Sequential Topic Models for Mining Recurrent Activities and their Relationships : Application to long term video recordings

THÈSE N^o 5469 (2012)

PRÉSENTÉE LE 09 JULY 2012
À LA FACULTÉ SCIENCES ET TECHNIQUES DE L'INGÉNIEUR
LABORATOIRE DE L'IDIAP
PROGRAMME DOCTORAL EN GÉNIE ÉLECTRIQUE
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES
PAR

Jaganndan Varadarajan

acceptée sur proposition du jury :

Prof. Pascal Fua, président du jury
Dr. Jean-Marc Odobez, directeur de thèse
Prof. Shaogang Gong, rapporteur
Prof. Bernt Schiele, rapporteur
Prof. Pascal Frossard, rapporteur

Lausanne, EPFL
2012

Abstract

In this thesis, we address the analysis of activities from long term data logs with an emphasis on video recordings. Starting from simple words from video, we progressively build methods to infer higher level scene semantics. The main strategies used to achieve this are: the use of simple low-level visual features that can be readily extracted, and of probabilistic topic models that come with powerful learning and inference tools.

In the initial part of the thesis, we investigate the use of a simple topic model called *Probabilistic Latent Semantic Analysis (PLSA)* for video scene analysis. By quantizing location, optical flow direction and foreground blob size into words, and considering short video clips as documents, we discover topics from PLSA that represent recurrent activities in the scene. We then demonstrate how the topics can be used to analyze the scene activities, segment the scene into homogeneous activity regions and detect abnormalities.

The topics from PLSA have no temporal structure and hence do not represent activities well. To address this issue, we develop a novel sequential topic model called *Probabilistic Latent Sequential Motifs (PLSM)* which automatically discovers sequential patterns called motifs that include temporal information from videos. To address the problem of observations caused by multiple activities in the scene, the PLSM formulation uses explicit random variables to represent time at different levels: at a higher level to determine when a motif starts in the video, and at a lower level to know the order of words within the motif. Using a sparsity constraint on the event start times, and MAP priors on the temporal axis of the motifs, we designed an inference algorithm. When applied to surveillance videos, the model captures motifs that resemble trajectories. The model provides more information than PLSA, giving clues about when and where an activity starts, when it ends and how it is executed. As in many unsupervised topic models, deciding the most appropriate number of topics is a difficult problem. To address this, we reformulate PLSM using principles of Bayesian non-parametrics. The new method called *Hierarchical Dirichlet Latent Sequential Motifs (HDLSM)* uses Dirichlet processes at multiple levels to select a suitable number of motifs and identify their occurrences in the data.

The final objective is to analyze how events in a scene are organized. At a global level, a scene can be thought of as undergoing a sequence of phases, each with distinct characteristics. At a more local level, the individual activities can exhibit dependencies that are possibly causal in nature. Following this, we propose a new graphical model called *Mixed Event Relationship (MER)* model, that incorporates the learning of both local rules and global states simultaneously from a binary event matrix. Learning these scene semantics is achieved using an iterative Gibbs sampling procedure. While the global scene states recover traffic cycles, the local rules provide information about single and multi-object activity interactions.

We validate the proposed methods with elaborate experiments on nine different challenging datasets with a wide variety of activity content. The results prove the general applicability of the different methods proposed in this thesis. We believe that they can have wider applications on data coming from sensor logs of other modalities too.

Keywords: video, activity, scene segmentation, abnormality, event detection, event relationships, multi-camera, sequential, motifs, pattern recognition, data mining, unsupervised, probabilistic topics models, gibbs sampling PLSA, LDA, PLSM, DP, HDP, HDLSM, MER.

Résumé

Dans ce manuscrit, nous nous intéressons à l'analyse d'activités à partir de longs enregistrements avec un intérêt particulier pour les données vidéo. Partant de simples mots extraits de vidéos, nous proposons des méthodes pour obtenir une compréhension de plus haut niveau de la sémantique de la scène observée. Les principales stratégies utilisées pour cela sont : l'utilisation de descripteurs visuels de bas niveau et la proposition de modèles probabilistes avec les outils d'apprentissage et d'inférence associés.

