every layer, i.e. does not consider sequential

structure. The generative process looks as follows:

$$\begin{aligned}
\phi_k &\sim H,\ k = 1 \ldots \infty \\
\beta_j^l &\sim GEM(\gamma^l);\ \theta_j^l \sim DP(\alpha^l, \beta_j^l),\ j = 1 \ldots \infty,\ l = L \ldots 1 \\
Z^l(g) &\sim \theta^l_{Z^{l+1}(g')},\ g = 1 \ldots G_l,\ \ l = L \ldots 1,\ \ Y_i \sim \phi_{Z_i^1},\ i = 1 \ldots N
\end{aligned} \tag{7.3}$$

Here, $g' = Z^{l+1}(D^{l+1}(g))$ In each layer l, a countable set of measures θ^l is first constructed by drawing from a DP with a distribution over integers as a base distribution. These measures as a result also have support over integers, which serve as indexes to the measures at the next lower layer, which also form a countable set. Once we have this hierarchy of measures, the cluster variable $Z^l(g)$ for each group at each layer l is sampled from the measure indexed by the group's cluster $Z^{l+1}(g)$ assigned at the previous (higher) layer. The measures at the lowest layer (layer 1) are sampled from a suitable base distribution H. H could be Dirichlet when each ϕ_g is a multinomial parameter.It is easy to verify that the above process satisfies Complete Exchangeability (CE). As such, we call this model the CE-LaDP.

Layered Dirichlet Process with sequential structure Since CE models do not capture sequential structure of data, they are not useful for segmentation. We next incorporate sequential structure within LaDP. The key to this is to relax the *iid* assumption for the group variables, within a layer as in the HDP-HMM, and additionally across layers, and generate $Z^l(g)$ conditioned on some of the previously sampled groups $\{Z^{l'}(g') : g' < g,\ l' > l\}$. The HDP-HMM makes the Markovian independence assumption that $P(Z_i|Z_{<i}) = P(Z_i|Z_{i-1})$. Accordingly, it defines transition distribution θ_j over next states for each state j.

In our case, we make the following independence assumption: $P(Z^l(g)|Z^{>l}, Z^l(< g)) = P(Z^l(g)|Z^{l+1}(g), Z^l_{p(g,l)})$, where $Z^{>l} \equiv \{Z^{l'}(g) :\ l' > l\}$, $Z^l(< g) \equiv \{Z^l(g') :\ g' < g\}$, and $p(g,l) \equiv \{j : Z^{l+1}(g') = Z^{l+1}(g), g' < g,\ Z^{l+1}(k) \neq Z^{l+1}(g), g' < k < g\}$ is the previous group in the l-layer having same cluster as g at layer $l+1$. This means that the cluster assignment to group g at layer l depends on its cluster at the layer $l+1$ (like in CE-LaDP), and also on the group assignment at layer l of its parent group $p(g,l)$. We later overload the notation $p(g,l)$ for brevity to refer to the group value $Z^l(p(g,l))$ as well. We accordingly define transition distribution $\theta^l_{j,j'}$ over next cluster index from each previous cluster j' at layer l, in each assigned cluster j in layer $(l+1)$. The generative process for layer l $(L \geq l \geq 1)$ is defined as:

$$\begin{aligned}
\beta_j^l \sim GEM(\gamma^l),\ \theta^l_{j,j'} \sim DP(\alpha^l, \beta_j^l),\ j, j' = 1 \ldots \infty, \\
Z^l(g) \sim \theta^l_{Z^{l+1}(g),p(g,l)},\ g = 1 \ldots G_l
\end{aligned}$$

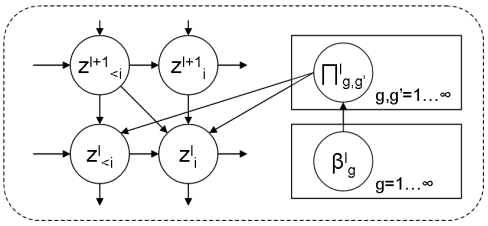


Figure 7.4: Graphical model of LaDP focused on the i^{th} data point in two adjacent layers

Part of the graphical model is shown in Fig. 7.4.

For the first data point in any group in layer $(l+1)$, $p(g,l)$ is undefined, and $Z^l(g)$ is sampled from $\beta^l_{Z^{l+1}(g)}$. It can be shown that this generative process satisfies ME within each group at layer l. When this process is used at all layers, we call the model ME-LaDP. As in sticky HDP-HMM, we may add more probability κ^l for self-transitions: $\theta^l_{j,j'} \sim DP(\alpha^l + \kappa^l, \frac{\alpha^l \beta^l_j + \kappa^l \delta_{j'}}{\alpha^l + \kappa^l})$, where κ^l is a *c*ontinuity parameter. This is done to encourage the same mixture component for adjacent data points. This captures the temporally smooth nature of most real-world data, and also encourages segmentations (based on group index assignments).

Layer-specific Exchangeability: We have defined CE-LaDP as using CE at all layers, and ME-LaDP as using ME at all layers. However, each of the processes can be defined specific to a single layer, and it is possible to use layer-specific exchangeability assumptions, as demanded by particular applications. Indeed, we use such *mixed exchangeability models* in our experiments.

Incorporation of Domain Knowledge: The $\{\beta^l_j\}$ variables at each layer l in Eqns. 7.3 and 7.4 are group-specific distributions over indexes (and measures) at the next layer $l-1$. These are useful for incorporating domain knowledge such as distribution over topics for specific documents. For example, we can indicate that the j^{th} document is dominated by cluster-index k at layer L by setting distribution β^L_j over category indexes at the appropriate layer to $\sum_{c=1}^{\infty} \delta_k(c)$.

In some cases, one may also wish to bias the ϕ-distributions using domain knowledge. One option is to directly specify these ϕ_k. For weaker supervision, we may introduce an additional layer $l = 0$:

$$\begin{aligned} \beta^0_j \sim GEM(\gamma^0);\ \theta^0_j \sim DP(\alpha^0, \beta^0_j), \quad j = 1 \ldots \infty \\ Z^0_i \sim \theta^0_{Z^1_i};\ Y_i = \mathcal{W}_{Z^0_i}, \quad i = 1 \ldots n \end{aligned} \tag{7.4}$$

Now, on specifying some of the $\{\beta^0_j\}$ distributions, the corresponding distributions $\{\theta^0_j\}$ will be similar to these, depending on the concentration parameter α^0, and the data-points will be drawn from these $\{\theta^0_j\}$ distributions. Observe that we use complete (group) exchangeability at

layer $l = 0$.

Relation with Other Models: Observe the relation between the CE-LaDP (Eqn. 7.3) and the HDP mixture model (Eqn. 2.5). Recall that the group at the highest layer Z_i^{L+1} is the input group label D_i^{L+1}. For $L = 1$, this is exactly the HDP mixture model. However, by separating the group index in the HDP generative model, and identifying the Z_i variable as the random group variable leading to the next layer, the CE-LaDP naturally extends the HDP generative process to generate layered grouping. A similar relation holds between the ME-LaDP with $L = 1$ and the HDP-HMM. The MLC-HDP [91] extends HDP to 3 layers, with each data point Y_i having input group indices D_i^3, D_i^2, D_i^1. When the cluster indices Z^3 are observed (rather than sampled from θ^3, as in [91]) and are identified with the indices D_i^3 for LaDP, and additionally the input group indices $D_i^3 = i$, $D_i^2 = 1$ and $D_i^1 = 1$ are shared by data points Y_i, we get back the CE-LaDP with $L = 2$. Thus the LaDP framework can be used to generalize existing models to any number of layers. Secondly, the LaDP enables incorporation of domain knowledge in all layers. Among existing models, only the recently-proposed DP-MRM [43] is equipped to incorporate such domain knowledge, though only for a single layer. Finally, while all existing methods only use a single exchangeability property (CE or ME), LaDP has the attractive property that different layers can have different exchangeability properties. In the next section, we define a new notion of exchangeability, and show how it can be incorporated in any layer of LaDP.

Layered Chinese Restaurant Process (LaCRP) In hierarchical Bayesian non-parametric models, the conditional distributions of latent variables, given assignments to earlier ones are typically associated with restaurant analogies. For the LaDP, we may consider a hypothetical restaurant that has layers consisting of infinite number of tables, each layer possibly corresponding to one course in the menu. Each customer, unlike in a formal dinner, has to move from one layer to the next after each course. The restaurant has multiple entrances, corresponding to each input group, and in the first layer, each customer randomly chooses a table based on table assignments of previous customers who came in through the same entrance. After completing the i^{th} course, each customer randomly chooses a table for the next $(i+1)^{th}$ course based on tables assigned to previous customers who shared his table in the i^{th} course.

7.7.2 Inference using LaDP

The inference problem in LaDP, given observations $\{Y_i\}$, is to find posterior distributions over the group variables $\{Z_i^l\}$ at all layers l for each data point. As for models such as HDP, HDP-HMM and sHDP-HMM, exactly computing this posterior distribution is not tractable, and we

resort to Gibbs Sampling for approximate inference as for the other models. One possibility is to perform collapsed Gibbs Sampling using only the group variables after integrating out all the parameter variables such as θ_j^l and β_j^l. When the β_g^l variable takes the same value across groups in any layer l, the distribution of the variables at that layer is identical to the HDP. The predictive distribution of the Z_i^l in that case is given by the CRF equations as for the HDP [76]. However, in cases where some of the β_j^l distributions are specified through domain knowledge, we integrate out only the θ_j^l distributions.

Predictive Distributions: For the different LaDP models, we first derive the predictive distributions for Z_i^l, the i^{th} group variable in the l^{th} layer, given the assignments to all group variables in the layers above (denoted $Z^{>l}$), and the first $i-1$ group variables in layer l (denoted $Z_{<i}^l$), after integrating out the $\theta_{j,j'}^l$ distributions from which they are drawn.