Dans la première partie de cette thèse, nous abordons l'utilisation d'un *topic model* simple appelé *Probabilistic Latent Semantic Analysis (PLSA)* pour l'analyse de scène vidéo. En quantifiant la localisation, la direction du flux optique et la taille de l'objet de premier plan pour en faire des mots, et en considérant de courts clips vidéos comme des documents, nous découvrons des *topics* à l'aide de PLSA qui représentent les activités récurrentes dans la scène. Nous montrons ensuite comment les *topics* extraits peuvent être utilisés pour analyser les activités présentes dans la scène, segmenter la scène en régions d'activité homogènes et détecter des anomalies.

Les *topics* extraits par PLSA n'ont pas de structure temporelle et ne représentent donc pas efficacement les activités. Pour traiter ce problème, nous développons un nouveau *topic model* séquentiel appelé *Probabilistic Latent Sequential Motifs (PLSM)* qui permet de découvrir de manière automatique des *topics* séquentiels appelés motifs qui capturent l'information temporelle des vidéos. Pour faire face au problème du mélange des observations issues d'objets différents de la scène, la formulation du modèle PLSM fait appel à des variables aléatoires explicites pour représenter le temps à plusieurs échelles : à un haut niveau pour déterminer le début d'un motif et à un niveau plus bas pour expliciter l'ordre d'apparition des observations dans le motif. En utilisant une contrainte d'éparsité sur les instants de début des événements et des *a priori* sur la dimension temporelle des motifs, nous proposons un algorithme d'inférence. Appliqué à des scènes vidéos, le modèle capture des motifs assimilables à des trajectoires. Le modèle fournit plus d'information que PLSA, donnant une information sur le début et la fin des activités et sur la façon dont elles sont exécutées. Comme pour de nombreux *topic models* non supervisés, il est difficile de décider du nombre approprié de motifs à retenir. Pour cela, nous reformulons le modèle PLSM en utilisant les principes bayésiens non paramétriques. La nouvelle méthode appelée *Hierarchical Dirichlet Latent Sequential Motifs (HDLSM)* utilise des processus de Dirichlet à plusieurs niveaux pour sélectionner le nombre adéquat de motifs et identifier leurs occurrences dans les données.

Notre objectif final est d'analyser comment les événements sont organisés dans une scène. À un niveau global, une scène peut être vue comme une séquence de phases, chaque phase ayant des caractéristiques propres. Â un niveau plus local, les activités individuelles peuvent exhiber des dépendances qui peuvent łtre de nature causale. En conséquence, nous proposons un nouveau modèle graphique appelé *Mixed Event Relationship (MER)* qui comprend l'apprentissage simultané de règles locales et d'états globaux à partir d'une matrice binaire d'événements. L'apprentissage de la sémantique de la scène se fait à l'aide d'une procédure d'échantillonnage de Gibbs itérative. Alors que les états globaux correspondent aux cycles de traffic, les règles locales fournissent de

l'information à propos d'interactions entre activités impliquant possiblement plusieurs objets.

Nous validons les méthodes proposées par des expériences sur neuf jeux de données complexes contenant une grande variété d'activités. Les résultats prouvent l'applicabilité générale des différentes méthodes proposées dans ce manuscrit. Nous pensons qu'elles peuvent avoir des champs d'applications plus larges sur des données issues d'enregistrements de capteurs d'autres modalités.

Mots-clés : vidéo, activité, segmentation de scène, anomalies, détection d'événement, relations entre événements, multi-caméras, séquentiel, motifs, reconnaissance de motifs, fouille de données, non supervisée, *topic models* probabilistes, échantillonnage de Gibbs, PLSA, LDA, PLSM, DP, HDP, HDLSM, MER.

The Chiseled Child

A shapeless rock in the hands of an adept sculptor becomes a great peace of art. So too is life; in this untiring journey towards perfection, we meet sculptors great and wise, who shape and chisel our thoughts, career and destination. At this point in time, when I look back with a sense of nostalgia, I see all those sculptors who by their chisels of thought, word and deed have shaped me so far. On such occasions, there are some feelings that can never find their way to expressions, and one such is this feeling of gratitude. I pause here, with uncertainty as to whether to sully this deep sense of thanks, by wrapping it in plain words. Yet, I would be failing in my duty if I do not express the same.