If the l^{th} layer uses CE (Eq. 7.3), the predictive distribution is given by

$$p(Z_i^l = a | Z_{<i}^l, Z^{l+1}) \propto n^l_{Z_i^{l+1},i,a} + \alpha^l \beta^l_{Z_i^{l+1}}(a) \tag{7.5}$$

where $n^l_{j,i,a} = |\{t : Z_t^l = a, Z_t^{l+1} = j, t \in [1, i-1]\}|$. This is the number of data points before datapoint i in group j of layer $l+1$ were assigned to group a in layer l.

If the l^{th} layer uses ME (Eq. 7.4), the predictive distribution becomes

$$p(Z_i^l = a | Z_{<i}^l, Z^{l+1}) \propto n^l_{Z_i^{l+1},i,p(i,l),a} + \kappa\delta(p(i,l),a) + \alpha^l \beta^l_{Z_i^{l+1}}(a) \tag{7.6}$$

where $n^l_{j,i,b,a} = |\{t : Z_t^l = a, p(t,l) = b, Z_t^{l+1} = j, t \in [1, i-1]\}|$ is the number of times successive data points before datapoint i in group j of layer $l+1$ assigned to groups b and a respectively in layer l.

Inference using Gibbs Sampling: We sample each of the Z_i^l variables conditioned on all the others sequentially in each iteration until convergence. In each iteration we traverse all group variables for one data point before moving to the next data point, and for a specific data point we traverse layers top down. The conditional distribution is given by $p(Z_i^l | Z_{-i}^l, Z^{l-1}, Z^{l+1}) \propto p(Z_i^l | Z_{-i}^l, Z^{l+1}) p(Z^{l-1} | Z^l)$. The second term can be computed using the chain rule and the predictive distributions described above: $p(Z^{l-1}|Z^l) = p(Z_1^{l-1}|Z_1^l) \prod_{i=2}^N p(Z_i^{l-1}|Z_{<i}^{l-1}, Z^l)$. At layer $l = 1$ this is the likelihood of the data, conditioned on the table assignments of layer 1. The form of the first term depends on the exchangeability assumption.

If layer l uses CE the i^{th} variable can be swapped with the last to get

$$p(Z_i^l = a | Z_{-i}^l, Z^{l+1}) \propto n^l_{-i,Z_i^{l+1},a} + \alpha^l \beta^l_{Z_i^{l+1}}(a) \tag{7.7}$$

where $n^l_{-i,j,a} = |\{t \neq i : Z^l_t = a, Z^{l+1}_t = j\}|$. Swapping is possible by CE property.

If layer l uses ME with sticky transitions, we make use of the conditional distribution for the sHDP-HMM [30] to get:

$$p(Z^l_i = a | Z^l_{-i}, Z^{l+1}) = (\alpha^l \beta^l_j(a) + s(p(i,l),a) + \kappa\delta(p(i,l),a)) \times \frac{\alpha^l \beta^l_j(c(i,l)) + s(a, c(i,l)) + \kappa\delta(c(i,l),a) + \delta(c(i,l),a)\delta(p(i,l),a)}{\alpha^l + s(a,.) + \kappa + \delta(p(i,l),a)} \quad (7.8)$$

where $j = d^{l+1}_i$, $s(a,b) = |\{t : Z^l_t = a, c(t,l) = b\}|$. $p(i,l)$ is as defined before Eqn. 7.4, and $c(i,l)$ is defined analogously with $i+1 \leq j \leq n$ instead of $1 \leq j \leq i-1$.

7.8 News Transcript Segmentation

We want to extend the generative framework for grouped sequential data (Algorithm 1) for modeling news transcripts. This data is hierarchical since there are broad news categories like politics, sports etc, under which there are individual stories or topics. In the Bayesian approach, we consider mixture components $\{\phi\}$ that correspond to these stories, and the broad categories are represented with distributions $\{\theta^1\}$ over these stories. As usual, each θ^1-distribution is specific to a level-2 cluster (segment), and such clustering is induced by $\{\theta^2\}$, specific to the level-3 groups (the transcripts). The transcripts are not clustered. The observed datapoints Y_i are word-tokens, each represented as an integer (index of the word in the vocabulary). We define $prev(i) = i-1$ if $(i,\ i-1)$ are in the same sentence, otherwise $prev(i) = -1$. Similarly, $next(i)$ is defined within sentences. Also, $prev$ and $next$ are defined for sentences. Z^1_i indicates the news story (level-1) and Z^2_i indicates the news category (level-2) that token i is associated with. Each sentence is a level-2 group.

News transcripts can modelled by Layered Dirichlet Process (LaDP). To capture temporal coherence, we must use ME-LaDP at both level-1 and level-2, i.e. the assignment of Z^1_i and Z^2_i are conditioned on $Z^1_{prev(i)}$ and $Z^2_{prev(i)}$ respectively. The DoS-classification of this version of LaDP is $C-F-F; C-F-NP-S; F-NP-S; N$.

7.9 Bayesian Modeling of News Transcripts

News transcripts have characteristic temporal features regarding assignments of Z^2 and Z^1. LaDP is insufficient for news transcripts, because it does not capture all of them. To model these, GBP-GSD needs to be modified appropriately. These features are discussed below.

7.9.1 Semi-Markov Modeling at Category-level

In case of news transcripts from a particular source, it can be expected to have K news categories in fixed order (say politics, national affairs, international affairs, business and sports, in that order). So, the number of Level-2 clusters (segments) are fixed and known. **Segmentation** is the task of *linear clustering* of words/sentences, i.e. each word/sentence s can be assigned to either $Z^2_{prev(s)}$ or to $Z^2_{prev(s)} + 1$. In LaDP, each datapoint i is assigned a value of Z^1_i and Z^2_i based on the assignments of $prev(i)$, and segmentation happens based on these assignments. But this does not guarantee the formation of K segments. To overcome this issue, let it be known to model that the observed data sequence has K level-2 segments. Then the sequence can be partitioned into K parts of sizes $N_1, N_2, \ldots, N_K$. These sizes may be modeled by a Dirichlet distribution where the parameters γ_k signify the relative lengths/importance of the news categories.

$$\{\frac{N_1}{N}, \ldots, \frac{N_K}{N}\} \sim Dir(\gamma_1, \ldots, \gamma_K); Z^2_j = s \quad \textbf{where } \sum_{k=1}^{s-1} N_k < j \leq \sum_{k=1}^{s} N_k \tag{7.9}$$

In the GBS-GSD, the θ^2 needs to be defined as a deterministic function, conditioned on $\{N_1, \ldots, N_K\}$. It may be noted that this is a **Semi-Markovian** (explicit-duration) approach, instead of the Markovian approach of LaDP.

7.9.2 Temporal Structures at Topic-level

Temporal Coherence of topics has been considered in very few text segmentation papers like [25]. This is the property that within the same level-2 segment, successive datapoints are likely to be assigned to the same topic (mixture component). This can be easily modelled by the Markovian approach, i.e.

$$Z^1_i \sim \rho\delta_{Z^1_{prev(i)}} + (1-\rho)(B_i \circ \theta^1_s) \textbf{ where } s = Z^2_{D^2_i} \tag{7.10}$$

This means that the i-th datapoint can be assigned the Z^1-value of its predecessor $pred(i)$ with probability ρ, or any value with probability $(1-\rho)$. The other available values are dictated by B_i, as discussed next.

Level-2 segments do not share topics, because each individual news story (topic) can come under only one news category. Also, **Topics do not repeat inside a Level-2 segment**. Inside a level-2 segment s, successive datapoints are expected to be assigned to the same mixture

component due to temporal coherence. However, in news transcript, a news story will be told only once, which means that *a particular component may be present only in a single chunk, and cannot reappear in non-contiguous parts of the segment.* For this purpose the generative process needs to be manipulated through B_i. Initially we set B_i to be all 1s, and whenever a component ϕ_k is sampled for any point, we set $B_{ik} = 0$ for all following points in the segment, so that ϕ_k cannot be sampled again. The generative process is as follows:

Algorithm 9 Generative Model for News Transcripts

$H_c \sim Dir(\beta)\ \forall c$
$c \sim U(K)$, $\phi_k \sim H_{c(k)}\ \forall k$
$\theta^1_s \sim GEM(\alpha)$ where $s \in [1, K]$
for $g = 1$ to G^3 **do**
    $B_{gk} = 1\ \forall k$
    $\{\frac{N_{g1}}{N_g}, \ldots, \frac{N_{gK}}{N_g}\} \sim Dir(\gamma)$
end for
for $j = 1$ to G^2 **do**
    $Z^2_j = s$ based on $(N_{g1}, \ldots, N_{gK})$ where $g = D^3_j$
end for
for $i = 1 : N$ **do**
    if $Z^2_{D^2(i)} \neq Z^2_{D^2(prev(i))}$ set $\rho = 0$
    $Z^1_i \sim \rho\delta_{Z^1_{prev(i)}} + (1-\rho)(B_g \circ \theta^1_s)$ where $s = Z^2_{D^2_i}$, $g = D^3_i$
    if $(Z^1_i \neq Z^1_{prev(i)})$ set $B_{gk} = 0$ where $k = Z^1_i$, $g = D^3_i$
    $Y_i \sim mult(\phi_k)$ where $k = Z^1_i$
end for

Here G^3 is the number of transcripts, and G^2 the number of sentences across all the transcripts. Clearly this model has 3 levels, and sequential structure is considered at level 2 (sentences) and at level 1 (word-tokens). Any topic k belongs to a broad category $c(k)$ ($\in \{1, \ldots, K\}$ uniformly at random), and corresponding to each category we have a base distribution H_c, which in turn are all drawn from a common base distribution $Dir(\beta)$. This helps to capture the fact that mixture components are specific to level-2 segments. The documents are not clustered, the sentences are clustered (segmented) with fixed number of segments, and the number of topics (word-clusters) is not fixed. The topics are shared across all transcripts, but are specific clusters of sentences, the θ^1-distributions are specific to level-2 segments (clusters of sentences) but shared across transcripts, the θ^2-distributions are transcripts-specific (parametrized by $\{N_g\}$) and θ^3 are not used. So the DoS-classification for the generative model of news transcripts is $C - C - F; C - F - NP - S; G - P - S; N$.