Foremost, I express my deepest sense of gratitude to my guide Jean-Marc, for his immense patience, encouragement and guidance. I always admired his analytical skills, uncompromising attention to details and grasp of the subject. Indeed, I could not have asked for a better supervisor. Merci beacoup Jean-Marc! Besides, I am deeply grateful for the invaluable help and support from Rémi. From simple little software tools to involved technical discussions he has helped in many ways to progress in my work. Merci beacoup Rémi!

I would like to thank Prof. Pascal Fua, Prof. Shaogang Gong, Prof. Bernt Schiele and Prof. Pascal Frossard for agreeing to be part of my defense committee. I appreciate their kind gesture, and I am thankful to them for taking time out of their busy schedule to review my thesis and provide valuable comments.

There is a popular adage which says “You grow when you surround yourself with smart people”. Being in Idiap, one certainly grows, and I am fortunate enough to be around such a *smart environment*. Firstly, my thanks to all the past and present members of Jean-Marc’s group; I thank Carl, CC, Dinesh, Elie, Elisa, Kenneth, Paul, Sileye, Samira, Stefan and Vasil for the many interesting discussions during the team meetings and reading groups. I would like to thankfully mention the help extended by Alex and Romain in proof reading my thesis. Their prompt comments and suggestions were very useful.

Any one traveling from a crowded Indian city to scenic Martigny in Switzerland would know what makes Martigny quite distinct. The lofty mountains, the gusty winds, the scenic valley are certainly the most awe-inspiring. But the sparse population, the sound of silence and the breeze of loneliness are certainly conspicuous and greets every visitor alike. In the last four years, many have helped me brave this lonely journey through their camaraderie. My thanks and kudos goes to Anindo, Anirudh, Ashtosh, Cosmin, Deepu, Gokul, Hari, Harsha, Hugo, Lakshmi, Laurent, Leo, Marco, Mathew, Nik, Radu, Ramya, Roger, Tatiana and Venke. I have enjoyed all those tiring hikes in the breath-taking Alps, competitive badminton shots, fitness-freaking jogging rounds and swimming lessons, funny karoke sessions, spicy barbeques and Indian dinners, intense debates and stimulating discussions and many more such fun-filled activities with them. It has made my stay in Martigny a memorable one and I certainly consider myself fortunate to have known them all.

My life in Martigny was made so comfortable, thanks to the arrangements by Idiap, especially Nadine and Sylvie for providing all the help and support. Thanks also to Corinne, Chantal, Vanessa and their team for all the help and support during the course of my PhD. A special thanks to the Idiap system team for their timely support in all system related matters. I also would like to acknowledge my funding sources: SNSF (HAI 198) and VANAHEIM who generously provided the financial support for my research work and travel.

My sincere gratitude to my teachers at SSSU Prof. G.V Prabhakar Rao, Prof. David Gries, Prof. Panchanathan, Prof. Ashok Srinivasan, Mr. Shakthi Kapoor, Dr. Raghunath Sarma and my seniors at work, Dr. Anji, Dr. Kalika, Dr. Sriganesh, Dr. Sitaram, Dr. Babu and Dr. Srikanth. Their encouragement and support has helped me come a long way. I also gratefully remember my seniors T. R. Kumar and Vineeth for their encouragement and help during my Ph.D application process.

My love and regards to my parents and my sweet brother. They have been a source of immense moral support during trials and tribulations. Be it not for their love and affection, nothing would have been possible.

Finally, my revered obeisance to my Divine Masters - Bhagawan Sri Sathya Sai Baba and Kanchi Sri Mahaperiyava. I owe my everything to them and dedicate this thesis as my humble offering.