7.9.3 Inference Algorithm

We now discuss inference for this model. We need an inference algorithm which ensures that K segments are formed. We start with the joint distribution.

$$\begin{aligned} p(Y, Z^1, Z^2, B, N, \Phi, \beta, \{\theta\}, \{H\}) \propto \prod_{g=1}^{G^3} p(\{N_g\}) \prod_{s=1}^{K} p(\theta_s^1) \\ \times \prod_c p(H_c|H) \prod_k p(\phi_k|H_{c(k)}) \prod_{j=2}^{G^2} p(Z_j^2|Z_1^2, \ldots, Z_{prev(j)}^2, \{N\}) \\ \times \prod_{i=1}^{N} p(Z_i^1, B_{next(i)}|Z_{prev(i)}^1, B_i, Z_i^2, \{\theta^1\}) p(Y_i|Z_i^1, \Phi) \end{aligned} \tag{7.11}$$

We can collapse the variables $\{H\}$,$\{\Phi\}$, and $\{\theta^1\}$, and perform Gibbs Sampling. The key feature of this likelihood function is the presence of the $\{N_g\}$ variables. To handle these, we introduce auxiliary variables $I_{g1}, \ldots, I_{g,K-1}$ which are the level-2 changepoints, i.e. the set of datapoints $\{i\}$ at which $Z_i^2 \neq Z_{prev(i)}^2$. Also note that $\{Z^2\}$, $\{I\}$ and $\{N\}$ are deterministically related. We introduce the I variables to simplify the sampling. We initialize the Z^2 variables by sampling a level-2 segmentation of the datapoints into K segments. The B and Z^1 variables are sampled accordingly. In each iteration of Gibbs Sampling, we consider the state-space of I_{gs} as $I_{gs} \in \{I_{g,s-1}, \ldots, I_{g,s+1}\}$, i.e. the level-2 potential changepoints lying in between $I_{g,s-1}$ and $I_{g,s+1}$. The process is described in Algorithm 10. Here, $B_{gs} = \{B_{set}\}$ where $set = \{i : D_i^3 = g, Z_i^2 = s\}$, i.e. the set of datapoints in transcript g in segment s. (similarly Z_{gs}^2, Z_{gs}^1, Y_{gs}) The major part in the Gibbs sampling is to sample the values $(\{B\}_s, \{Z^1\}_s)$ for any segment s, conditioned on the remaining B and Z^1 variables. This can be done using the Chinese Restaurant Process (CRP), where any component k may be sampled for Z_i^1 (where datapoint i is within segment s) proportional to the number of times it has been sampled, provided $B_{prev(i),k} = 1$. The procedure is detailed in Algorithm 3, which is called Global Inference as it considers the overall structure of the transcript (as described in Sec7.9).

7.10 Experiment on News Transcript Segmentation

We crawled archived pages from 5 news websites (Yahoo! News, The Hindu, The Times of India, Deccan Herald, The Telegraph) for a 30 day period (April 1-30, 2012), where news articles for each day were arranged in sequence like news transcripts. We selected stories from 5 categories — politics, national affairs, international affairs, business and sports, to create one transcript for each day for each news source. This produced a dataset of 150(30×5) virtual news transcripts, consisting of 2600 individual news articles, spread over the 5 categories. From these, 60 transcripts were used for training and the rest for testing. After eliminating stop-

Algorithm 10 Global Inference Algorithm by Blocked Gibbs Sampling (GI-BGS)

for transcript $g = 1$ to G^3 **do**
  Initialize I_g with $(K-1)$ points by sampling from $Dir(\gamma)$;
  Set $\{Z^2\}$ according to I;
  Initialize $\{B\}$,$\{Z^1\}$ variables;
end for
Estimate components $\hat{\phi} \leftarrow E(\phi|Z, B, S, Y)$;
while Not Converged **do**
  for transcript $g = 1$ to G^3 **do**
    for segment $s = 1 : K$ **do**
      $I_{gs} \in \{succ(I_{g,s-1}), \ldots, pred(I_{g,s+1})\} \propto p(Y_{gs}|B_{gs}, Z^1_{gs}, Z^2_{gs}, \hat{\phi})$;
      Update Z^2 according to I
      $(\{B\}_{gs}, \{Z^1\}_{gs}) \propto p(\{B\}_{gs}, \{Z^1\}_{gs}|\{B\}_{-gs}, \{Z^1\}_{-gs}, Z^2, Y, \hat{\phi})$;
    end for
  end for
  Update components $\hat{\phi} \leftarrow E(\phi|B, Z^1, Z^2, Y)$;
end while

words and rare words, we had a vocabulary of size 7204, with a total of 0.4 million tokens in the complete dataset.

In this dataset, the datapoints per sequence are too few in number to learn the level-1 mixture components (topics). Moreover, as already explained, each story occurs only once in a transcript, thus reducing learnability. Hence, we considered 60 randomly chosen transcripts, and using initial segmentations of each sequence by the level-1 changepoints, 136 topics were learnt using HDP. These topics form our initial estimate of Φ, using which we performed inference on individual sequences. The inference provides us with the Z^2 and Z^1 variables, based on which we can infer the segmentation at the two levels. We have gold standard segmentation available at both layers, and so we compute the segmentation errors $(S1, S2)$ at both layers. Segmentation error is the probability that to word-tokens, placed k positions apart, are in the same gold-standard segment but in different inferred segments, or vice versa. S1 and S2 are computed by taking the average segmentation error for three different values of k, namely the maximum, minimum and average lengths of gold-standard segments (level-1 segments for S1 and level-2 segments for S2).

We can look upon segmentation as a *retrieval problem*, and define the *Precision and Recall of level-2 segments (PR2 and RC2)*, and also for level-1 segments (PR1 and RC1). Let i and j be the starting and ending points of an inferred segment s, i.e. $Z^2_i = Z^2_{next(i)} = \cdots = Z^2_j = s$, but $Z^2_{prev(i)} \neq s$ and $Z^2_{next(j)} \neq s$. Then, if there exists $(i0, j0, s0)$ such that $(i0, j0)$ defines a gold-standard segment $s0$ satisfying $|i - i0| < k$ and $|j - j0| < k$, then inferred segment (i, j, s) is *aligned* to gold standard segment $(i0, j0, s0)$. Precision, recall of a segmentation are defined as

$$\textbf{Precision} = \frac{\textbf{\#inferred segments aligned to a gold-standard segment}}{\textbf{\#inferred segments}}$$

Data	GI-BGS			LaDP			sHDP-HMM		
	PR1	RC1	S1	PR1	RC1	S1	PR1	RC1	S1
Trans1	**0.38**	**0.46**	**0.06**	0.33	**0.46**	0.07	0.20	0.40	0.08
Trans2	**0.33**	**0.37**	**0.10**	0.27	0.34	0.11	0.18	0.34	0.12
Trans3	**0.26**	**0.41**	0.09	0.25	**0.41**	**0.08**	0.13	0.32	0.11
Trans91	**0.15**	**0.28**	0.16	0.13	**0.28**	**0.13**	0.06	0.21	0.16
Trans92	**0.14**	**0.25**	0.14	0.10	0.20	0.14	0.08	0.23	0.14
Trans93	**0.22**	**0.22**	**0.09**	0.17	0.08	0.11	0.12	0.03	0.11

Data	PR2	RC2	S2	PR2	RC2	S2
Trans1	0.20	0.20	**0.08**	**0.33**	**0.40**	0.11
Trans2	**0.80**	1.00	0.04	0.71	1.00	**0.01**
Trans3	1.00	1.00	0.13	1.00	1.00	**0.04**
Trans91	**0.60**	0.60	**0.06**	0.50	**0.80**	0.07
Trans92	**0.60**	**0.60**	0.05	0.20	0.40	**0.04**
Trans93	**0.60**	0.75	**0.04**	0.50	0.75	0.06

Table 7.1: Above: Comparison of news transcript segmentation at level-1 by sticky HDP-HMM, LaDP and GI-BGS. Below: News transcript segmentation at level-2 by LaDP and GI-BGS. Lower value of S1, S2 indicate better segmentation.

$$\textbf{Recall} = \frac{\textbf{\#gold-standard segments aligned to an inferred segment}}{\textbf{\#gold-standard segments}}$$

For level-2, the alignment threshold is set to 500, and at level-1 it is set to 10. As a baseline, we use sticky HDP-HMM [29] and LaDP at level-1, once again using the 136 HDP topics. For level-2, LaDP is the only baseline.

From the 60 news transcripts from which we learnt the 136 topics through HDP, we selected 3 (Trans1, Trans2, Trans3) to report the segmentation. Also, we selected another 3 (Trans91, Trans92, Trans93) from outside the learning set, for which we used the same initial values of Φ. The results are reported in Table 1. It is clear that in terms of all the measures we considered, GI-BGS outperformed both competitors at level-1. At level-2 also, GI-BGS is competitive on the three measures on all the transcripts except Trans1.

7.11 Implementation Details

Data Collection: There are no standard datasets available (to the best of our knowledge) for this task of news transcript segmentation. So we constructed our own dataset. Actual transcripts of TV/radio news are prepared by the respective stations for the newsreaders to read out, or alternatively they can be generated by a listener using voice-to- text converters. Unfortunately, we had no access to either prepared transcripts or a voice-to-text conversion device. So, we simulated the transcripts using articles appearing in various news websites, while preserving the ordering structure observed in TV/radio news broadcasts, and also in the websites themselves. We chose a 30-day period (1st to 30th April, 2012), and chose news articles from 5 news websites. For co-segmentation it is necessary that the transcripts should have plenty of common topics, but news published in different websites may focus on completely

different topics. For this reason we chose similar sources- 5 news websites based in India having similar focus. Also, news broadcast on TV/radio contain only the most important stories, while the news websites contain many more. So we selected from each source only those stories which have the highest number of comments-a good indicator of importance. It turned out that for each selected article, there was at least one more article on the same topic in some other source.