Contents

Glossary and acronyms

PTM	Probabilisitic Topic Models
PLSA	Probabilistic Latent Semantic Analysis
LDA	Latent Dirichlet Allocation
PLSM	Probabilistic Latent Sequential Motifs
DP	Dirichlet Process
HDP	Hierarchical Dirichlet Processes
HDLSM	Hierarchical Dirichlet Latent Sequential Motifs
MER	Mixed Event Relationship Model
DBN	Dynamic Bayesian Networks
BIC	Bayesian Information Criteria
GMM	Gaussian Mixture Model
MCMC	Markov Chain Monte Carlo

Chapter 1

Introduction

Inventions are the crest jewels of human intelligence and intuition. One such invention that has achieved an exalted position in human history is that of computers. While initially created for high speed number crunching, it has progressively invaded human lives, assisting and supplementing humans in their day to day chores. Ever since Prof. John McCarthy coined the term *Artificial Intelligence*, researchers have been attempting to make intelligent machines *i.e.,* computers that speak, understand, see and in general, perceive the world through their sensors just as humans use their senses. This is easier said than done due to one important reason among others. We, through evolution have acquired an astounding ability to process complex and abstract patterns to the extent that we fail to recognize them as complex tasks.

For example, we perceive a number of actions like walking, running, eating, drinking, etc., everyday as part of our daily routine. If we look carefully, each of them involves complex movements of the limbs, with possibly millions of variations due to time, place, circumstances and more importantly due to the individual's anatomy. Nevertheless, we recognize and understand such complex patterns quite accurately. Surprisingly, we even pick an individual's idiosyncrasies in gait and personal habits. Not to mention, the complexities involved due to view point variations and the presence of other activities in the background.

In another example, consider that a person is asked to look at a public scene like a metro station ticketing hall for a few minutes and asked to present his understanding of the scene. He or she would most likely come up with a summary that looks like this: "people enter the hall from the south-west entrance, leave through the north-east exit. People often consult the map in the northern side wall, before buying the ticket. On average, there is a crowd flux in the station once every five minutes due to arrival of a train, and so on". Such an analysis, though apparently trivial, is achievable because of the innate ability of the human visual system to process low-level visual information at an extremely high speed, and pass selective information to the brain to derive abstract semantic descriptions of the observations.

We are entering an era of pervasive computing. More and more private and public settings are equipped with sensors including CCTV cameras, generating tones of data everyday. It is therefore

vital to create intelligent machines that can mimic human abilities; machines that can observe colossal amounts of data and churn out information with semantic significance and human interpretability.

In this thesis we target one such problem of designing automatic methods to discover latent knowledge from data. Before going into details, we will take a detour to convince ourselves as to why data mining problems in general are important, and list some of the open challenges in the domain. Subsequently, details about the specific problem solved in this thesis, the approach adopted and the contributions made will be presented.

1.1 Motivation

Significant development in sensing technologies has resulted in a number of sensors that can detect and record human activities. Today, Global Positioning Systems (GPS), SMS and call logs record our position and networking activities. Passive Infra-Red (PIR) sensors detect our movements in buildings. Consumer electronic devices come with accelerometers and gyroscopes that can record our motion activities. Even while on the web, our browsing activities are recorded by online advertising companies. Understanding and analyzing this vast data pool could reveal important clues about human activities and their interactions, which can be further used to anticipate and detect interesting events automatically. Achieving this is of prime interest to many applications, *e.g.,* customized advertising through reality mining, health care through activity mining and action detection, security and customer analysis through video analytics to name only a few.

Video content analysis, more popularly known as video analytics is one such research domain that uses data from CCTV cameras monitoring private and public settings to discover interesting information of the scene. Activity analysis, specifically using videos has wide reaching applications in many domains. We list some of them here.

Video Surveillance. These days, practically every public setting like train stations, airports, shopping malls, traffic junctions and streets are monitored using surveillance cameras. But much of this content is rarely screened and merely serve as record for forensic analysis. Moreover, searching for a specific occurrence in this enormous quantity of data amounts to looking for a needle in a haystack. A surveillance camera becomes more usable if it is packaged with intelligence to detect events that require attention as they happen and take action in close to real time. This has motivated several corporate bigwigs and Original Equipment Manufacturers (OEM) like IBM, Google, Honeywell, GE security, ObjectVideo, Philips, Sony, etc., to have dedicated research teams to create end-to-end surveillance systems. Several academic research projects like VSAM [1], CARETAKER, VANAHEIM [2] and SAMURAI [3] have also invested time and effort to create intelligent surveillance systems that perform tasks such as tracking, trajectory analysis, crowd monitoring, counting and so on.