In the first piece of work (LaDP), we did not consider sentences, and looked upon the documents as a sequence of word tokens. In the later work the sentences became more important as we wanted to investigate the effects of grouping. In text documents, sentence breaks can be found easily by full-stops, while in case of speech-to-text conversion it may be possible to identify them using pauses.

Training and Testing: Of the 150 simulated transcripts we created, we used 60 for training (all 30 transcripts from 2 sources) and the remaining 90 for testing (from the remaining 3 sources). In case of the simulated transcripts, the segmentations were made known to the model, at both the category-level and the story-level. For the category-level ($l = 2$), the categories were also labelled, so that in these segments, β_g^2 were set to δ-distributions, spiked at the labelled values. At the level of stories ($l = 1$), we did not use any labels (as these are not known), but during inference all the tokens in the segment were constrained to be assigned the same mixture component (topic ϕ). As discussed earlier, we also wanted to specify some of the topics ϕ_k, for which we split all the training documents using the available category-level segmentation, and ran HDP. The topics learnt this way were used during the inference.

Implementation: The entire system was implemented in Java. In the first part, the data was processed- stopwords and infrequent words identified and removed, the vocabulary constructed and each transcript converted to a sequence of integers (each of which indexes a word in the vocabulary), for both the training and testing transcripts. Then, the training documents are split up according to the segmentations provided for level 2, to form lots of smaller documents. Then HDP was run on these smaller documents, using standard Gibbs Sampling, and the biasing topics obtained, which were to be used in the main inference.

First an initialization was carried out using forward sampling. We started with all the biasing topics, and sampled new ones if needed. Then the main Gibbs Sampling iterations were carried out, using the inference equations already discussed (for either LaDP or GI-BGS). Most latent variables converged to values after a small number of iterations, simplifying the task. In case of LaDP, the estimated category-to-topic and topic-to-word distributions were saved, to be used in the testing phase. In the testing phase, in case of LaDP we tried both individual transcript segmentation as well as co-segmentation with all the testing documents taken together. It was seen that the second approach gave better results. So all the results

presented in the tables are for co-segmentation. Finally for LaDP, we computed the perplexity.

7.12 Applications, Limitations and Extensions

The aim of this chapter is multi-level clustering of sequential data, and one immediate application of this is multi-level segmentation, which we have already explored in some detail. Apart from news transcripts, these models can also be used for discourse analysis. Clustering or segmentation of more complex documents, possibly with timestamps (so that the sequential structure is present even at the topmost layer) can be considered.

One issue with the experiments we have demonstrated here is that the datasets are relatively small by contemporary standard. This happenned because we had to construct our own dataset. It is of interest to create larger datasets for this task, and re-run the experiments.

Chapter 8

Bayesian modeling of Confusion Matrices for User Opinion Reconciliation

8.1 About this Chapter

In the final chapter of this thesis, we consider confusion matrices, where each column is a probability distribution, and they are related in various ways. We again propose appropriate Bayesian models. We show one application- inferring the correct answers to questions using the opinions (guesses) by users, whose expertise are modeled with confusion matrices. This work is relevant to the general problem of aggregation of user opinions in various online platforms for voting, ranking, rating or answering questions.

Publication: The work on Bayesian models for inferring correct answers using users' opinion has been published in the Conference of Information and Knowledge Management (CIKM), 2013 held in San Francisco, USA.

1. Adway Mitra, Srujana Merugu. *Reconciliation of categorical opinions from multiple sources.* CIKM 2013

8.2 Modeling Distribution Matrices

We now discuss a special type of matrix, namely the *confusion matrix*. Here, every column is a discrete probability distribution. In case of a $D \times K$ matrix, there are K columns, each of which is a point on the D-dimensional simplex. Every entry in this matrix is non-negative, and each column must sum to 1. Such a matrix can be low-rank when there are sets of identical

columns, or when each column can be expressed as a *convex* combination of other columns (unconstrained linear combination will not work here). We consider generative models for this kind of matrices.

We consider Dirichlet base distribution, just as we had done for modeling topics in Chapter 6. We consider the following variants of the Distribution Matrix:

All columns unrelated: Here, all the columns are considered to be generated from Dirichlet distributions with different parameters.

$$\theta_i \sim Dir(\alpha_i) \forall i \in \{1, \ldots, K\} \tag{8.1}$$

Here, each α_i is a K-dimensional vector.

Sets of related columns: Here, each column (say i) is assigned a group g_i, and for each group there is a separate Dirichlet parameter vector.

$$\theta_i \sim Dir(\alpha_{g_i}) \forall i \in \{1, \ldots, K\} \tag{8.2}$$

Each column a convex combination: Finally we consider the case where there are a certain number of basis distributions u, and each column is a convex combination of these. The convex combination components are also generated from a Dirichlet distribution. The basis distributions may have been generated from a single Dirichlet Distribution H.

$$\begin{aligned} u_k &\sim H \forall k \\ \beta_i &\sim Dir(\alpha) \forall i \in \{1, \ldots, K\} \\ \theta_i &= \sum_k \beta_{ik} u_k \forall i \in \{1, \ldots, K\} \end{aligned} \tag{8.3}$$

Observations from such models will usually be normalized count vectors, i.e. relative frequencies. Corresponding to each distribution, we will get a relative frequency vector (which also lies in the same simplex), and we will have a relative frequency matrix F. By Central Limit Theorem, each of the column vectors f_i follows a multivariate Gaussian distribution with mean θ_i.

The second type of matrices considered above are **near-low-rank**, i.e. they may be approximated with a low-rank matrix with sets of repeated columns. The columns which are sampled from the same Dirichlet distribution should be equal to the expectation in the approximate matrix. The third type of matrices will obviously have low rank. The rank will be equal to the number of basis distributions. We consider the low-rank matrix recovery problem from

the observed relative frequency matrices, both in presence and absence of missing values. We can try existing low-rank matrix recovery techniques. The other option is by Gibbs Sampling inference of the above generative processes.

8.3 Opinion Reconciliation from Multiple Sources

Next, we consider an application of the distribution matrices discussed above. Specifically, we look into *confusion distributions*, which involves several discrete values, one of which is correct and has high probability of being selected, but the others may be selected with lower probabilities. The application is to find the correct answers (categorical-valued) to different questions, from several categorical-valued opinions provided by users. This application is very relevant to online Q & A forums (e.g., Quora), prediction markets (e.g., Intrade), online diagnostic systems(e.g., interactiveMD), wiki-compilations (e.g., Wikipedia, Wikimapia). Harnessing "wisdom of the crowd" requires an effective solution for integrating sparse opinions from multiple unreliable, and possibly malicious sources.

In a collaborative information system, there are multiple sources offering opinions on various subjects or questions of interest. Unlike a subjective question like "who is the best US president ever?", there is a unique correct answer for an objective question like "who won the 2012 US Presidential election?". In such scenarios, the goal of opinion integration is to determine this correct answer. We focus on such *Opinion Reconciliation (OR)*, where each question is associated with a unique correct answer.

Let $\{U_1, \cdots, U_i, \cdots, U_{N_u}\}$ and $\{S_1, \cdots, S_j, \cdots, S_{N_s}\}$ denote the multiple sources and subjects (questions) in the information system respectively. Each question S_j is associated with a single correct answer M_j and one or more opinions $\{O_{ij}\}_{i,j}$ from a subset of the sources $\{U_i\}_i$. Often, one might also have access to attributes of sources and subjects, denoted by X_i and Y_j respectively. *The opinion reconciliation (OR) problem can then be stated as follows: Given opinions* $\{O_{ij}\}_{i,j}$*, limited (or even none) observations on the correct answers* $\{M_j\}_j$*, source attributes* $\{X_i\}_i$*, and subject attributes* $\{Y_j\}_j$*, predict the unknown correct answer* M_j*,* $\forall j$*,* $[j]_1^{N_s}$.

There can be multiple variants of the above problem depending on the nature of the opinions and the correct answers, which could be real-valued (e.g., US GDP in dollars), ordinal (e.g., US credit rating), binary (e.g., Is US a monarchy?), categorical from a small known set (e.g., type of governance in US) , text phrases from a given vocabulary (e.g., US national anthem), set-valued (e.g., list of US presidents from New York), etc. Among all the variants, a particularly important formulation is the one where the space of possible answers and opinions for a question S_j comprises of a large number of categorical values or text phrases $\mathcal{O}_j$, which could vary across questions. Applications include information systems with heterogeneous questions permitting

factoid style answers, which is fairly common in prediction markets, specialized Q & A forums, and diagnostic systems. Table 8.3 shows a toy example of such a scenario with multiple human sources, heterogeneous questions on various topics and opinions corresponding to text phrases. Existing approaches to opinion reconciliation can be broadly grouped as :

Axiomatic approaches. These include simple non-data-driven techniques assuming inter-source and inter-subject independence where the correct answer of a question is obtained in terms of various characteristics (mode, mean, median) of the distribution over the corresponding opinions.

Discriminative Meta-learning. These approaches involve learning a functional mapping (e.g., linear model or decision tree) from the opinions and features of subjects and sources to the correct answer through supervision. But this form of supervised learning is not effective when the opinions are very sparse and supervision is highly limited.

Trust Propagation. There is a large body of work on trust propagation over graphs that allow one to estimate the "reputation" (e.g., trustworthiness or correctness) of sources (represented by nodes). Truthfinder algorithm [96] adapts these ideas to OR by considering a bipartite graph over facts (subject-opinion pairs) and sources, with an edge corresponding to a positive assertion on a fact by a source (missing edges are negative assertions). The likelihood of a source making true assertions and the probability of a fact being true are iteratively estimated in terms of each other, but the algorithm is not guaranteed to converge.