1. www.cs.cmu.edu/vsam/
2. www.vanaheim-project.eu/
3. www.samurai-eu.org/

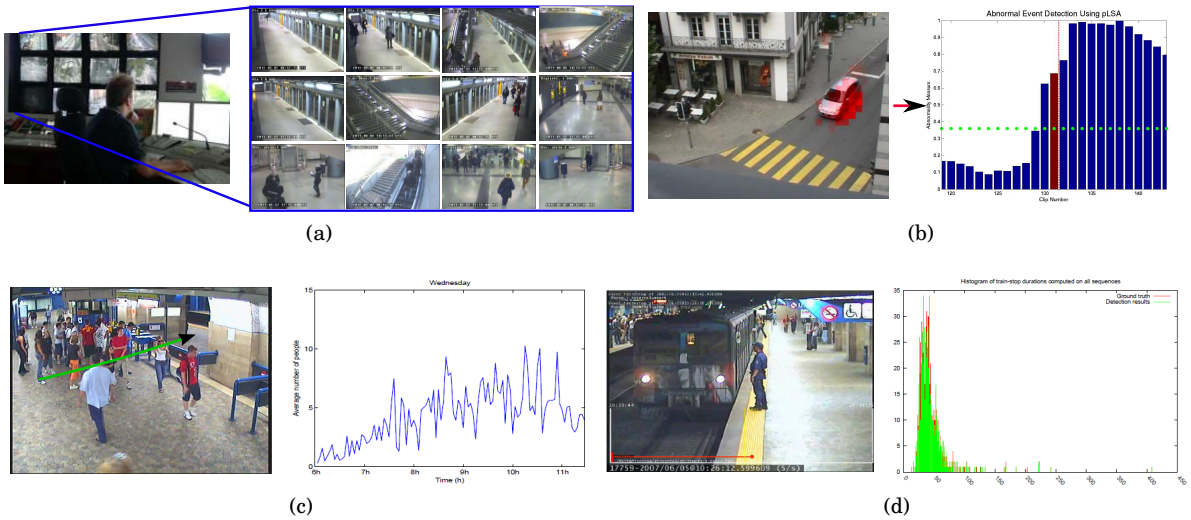


Figure 1.1. Various applications of video activity analysis. (a) A monitor operating multiple cameras from a metro station site in a control room. (b) Abnormal occurrences detected and highlighted. (c–d) Event statistics collected from a metro station, *e.g.,* (c) Average number of people crossing the turnstile during the day shows a peaking trend around 8h30-11h00. (d) Histogram of waiting times of trains in the station.

Automatic stream selection is another application where video analysis can play a crucial role. Nowadays, many of the transport-infrastructure settings like metro stations and airports, monitor different parts of their sites by employing human operators. The videos from different locations of the site are sent to a control room where the operators scan the scenes online (see Figure 1.1(a)) and alert the concerned authorities in case of an unusual event like an accident or property vandalism. But there is a huge issue involved in this mode of working. To quote the New Scientist magazine [4]: "We are having too many cameras and not enough pairs of eyes to watch them all". More precisely, having human operators to scan multiple views at any time has been quite inefficient because of the short attention span that humans can afford. Also, monitoring the same scene repeatedly over time creates tiredness, boredom and results in missing potential unusual events. A way to reduce the strain on the operators is to have an automated procedure that pre-screens the views and passes on only the most interesting views from the site. This gives a two stage screening process (automated and manual) while making efficient use of the operator's attention.

Content based retrieval. In many occasions like criminal investigations, it is necessary to browse through several hours or days of video recordings. This may be done using some prior knowledge on place or time of occurrence of an event. Otherwise, it is a tedious process in the absence of any meta-information about the content of the video. One way to address this problem is to use video pattern mining methods to learn the activity patterns in the video, derive semantic interpretations and natural language descriptions. This can serve as a potent annotation of the data and a tool to browse the video more easily.

4. New Scientist 13 November 2003

Infrastructure planning. End-users of infrastructure projects look for event statistics to help in their planning and management. For example, information about crowd congestion and their duration, queue formation and their average waiting times etc., could provide interesting statistics on events occurring in the scene. Figures 1.1(c–d) show some statistics collected about the average number of people crossing the turnstiles during morning busy hours and average waiting time of trains in a Rome metro station. Similarly, a manager of a large commercial complex may re-arrange and improve the store periodically by observing places where customers spend most of their shopping time.