Bayesian Models Galland et al. [31] propose a generative model for binary opinions based on source-specific probability of error and not-opinionating as well as subject-specific probability of error and not-opinionating, with the opinion by a source on a subject being determined by the interaction of these parameters. Unfortunately, the algorithms proposed in the paper are based on heuristics not related to the generative model and are not guaranteed to converge. Rayakar et al. [66] and Zhao et al. [100] both propose similar generative models for binary opinions that take into account source-specific probabilities of Type-1 and Type-2 errors, but use different inference approaches. Both works have also been extended to real-valued opinions [99]. A more advanced approach is the Multi-Source Sensing (MSS) model [64], which considers latent groups of sources and models error probabilities as property of each source group.

However, most of the existing techniques are focused on the scenarios where the correct answer M_j and the opinions O_{ij} are binary variables [100, 96, 31]. To some extent, these techniques can be adapted to handle other scenarios involving categorical, textual and set-valued answers by transforming the original subject space to one that permits binary opinions/answers. For example, the question "What is the type of US government?" with possible answers {"democracy", "monarchy", "oligarchy"} can be alternately represented as three questions: {"Is US a democ-

Question	User	Opinion
Materials used in namesake displays	Mr. A	lcd
in computer monitors and HDTVs	Ms. B	liqiud crystal
Answer: (known) liquid crystal	Prof. C	liquid crystal
Category: chemistry	Mr. D	cathode ray
Found by solving $det(A - \lambda I) = 0$	Ms. B	prime
Answer:(unknown) eigenvalues	Mr. F	prime numbers
Category: math	Prof. C	eigenvalues
Architect of the first theory of communism	Ms. B.	Karl Marx
Answer:(unknown) Karl Marx	Prof. C.	Lenin
Category: history	Mr. A	Marx

Table 8.1: Toy example of categorical opinions

racy?", "Is US a monarchy?", "Is US an oligarchy?" }, each of which permits a binary yes/no answer independently. However, these techniques cannot effectively exploit the implicit mutual exclusivity in the categorical valued variables.

This work is an attempt to directly address the opinion reconciliation problem for heterogeneous questions with a large number of categorical opinions which requires taking into account various practical issues shown in Table 1:

Variations in source behavior. In Table 1, we observe that Mr. A is more accurate than other users, indicating that majority vote may not suffice, and source expertise and reliability needs to be accounted for.

Variations in same source's expertise across topics. In the example, Prof. C is an expert in math and chemistry, but ignorant in history pointing to the need for topic-specific modeling of expertise.

Highly limited supervision. The correct answer is often known only for a small subset of questions and one needs to use this supervision to calibrate the expertise of sources.

Opinion Sparsity. Each source provides opinions on only a small subset of questions, which makes it difficult to employ traditional meta-learning techniques.

Textual Variations. Certain opinions are minor variations of the correct answer (eg. typos), which need to be treated differently from an outright wrong answer.

To address some of the above challenges, we propose a Bayesian framework for opinion generation that jointly models the source behavior and subject-specific correct answers as latent variables and make the following contributions:

1) A generic framework for opinion reconciliation via Bayesian modeling that can incorporate latent and observed attributes of sources and subjects. Existing Bayesian approaches such as the LTM [100] can be shown to be special cases.

2) To reconcile categorical-valued opinions, we propose three instantiations (CTM, CTM-OSF, CTM-LSG) of the generic approach to capture the latent source behavior, variations across subject groups, and inter-source correlations.

3) Empirical evaluation of predictive performances of the proposed models and multiple base-

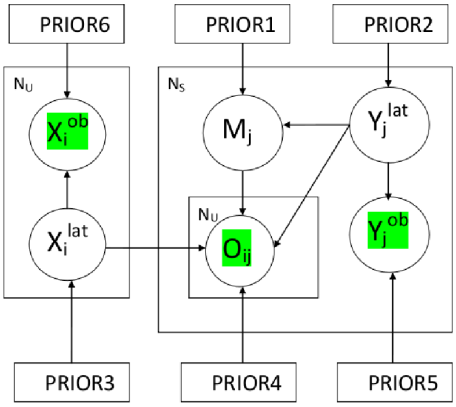


Figure 8.1: Graphical model for Generic Bayesian Opinion Reconciliation

lines on real-world datasets, which points to the relative efficacy of the proposed models.

8.4 Solution Approach by Confusion Distributions

In this section, we describe our generic Bayesian framework for opinion reconciliation, and present three models for categorical opinions. The graphical model encodes two assumptions: (a) Opinions $\{O_{ij}\}_{ij}$ are independent of each other given $\{M_j, X_i, Y_j\}$, (b) Dependencies among sources and subjects are captured entirely in terms of X_i and Y_j.

8.4.1 Generic Bayesian Opinion Reconciliation

The dependencies between the opinions of a source and the correct answer can be succinctly expressed in terms of other variables of interest, such as the source expertise, which are often unobserved or partially observed. An effective solution strategy is to simultaneously infer these latent variables as well as the precise form of the dependencies, in addition to inferring the primary target variable, the correct answer of a subject M_j. A natural mechanism is to encode the dependencies between the different variables of interest (O_{ij}, M_j, X_i, Y_j) in the form of a joint probability distribution that can be factored into conditional distributions amenable for learning. Figure 1 shows a graphical model corresponding to such a factorization. Here X_i^{lat} and X_i^{obs} denote the latent and the observed features of source U_i such that $X_i = [X_i^{lat}, X_i^{obs}]$. Similarly, Y_j^{lat} and Y_j^{obs} denote the latent and observed features of source S_j such that $Y_j = [Y_j^{lat}, Y_j^{obs}]$. The priors allow one to encode domain knowledge as well as data constraints. The process is explained through a graphical model in Fig 8.1.

The above framework provides an elegant way to model some of common factors relevant to opinion generation such as source expertise, source bias, difficulty of a question, inter-source

correlations. Though Figure 1 depicts a specific directionality for the dependence between latent and observed source and subject attributes, e.g., between Y_j^{lat} and Y_j^{obs}, the appropriate directionality depends on which of the conditional probabilities (e.g., $p(Y_j^{lat}|Y_j^{obs})$ or $p(Y_j^{obs}|Y_j^{lat})$) is more learnable given the nature of the variables.

8.4.2 Categorical Truth Model & variants

We now consider the specific scenario where M_j, O_{ij} are both categorical values with support sets $\mathcal{M}$ and $\mathcal{O}$ respectively. We propose three different models, where the conditional probability $p(O_{ij}|M_j, X_i, Y_j)$ can be viewed as *confusion profile* parametrized by X_i and Y_j. This confusion profile is essentially a set of $|\mathcal{M}|$ distributions on the $|\mathcal{O}|$ simplex. In the simple scenario where the variables M_j and O_{ij} are binary, it reduces to the distribution of Type I and Type II errors as done in [100], but can capture more intricate dependencies among categorical variables in general. The three models are described below.

CTM. This model attempts to capture hidden source behavior such as expertise and common mistake patterns in terms of a source-specific confusion profile θ_i, which can be viewed as a latent source-specific feature X_i^{lat}. The priors on the correct answer M_j and each of components in $X_i^{lat} = \theta_i$ are assumed to be Dirichlet-Multinomial and Dirichlet respectively. The generative process is as follows:

$$\begin{aligned} \phi \sim Dir(\beta); \theta_{im} \sim Dir(\alpha_m),\ [i]_1^{N_u}, [m]_1^{|\mathcal{M}|}, \\ M_j \sim \phi; O_{ij} \sim Mult(\theta_{iM_j}),\ [i]_1^{N_u},\ [j]_1^{N_s}. \end{aligned} \tag{8.4}$$

CTM with Observed Subject Features (CTM-OSF). This model attempts to capture the variations in source behavior across observed categories of subjects. In this scenario, each subject is associated with a categorical observed attribute $Y_j^{ob} \in \{1, \cdots, N_{sf}]$ and and each source is associated with $|N_{sf}|$ confusion profiles corresponding to each of the subject categories. The generative process is given by:

$$\begin{aligned} \phi_a \sim Dir(\beta),\ [a]_1^{N_{sf}}, \\ \theta_{iam} \sim Dir(\alpha_m), [i]_1^{N_u},\ [m]_1^{|\mathcal{M}|},\ [a]_1^{N_{sf}}, \\ M_j \sim \phi_{Y_j}; O_{ij} \sim Mult(\theta_{iY_jM_j}),\ [i]_1^{N_u},\ [j]_1^{N_s}. \end{aligned} \tag{8.5}$$

CTM with Latent Source Groups (CTM-LSG). Rather than model the confusion profiles at an individual source level, this model assumes that each source U_i belongs to a hidden group $G_i \in \{1, \cdots, N_{sg}\}$, and associates each group with a confusion profile. The generative process

includes assignment of group indices to the sources.

$$\begin{aligned}\phi \sim Dir(\beta), \theta_{km} \sim Dir(\alpha_m), [k]_1^{N_{sg}}, [m]_1^{|\mathcal{M}|},\\ \psi \sim Dir(\gamma), G_i \sim \psi,\ [i]_1^{N_s},\\ M_j \sim \phi_{Y_j}; O_{ij} \sim Mult(\theta_{G_i,M_j})\ [i]_1^{N_u},\ [j]_1^{N_s}.\end{aligned} \tag{8.6}$$

It is easy to relate these models with the generative processes described in the previous section.

8.4.2.1 Inference

The generative process for CTM, as described above results in the following joint distribution:

$$\begin{aligned}p(O, M, \theta, \phi) &\propto \prod_{k=1}^{|\mathcal{M}|} \phi_k^{\beta_k - 1} \prod_{i=1}^{N_u} \prod_{k=1}^{|\mathcal{M}|} \prod_{l=1}^{|\mathcal{M}|} \theta_{ikl}^{\alpha_{kl}-1} \prod_{j=1}^{N_s} \prod_{k=1}^{K} \phi_k^{\delta(M_j,k)},\\ &\times \prod_{i=1}^{N_u} \prod_{j=1}^{N_s} \prod_{k=1}^{|\mathcal{M}|} \prod_{l=1}^{|\mathcal{M}|} \theta_{ikl}^{\delta(M_j,k)\delta(O_{ij},l)},\\ &\propto \prod_{k=1}^{|\mathcal{M}|} \phi_k^{n_k+\beta_k-1} \prod_{i=1}^{N_u} \prod_{k=1}^{|\mathcal{M}|} \prod_{l=1}^{|\mathcal{M}|} \theta_{ikl}^{m_{ikl}+\alpha_{kl}-1}.\end{aligned} \tag{8.7}$$

Here m_{ikl} is the number of times source i has provided opinion l to a subject whose correct answer is k, and n_k is the number of subjects which have k as the correct answer. On integrating out θ and ϕ, the Gibbs sampling equation is:

$$p(M_j = k | M_{-j}, O) \propto (n_k^{-j} + \beta_k) \prod_{i=1}^{N_u} \prod_{l=1}^{|\mathcal{M}|} (m_{ikl}^{-j} + \alpha_{kl}), \tag{8.8}$$

where n_k^{-j} and m_{ikl}^{-j} are n_k and m_{ikl} respectively, without considering subject j. In practice, while sampling M_j, we restrict ourselves to only those values in $\mathcal{M}$ which have been used in at least one of the opinions available on subject j. The inference steps for CTM-OSF and CTM-LSG are similarly derived, but are skipped here for brevity.

We choose the hyperparameters α and β based on the data. α_{kl} is set proportional to the number of times l is provided as opinion on a subject where the most frequent opinion is k. We set β_k to be proportional to the number of times k is provided as opinion. In the presence of supervision, β_k is boosted proportional to the number of times it occurs as the correct answer for the supervised subjects.

Figure 8.2: Graphical model for generic Bayesian opinion generation. Observed variables are marked in green.

8.5 Empirical Evaluation

In this section, we present empirical results comparing the performance of the proposed models (CTM, CTM-OSF and CTM-LSG) relative to the state-of-the-art methods on real-world datasets in supervised and unsupervised settings.

8.5.1 Experimental Set-up

Datasets: We consider two datasets comprising of categorical-valued opinions: (a) *Quizmaster Dataset* [10], which contains questions on 11 different topics (physics, chemistry, history, literature, etc.) and the *Hubdub Dataset* [31] which contains questions pertaining to the outcome (winner, victory margin) of upcoming sports matches. For both the datasets, each question has a single correct answer, and the users (sources) attempt a variable number of questions. In the Quizmaster dataset, opinions also have typos and linguistic issues, which was addressed via string-matching and normalization as a preprocessing step though in principle, it could be incorporated into the confusion profile. Table 2 provides the details of the datasets. To evaluate the techniques in the presence of supervision, we also created a subset of the QuizMaster dataset (Quizmaster2), where we randomly select 80% of the subjects and their associated opinions. The remaining 20% is kept aside for supervision.

Algorithms: We consider the following baselines: Voting, TruthFinder (TF) [96], 3-Estimates [31], and LTM [100]. For LTM, TF and 3-Estimates, all opinions are transformed into binary-valued facts (question-opinion pairs), and each fact is assigned a score. The opinion corresponding to the fact with the best score is selected as the predicted correct answer for a subject. We consider two versions of LTM: (a) LTM-1 which corresponds to the original LTM and may infer more than one opinion as the correct answer for a question since each question-opinion pair is considered independently and inferred as true/false, and (b) LTM-2, which explicitly chooses a single fact per subject.

In the presence of supervision, we also consider an additional baseline algorithm *Discriminative* based on discriminative modeling. As with TF [96] and 3-estimate, we construct question-opinion pairs, which can be associated with a binary label of TRUE ("opinion is correct answer for question") or FALSE. We also construct features based on the opinion distribution and learn a generalized linear model.

We compare the newly proposed methods: CTM, CTM-OSF and CTM-LSG, against the

Dataset	#Subjects	#Sources	#Opinions	#Distinct opinions per subject
Quizmaster	6076	458	33243	min 1, max 22
Quizmaster2	4876	447	26841	min 1, max 22
Hubdub	357	447	3051	min 1, max 6

Table 8.2: Details of experimental datasets.

above methods. In case of the quizmaster dataset, the topics for each question (11 topics such as physics, chemistry, history) can be used as subject features (Y_j^{ob}) in CTM-OSF. Since the Hubdub dataset does not have such features, we do not evaluate CTM-OSF on this data.

Metrics: All the algorithms except LTM1 output one answer for each subject. As performance metrics, we evaluate the number of predicted answers that match the Ground Truth. We note that, in the Quizmaster dataset, none of the opinions are correct in 395 out of the 6076 questions, and so in these 395 questions the correct answer may never be found, and hence, the maximum number of correct predictions achievable on this dataset is 5681.

8.5.2 Results and Discussions

Unsupervised Setting: Table 3 presents the prediction results of various algorithms on the two datasets (Quizmaster and Hubdub) in the absence of supervision. The values for CTM-LSG correspond to $N_{sg} = 5$, but variations of N_{sg} did not significantly affect the prediction. In case of QuizMaster, the proposed methods are clearly superior to all the baselines while in case of Hubdub, these methods are superior to the Bayesian models, but comparable to Voting and TruthFinder. A possible reason for this is that source-specific confusion profiles can effectively capture the latent interactions in QuizMaster dataset. In case of Hubdub dataset, the representation of the opinions and correct answers (e.g., win by 5 points) may not encode the relevant semantics (Soccer Team A wins over Soccer Team B by 5 points or Hockey Team C wins over Hockey Team D by 5 points) which are not the same from a source perspective. This problem would have been alleviated in CTM-OSF in the presence of observed subject-specific features, which were not available in readily usable form.

Since most of the baselines are primarily meant for binary-valued opinions, we also transformed the categorical opinions to binary facts (subject-opinion pairs) and measured the prediction quality in each case, in terms of Precision and Recall. In case of TruthFinder, we obtained precision-recall values of (0.87, 0.58) while for 3-estimate, we obtained (0.85,0.94), with thresholds chosen so as to maximize the F-measure. For LTM-2 the values were (0.86,0.86). For CTM, CTM-OSF and CTM-LSG, these values are (0.91, 0.91). So it appears that most of the gain is coming from better utilization of the mutual exclusivity between categorical values.

Effects of Supervision: Next, we study the effect of providing limited supervision in the form of correct answers to a few subjects being known. We choose these subjects randomly

Method	QuizMaster	Hubdub
Maximum	(5681)	(357)
Voting	5317	236
TF	5348	239
3-Est	5340	215
LTM1	3846	171
LTM2	5242	158
CTM	5513	239
CTM-OSF	**5523**	-
CTM-LSG	5508	**240**

Table 8.3: Number of correct answers found by different models on Categorical Data.

Method	0%	25%	50%	100%
Voting	4280	4280	4280	4280
TF	3789	4268 ± 1.92	4287 ± 8.44	4319
Discrim	-	4226	4249	4249
3-Est	4253	4248 ± 16.63	4237 ± 31.4	4275
CTM	4429 ± 3.81	4434 ± 3.67	4434 ± 5.08	4437 ± 6.13
CTM-OSF	4433 ± 4.25	4438 ± 4.13	4443 ± 4.47	4449 ± 4.45
CTM-LSG	4427 ± 3.78	4430 ± 3.5	4429 ± 4.29	4429 ± 8.12

Table 8.4: Effects of Supervision on prediction accuracy on Quizmaster2 dataset. Values and standard deviations computed over 10 runs each.

from 20% of Quizmaster dataset kept aside for supervision, and perform the predictions on the test partition (Quizmaster2). We consider 4 levels of supervision- 0%, 25%, 50% and 100% of the training subset. Table 4 shows the results pointing to the superior performance of the proposed models. However, the performance of the proposed methods is relatively invariant to the amount of supervision provided, unlike TF which clearly benefits from supervision.

Chapter 9

Conclusions and Future Work

The first part of this thesis has raised and answered several questions related to modeling temporal coherence in videos at semantic level. We have contributed to semantic modeling of videos in terms of entities, exploiting feature-level TC, modeling semantic-level TC, as well as to various applications of videos. The main conclusions we can draw from the research presented here are as follows:

- A video can be represented concisely and comprehensively as a sequence of tracklets corresponding to entities
- Feature-level TC can be exploited to get effective tracklet representation
- The features need to be chosen according to the applications
- Eigenprofile is a better way of modeling tracklet covariance matrix than Geodesic mean or feature-pooled covariance matrix
- Entity tracking can be carried out in severely challenging illumination conditions by using Eigenprofiles-based tracklet representation with Gabor features
- Bayesian nonparametric clustering of tracklets can be used to discover the entities
- Temporal coherence at semantic level can be modeled using the Bayesian nonparemtric approach, with improved tracklet clustering
- Video summaries can be defined semantically in terms of entities, and created by simple post-processing after entity discovery

- Video scenes and shots can be defined semantically in terms of entities (in case of entity-centric videos like movies and TV-series)
- Scenes and shots in videos can be discovered using Bayesian inference (can also be looked upon as temporal segmentation)
- It is more efficient to simultaneously discover the entities and the scenes, rather than do so separately
- In most matrix-based video representations, the matrix is expected to have sets of identical columns, and hence low rank. But existing low-rank matrix recovery methods are unable to capture such structural information
- Adding regularizers give partly improved results, but Bayesian nonparametric modeling of columns gives a more elegant and effective solution

The next part of the work deals with Bayesian modeling for hierarchically grouped sequential data, and how TC can be used in hierarchical clustering/segmentation of such data. The main conclusions we draw from our research on this matter are:

- Bayesian models for hierarchically grouped data can be systematically compared using the proposed DoS taxonomy
- Bayesian nonparametrics can be used to build models for such data, including temporal coherence at various levels
- Documents like news transcripts can be hierarchically defined using individual stories and broad news categories
- Semi-Markov models for temporal coherence can be more useful than Markov models on some applications like segmentations

This thesis spawns several questions and challenges, which would be interesting to explore in future. The important questions and challenges regarding videos and tracklets that remain unanswered are as follows:

- A dynamic Bayesian model for tracklet covariance matrices, using concepts like Wishart Processes will be interesting

- In entity discovery by TC-CRP and TC-CRF, several clusters are formed per entity. But ideally there should be only one cluster per entity. In the current set-up, attempting to reduce the number of clusters affects the purity of these clusters. It will be interesting to find a way that minimizes the number of clusters per entity while retaining the purity.
- The EntScene model produces a larger number of segments than the number of scenes. This is partly related to the issue of forming multiple clusters per entity, but also due to the fact that scenes in a video have complex structure. It will be interesting to have a model which can capture the scene structure better than EntScene, and therefore produce a more reasonable number of temporal segments.