Pattern mining. More generally, pattern mining from sequential data has much wider implications in domains other than video analysis. Consider for example a meteorology grid set up across a region collecting various atmospheric variables along time. This defines a time series consisting of multiple observations. Mining recurring patterns from such a data can reveal useful information regarding weather patterns and correlations between various atmospheric parameters (Basak *et al.*, 2004). Similarly, looking at latent temporal patterns in cellphone GPS and call logs reveal a variety of daily routines followed by people. In the area of elderly care, sensor information coming from accelerometers worn by people can be analyzed to extract patterns that correspond to routines like shopping, cooking and the like. Such a general view of the activity analysis problem allows us to extend our methods to several other potential applications ranging from meteorology, biology to sensor processing.

1.2 Challenges

Though automatic activity analysis has been receiving considerable research attention over the last few years, little success has been achieved in building intelligent deployable systems due to various challenges:

- **Sensor Placement:** Often we use the same sensors to solve multiple problems. An application that requires robust tracking results for example, should use data from overhead cameras to avoid occlusions. But data from such a view may not be suitable for applications that need human identification, behavior and interaction analysis. Since installing cameras along with the required networking and storage for each application becomes expensive, in practice, a compromise is achieved by many applications sharing the same data.
- **Activity definition:** There is a high degree of ambiguity in the definition of various terms used in activity analysis research. For example, consider the two frequently interchanged terms *Action* and *Activity*. It is unclear as to what in a video is an action and an activity. This ambiguity is more pronounced in the definition of terms such as “normal” and “unusual” actions or activities especially demanded in surveillance applications. Unusual activities are rare, hard to describe, hard to predict and can be subtle. Since this requires a more detailed look, we will revisit this aspect later in this chapter.
- **Data variability:** Every automated surveillance system suffers from acquisition problems.

Video data from such settings have cluttered background, camera motion, frequent occlusions, photometric variations due to lighting and shadows, view point variations, video artifacts due to compression etc.

- **Activity modeling:** The first step in activity analysis is activity modeling. For example, a vision based approach to analyzing a person's behavior would involve continuous tracking of the person, segmenting the person from the background, extracting visual features that are potentially view invariant and finally employing a suitable learning technique. Each of these steps are still challenging problems in computer vision. Furthermore, there are variations in time taken to perform the action, individual styles, physical appearances. Activity modeling often involves modeling of interactions (between people, people and environment) and context, which adds another layer of difficulty to the problem. From a computational perspective too, modeling activities remains a challenge.
- **Data labeling:** Using many of the learning approaches to identify activities especially in a supervised context, requires significant amount of labeling. But human labeling is tedious, time consuming, expensive and error prone. Especially in the case of videos, annotating activities in the presence of multiple other activities happening simultaneously is cumbersome.
- **Adaptation:** Requirements of surveillance systems vary from one user to another. For example, an operator monitoring a traffic scene might be interested in crowd congestion while a train or tube station operator is more interested in vandalism, violence and accidents like people falling on the tracks. Due to such varying needs, techniques developed for each site become exclusively fine-tuned to certain operational conditions with hard-coded rules, resulting in lesser adaptability.

1.3 Terminology

Terms such as "actions", "activity", "behavior" and "events" are often interchanged and not defined clearly in activity analysis research. Looking at Merriam Webster dictionary [5] definitions for the terms say the following: An **action** is *the state or process of doing something or being active*. An **activity** is *a lively action or movement* . Even from these definitions, there is considerable overlap and confusion on what constitutes an action and an activity. We go by the definition provided in (Turaga *et al.*, 2008) where, "action" is used to refer to simple motion patterns usually executed by a single person that typically lasts for a short duration of time in the order of few seconds, *e.g.,* walking, swimming. On the other hand "Activities" may refer to complex sequences of actions performed by one or more humans with a longer temporal extent, *e.g.,* shopping or two persons engaged in a discussion. In this work, since we use unsupervised, co-occurrence based methods to learn patterns, our results can be interpreted as actions sometimes (*e.g.,* a vehicle moving from left to right) and as activities (*e.g.,* a vehicle moving from left to right while people waiting to cross the road) at other times. Therefore, we will consistently use the term activity or activity patterns to refer to the

5. www.merriam-webster.com

learned patterns.