The part regarding grouped sequential data leaves open the following open directions:

- The DoS-classification scheme reveals several possible models that have been unexplored. It will be interesting to explore these, and see if and where these can be useful.
- Some statistical analysis of learnability of the models based on the concept of sharing components may be useful
- A general-purpose Gibbs sampling algorithm for the Generalized Bayesian Model should be looked into

The work on low-rank matrices, raises the following questions:

- Can we have a generative model for matrix columns based on Dirichlet Process for general low-rank matrices, where the columns are not necessarily repeated?
- The low-rank matrix recovery approaches based on convex optimization provide recovery guarantees. It will be interesting to have some guarantees (like PAC-Bayesian bounds) for the Bayesian approaches also.
- The problem is closely related to subspace clustering, but subspace clustering usually does not consider missing values in the data vectors. It is interesting to see if the proposed DP-based approach can be useful for subspace clustering also.

Bibliography

[1] Amit Adam, Ehud Rivlin, and Ilan Shimshoni. Robust fragments-based tracking using the integral histogram. In *Computer vision and pattern recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 798–805. IEEE, 2006. 24

[2] Tim Althoff, Hyun Oh Song, and Trevor Darrell. Detection bank: an object detection based video representation for multimedia event recognition. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 1065–1068. ACM, 2012. 11

[3] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. People-tracking-by-detection and people-detection-by-tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 23, 24, 25

[4] Ognjen Arandjelovic and Roberto Cipolla. Automatic cast listing in feature-length films with anisotropic manifold space. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1513–1520. IEEE, 2006. 25, 50, 72

[5] S Derin Babacan, Martin Luessi, Rafael Molina, and Aggelos K Katsaggelos. Sparse bayesian methods for low-rank matrix estimation. *Signal Processing, IEEE Transactions on*, 60(8):3964–3977, 2012. 11, 34, 61, 96, 100

[6] Vijay Badrinarayanan, Fabio Galasso, and Roberto Cipolla. Label propagation in video sequences. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3265–3272. IEEE, 2010. 25

[7] Christopher M Bishop et al. *Pattern recognition and machine learning*, volume 4. springer New York, 2006. 104

[8] David M Blei and Peter I Frazier. Distance dependent chinese restaurant processes. *The Journal of Machine Learning Research*, 12:2461–2488, 2011. 31, 59, 60

[9] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003. 27, 104

[10] Jordan Boyd-Graber, Brianna Satinoff, He He, and Hal Daumé III. Besting the quiz master: Crowdsourcing incremental classification games. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1290–1301. Association for Computational Linguistics, 2012. 133

[11] Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010. 33

[12] Emmanuel J Candes and Yaniv Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010. 32

[13] Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009. 32

[14] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011. 11, 32, 61, 94, 96

[15] Yudong Chen, Huan Xu, Constantine Caramanis, and Sujay Sanghavi. Robust matrix completion with corrupted columns. *arXiv preprint arXiv:1102.2254*, 2011. 33, 97

[16] Wei-Chen Chiu and Mario Fritz. Multi-class video co-segmentation with a generative multi-video model. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 321–328. IEEE, 2013. 25, 31

[17] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Kernel-based object tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(5):564–577, 2003. 23

[18] Yang Cong, Junsong Yuan, and Jiebo Luo. Towards scalable summarization of consumer videos via sparse dictionary selection. *Multimedia, IEEE Transactions on*, 14(1):66–75, 2012. 10, 25

[19] Mrinal Das, Suparna Bhattacharya, Chiranjib Bhattacharyya, and Gopinath Kanchi. Subtle topic models and discovering subtly manifested software concerns automatically.

In *Proceedings of The 30th International Conference on Machine Learning*, pages 253–261. 2013. 12, 107

[20] Bruno De Finetti and Bruno de Finetti. Theory of probability, volume i, 1990. 27

[21] Manfred Del Fabro and Laszlo Böszörmenyi. State-of-the-art and future challenges in video scene detection: a survey. *Multimedia systems*, 19(5):427–454, 2013. 10, 26, 78

[22] Persi Diaconis and David Freedman. De finettis generalizations of exchangeability. *Studies in inductive logic and probability*, 2:233–249, 1980. 27, 30

[23] Xinghao Ding, Lihan He, and Lawrence Carin. Bayesian robust principal component analysis. *Image Processing, IEEE Transactions on*, 20(12):3419–3430, 2011. 33, 61, 96

[24] Lan Du, Wray L Buntine, and Mark Johnson. Topic segmentation with a structured topic model. In *HLT-NAACL*, pages 190–200. Citeseer, 2013. 30, 31, 83, 105

[25] Jacob Eisenstein and Regina Barzilay. Bayesian unsupervised topic segmentation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 334–343. Association for Computational Linguistics, 2008. 12, 117

[26] Paul Fearnhead. Exact and efficient bayesian inference for multiple changepoint problems. *Statistics and computing*, 16(2):203–213, 2006. 30

[27] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010. 11, 17, 22, 24, 25, 52, 60, 67, 72

[28] Thomas S Ferguson. A bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230, 1973. 26, 27, 54

[29] Emily B Fox, Erik B Sudderth, Michael I Jordan, and Alan S Willsky. An hdp-hmm for systems with state persistence. In *Proceedings of the 25th international conference on Machine learning*, pages 312–319. ACM, 2008. 30, 59, 61, 81, 86, 104, 121

[30] Emily B Fox, Erik B Sudderth, Michael I Jordan, and Alan S Willsky. An hdp-hmm for systems with state persistence. In *Proceedings of the 25th international conference on Machine learning*, pages 312–319. ACM, 2008. 12, 29, 116

[31] Alban Galland, Serge Abiteboul, Amélie Marian, and Pierre Senellart. Corroborating information from disagreeing views. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 131–140. ACM, 2010. 128, 133

[32] Dilan Görür and Carl Edward Rasmussen. Dirichlet process gaussian mixture models: Choice of the base distribution. *Journal of Computer Science and Technology*, 25(4): 653–664, 2010. 55

[33] Thomas Griffiths and Zoubin Ghahramani. Infinite latent feature models and the indian buffet process. In *Advances in Neural Information Processing Systems*, 2005. 58, 81, 83

[34] Sam Hare, Amir Saffari, and Philip HS Torr. Struck: Structured output tracking with kernels. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 263–270. IEEE, 2011. 24

[35] Timothy M Hospedales, Jian Li, Shaogang Gong, and Tao Xiang. Identifying rare and subtle behaviors: A weakly supervised joint topic model. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(12):2451–2464, 2011. 11

[36] Chang Huang, Bo Wu, and Ramakant Nevatia. Robust object tracking by hierarchical association of detection responses. In *Computer Vision–ECCV 2008*, pages 788–801. Springer, 2008. 24, 25, 50, 51, 78

[37] Jonathan H Huggins and Frank Wood. Infinite structured hidden semi-markov models. *arXiv preprint arXiv:1407.0044*, 2014. 30

[38] Michael C Hughes, Erik B Sudderth, and Emily B Fox. Effective split-merge monte carlo methods for nonparametric models of sequential data. In *Advances in Neural Information Processing Systems*, pages 1295–1303, 2012. 81

[39] Hui Ji, Chaoqiang Liu, Zuowei Shen, and Yuhong Xu. Robust video denoising using low rank matrix completion. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1791–1798. IEEE, 2010. 11, 24, 31, 61

[40] Matthew J Johnson and Alan S Willsky. Bayesian nonparametric hidden semi-markov models. *The Journal of Machine Learning Research*, 14(1):673–701, 2013. 30

[41] Jaya Kawale and Daniel Boley. Constrained spectral clustering using l1 regularization. In *SDM*, pages 103–111. SIAM, 2013. 34, 61

[42] Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from a few entries. *Information Theory, IEEE Transactions on*, 56(6):2980–2998, 2010. 33, 61, 100

[43] Dongwoo Kim, Suin Kim, and Alice Oh. Dirichlet process with mixed random measures: a nonparametric topic model for labeled data. *arXiv preprint arXiv:1206.4658*, 2012. 114

[44] Himabindu Lakkaraju, Chiranjib Bhattacharyya, Indrajit Bhattacharya, and Srujana Merugu. Exploiting coherence for the simultaneous discovery of latent facets and associated sentiments. In *SDM*, pages 498–509. SIAM, 2011. 12

[45] Chao Liang, Changsheng Xu, Jian Cheng, and Hanqing Lu. Tvparser: An automatic tv video parsing method. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3377–3384. IEEE, 2011. 25

[46] Ce Liu, Jenny Yuen, Antonio Torralba, Josef Sivic, and William T Freeman. Sift flow: Dense correspondence across different scenes. In *Computer Vision–ECCV 2008*, pages 28–42. Springer, 2008. 23

[47] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE International Conference on*, volume 2, pages 1150–1157, 1999. 22

[48] Zhengdong Lu and T Leen. Penalized probabilistic clustering. *Neural Computation*, 19 (6):1528–1567, 2007. 34

[49] Vijay Mahadevan and Nuno Vasconcelos. Saliency-based discriminant tracking. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1007–1013. IEEE, 2009. 47

[50] D. Mahajan, F. Huang, W. Matusik, R. Ramamoorthi, and P. Belhumeur. Moving gradients: a path-based method for plausible image interpolation. *ACM Transactions on Graphics (TOG)*, 28(3), 2009. 24

[51] Raghu Meka, Prateek Jain, and Inderjit S Dhillon. Matrix completion from power-law distributed samples. In *Advances in neural information processing systems*, pages 1258–1266, 2009. 31, 33

[52] Stephen Milborrow and Fred Nicolls. Locating facial features with an extended active shape model. In *Computer Vision–ECCV 2008*, pages 504–513. 2008. 22

[53] Adway Mitra, BN Ranganath, and Indrajit Bhattacharya. A layered dirichlet process for hierarchical segmentation of sequential grouped data. In *Machine Learning and Knowledge Discovery in Databases*, pages 465–482. Springer, 2013. 31, 55, 56, 59

[54] Adway Mitra, Soma Biswas, and Chiranjib Bhattacharyya. Temporally coherent bayesian models for entity discovery in videos by tracklet clustering. *arXiv preprint arXiv:1409.6080*, 2014. 83

[55] H. Mobahi, R. Collobert, and J. Weston. Deep learning from temporal coherence in video. In *International Conference on Machine Learning (ICML'09)*. 25

[56] Shakir Mohamed, Katherine Heller, and Zoubin Ghahramani. Bayesian and l1 approaches to sparse unsupervised learning. *arXiv preprint arXiv:1106.1157*, 2011. 97

[57] Feiping Nie, Hua Wang, Xiao Cai, Heng Huang, and Chris Ding. Robust matrix completion via joint schatten p-norm and lp-norm minimization. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 566–574. IEEE, 2012. 33

[58] Yanwei Pang, Yuan Yuan, and Xuelong Li. Gabor-based region covariance matrices for face recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 18 (7):989–993, 2008. 22

[59] Yigang Peng, Arvind Ganesh, John Wright, Wenli Xu, and Yi Ma. Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11):2233–2246, 2012. 22, 32, 72, 73, 94

[60] Dinh Tuan Pham. Joint approximate diagonalization of positive definite hermitian matrices. *SIAM Journal on Matrix Analysis and Applications*, 22(4):1136–1152, 2001. 37, 40, 42

[61] Fatih Porikli, Oncel Tuzel, and Peter Meer. Covariance tracking using model update based on lie algebra. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 728–735. IEEE, 2006. 23, 36, 40

[62] Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid. Category-specific video summarization. In *Computer Vision–ECCV 2014*, pages 540–555. Springer, 2014. 26, 78

[63] Yael Pritch, Alex Rav-Acha, and Shmuel Peleg. Nonchronological video synopsis and indexing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(11): 1971–1984, 2008. 26

[64] Guo-Jun Qi, Charu Aggarwal, Pierre Moulin, and Thomas Huang. Learning from collective intelligence in groups. *arXiv preprint arXiv:1210.0954*, 2012. 128

[65] D. Ramanan and D. A. Forsyth. Using temporal coherence to build models of animals. In *International Conference on Computer Vision (ICCV'2003)*. 11, 24

[66] Vikas C Raykar and Shipeng Yu. Ranking annotators for crowdsourced labeling tasks. In *Advances in neural information processing systems*, pages 1809–1817, 2011. 128

[67] Abel Rodriguez, David B Dunson, and Alan E Gelfand. The nested dirichlet process. *Journal of the American Statistical Association*, 103(483), 2008. 28

[68] David A Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1-3):125–141, 2008. 24, 42

[69] Jitao Sang and Changsheng Xu. Character-based movie summarization. In *Proceedings of the international conference on Multimedia*, pages 855–858. ACM, 2010. 10, 26, 68, 78

[70] Jayaram Sethuraman. A constructive definition of dirichlet priors. Technical report, DTIC Document, 1991. 54

[71] Pramod Sharma and Ram Nevatia. Efficient detector adaptation for object detection in a video. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3254–3261. IEEE, 2013. 22, 64

[72] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000. 98

[73] Josef Sivic, Frederik Schaffalitzky, and Andrew Zisserman. Object level grouping for video shots. In *Computer Vision-ECCV 2004*, pages 85–98. Springer, 2004. 24

[74] Kevin Tang, Vignesh Ramanathan, Li Fei-Fei, and Daphne Koller. Shifting weights: Adapting object detectors from image to video. In *Advances in Neural Information Processing Systems*, pages 638–646, 2012. 22, 64

[75] Makarand Tapaswi, M Bauml, and Rainer Stiefelhagen. knock! knock! who is it? probabilistic person identification in tv-series. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2658–2665. IEEE, 2012. 25, 50, 72, 73

[76] Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Hierarchical dirichlet processes. *Journal of the american statistical association*, 101(476), 2006. 12, 28, 57, 59, 81, 105, 115

[77] Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005. 97

[78] Adrien Todeschini, François Caron, and Marie Chavent. Probabilistic low-rank matrix completion with adaptive spectral regularization algorithms. In *Advances in Neural Information Processing Systems*, pages 845–853, 2013. 34

[79] Oncel Tuzel, Fatih Porikli, and Peter Meer. Region covariance: A fast descriptor for detection and classification. In *Computer Vision–ECCV 2006*, pages 589–600. Springer, 2006. 22, 23

[80] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001. 8, 17, 22, 52, 64, 72

[81] Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl, et al. Constrained k-means clustering with background knowledge. In *ICML*, volume 1, pages 577–584, 2001. 34

[82] Xiang Wang, Buyue Qian, and Ian Davidson. On constrained spectral clustering and its applications. *Data Mining and Knowledge Discovery*, 28(1):1–30, 2014. 34

[83] Y. Wang, H. Fu, O. Sorkine, T-Y Lee, and H-P Seidel. Motion-aware temporal coherence for video resizing. In *ACM Transactions on Graphics (TOG)*, volume 28, page 127, 2009. 24

[84] Yair Weiss. Deriving intrinsic images from image sequences. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 68–75. IEEE, 2001. 36

[85] Sinead Williamson, Chong Wang, Katherine Heller, and David Blei. The ibp compound dirichlet process and its application to focused topic modeling. In *Proceedings of The 27th International Conference on Machine Learning*, 2010. 10, 12, 58, 60, 81, 83

[86] Alan S Willsky, Erik B Sudderth, Michael I Jordan, and Emily B Fox. Sharing features among dynamical systems with beta processes. In *Advances in Neural Information Processing Systems*, pages 549–557, 2009. 31

[87] John Wright, Allen Y Yang, Arvind Ganesh, Shankar S Sastry, and Yi Ma. Robust face recognition via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(2):210–227, 2009. 22

[88] Baoyuan Wu, Siwei Lyu, Bao-Gang Hu, and Qiang Ji. Simultaneous clustering and tracklet linking for multi-face tracking in videos. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2856–2863. IEEE, 2013. 25, 50, 60, 72

[89] Baoyuan Wu, Yifan Zhang, Bao-Gang Hu, and Qiang Ji. Constrained clustering and its application to face clustering in videos. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3507–3514. IEEE, 2013. 11, 25, 50, 53, 60

[90] Yi Wu, Jian Cheng, Jinqiao Wang, and Hanqing Lu. Real-time visual tracking via incremental covariance tensor learning. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1631–1638. IEEE, 2009. 23, 36, 40

[91] Drausin Wulsin, Shane Jensen, and Brian Litt. A hierarchical dirichlet process model with multiple levels of clustering for human eeg seizure modeling. *arXiv preprint arXiv:1206.4616*, 2012. 12, 28, 105, 114

[92] Shijie Xiao, Mingkui Tan, and Dong Xu. Weighted block-sparse low rank representation for face clustering in videos. In *Computer Vision–ECCV 2014*, pages 123–138. Springer, 2014. 25, 53, 61, 72

[93] Xiang Xuan and Kevin Murphy. Modeling changing dependency structure in multivariate time series. In *Proceedings of the 24th international conference on Machine learning*, pages 1055–1062. ACM, 2007. 30

[94] Bo Yang and Ram Nevatia. An online learned crf model for multi-target tracking. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2034–2041. IEEE, 2012. 25

[95] T-C Yen, C-M Tsai, and C-W Lin. Maintaining temporal coherence in video retargeting using mosaic-guided scaling. *IEEE Transactions on Image Processing*, 20(8):2339–2351, 2011. 24

[96] Xiaoxin Yin, Jiawei Han, and Philip S Yu. Truth discovery with multiple conflicting information providers on the web. *Knowledge and Data Engineering, IEEE Transactions on*, 20(6):796–808, 2008. 128, 133

[97] Shun-Zheng Yu. Hidden semi-markov models. *Artificial Intelligence*, 174(2):215–243, 2010. 30

[98] Yi-Fan Zhang, Changsheng Xu, Hanqing Lu, and Yeh-Min Huang. Character identification in feature-length films using global face-name matching. *Multimedia, IEEE Transactions on*, 11(7):1276–1288, 2009. 25, 50

[99] Bo Zhao and Jiawei Han. A probabilistic model for estimating real-valued truth from conflicting sources. *Proc. of QDB*, 2012. 128

[100] Bo Zhao, Benjamin IP Rubinstein, Jim Gemmell, and Jiawei Han. A bayesian approach to discovering truth from conflicting sources for data integration. *Proceedings of the VLDB Endowment*, 5(6):550–561, 2012. 128, 129, 131, 133