An **event** is defined as *something that takes place; an occurrence*. We will also use the term "events" to indicate any occurrence, start of an activity.

A **behavior** is defined as *the response of an individual, group, or species to its environment.* From the definition of the term "behavior", we understand that the interpretation should consider the environment or the context as well. Furthermore, an individual, a group or even a scene as a whole can exhibit a behavior. For example, an individual's behavior in a traffic controlled junction is different from its behavior in an uncontrolled road. A scene's behavior is the set of activities that occur defined by the scene context such as the number of roads, junctions, traffic lights, their durations and periodicity.

Unusual or **abnormal** events are another set of ambiguous terms. The term unusual is defined as "not habitually or commonly occurring or done", whereas abnormal is defined as "deviating from what is normal or usual, typically in a way that is undesirable". We can understand that these terms refer to the same thing except that the latter carries a strain of negativity with it. Using a data-driven perspective, unusual or abnormal events can be defined as something that cannot be explained using the training data or not seen in the training data.

1.4 Objectives and Approach

This thesis considers the challenging problem of deriving conceptual descriptions from a video input, otherwise called "automatic video activity analysis". The input videos come from complex video streams arriving from public scenes like busy traffic scenes or crowded metro stations. By deriving a high-level human understandable semantics of the scene we would like to address questions like: 1) what are the recurrent or dominant activities in the scene and how many of them occur? 2) are the event occurrences periodic or random? 3) what are the general rules such as right of way, traffic lights and pedestrian preferences observed in the scene? 4) what is the status of an event discovered: does it depend on others or does it occur independently? 5) are there interesting or abnormal event occurrences like vehicles jumping traffic lights?

This thesis presents several novel methods to answer the above questions. The approaches presented stem from the observation that videos are times series data and the activities occurring in the scene are hidden patterns in the time series. Therefore, although primarily motivated by video analysis applications, we also show that the methods proposed here could have a wider application to sensor logs other than videos.

The approach of this thesis is based on three paradigms. They are:

– **Low-level features:** Since our datasets come from crowded scenes with multiple objects of different kinds acting simultaneously, configuring a multi-object tracker individually for each scene becomes computationally expensive. Therefore, we aim to avoid object centric features that come from tracking and resort to simple visual features that include foreground pixels and optical flow.

– **Minimal supervision:** As noted before, labeling huge volumes of video is a tedious and error prone process. A solution to this labelling problem is to employ unsupervised or semi-supervised methods, where a model is learned initially with no supervision, and labels are used at a latter stage to perform detection, classification and evaluation [6]. Such approaches have been successfully applied for several vision tasks such as scene classification (Quelhas *et al.*, 2005) and face detection (Nguyen *et al.*, 2009).
– **Probabilistic Topic models (PTM):** Among the popular unsupervised methods to date, PTMs have been successfully used by the computer vision community for various tasks such as object recognition, human activity recognition (Niebles *et al.*, 2008). They have also been used with a variety of modalities like accelerometers (Huynh *et al.*, 2008), cell phone GPS (Farrahi and Perez, 2008), etc. Additionally since PTMs are essentially graphical models, they enable us to model complex real life phenomena which can be solved using established inference tools.

1.5 Contributions and Thesis Organization

By abstracting out minor details, Figure 1.2 gives a bird's eye view of the contributions and organization of this thesis. The chapters in this thesis are organized chronologically to reflect the order in which contributions were made. The thesis therefore progresses from simple models to more sophisticated ones at the latter chapters. A brief summary of each of the chapters and the contributions is as follows:

Chapter 2. A brief overview of various methods in the literature for activity analysis is presented and the contributions of this thesis are highlighted, contrasting it with the existing works.

Chapter 3. A detailed look at the various datasets used and the procedure followed to extract features are presented.

Chapter 4. As an initial step towards using topic models for activity analysis, we propose to use Probabilistic Latent Semantic Analysis (PLSA) along with simple visual features like location, foreground blob size and optical flow extracted from video to obtain dominant patterns of activities from a busy traffic junction video (Varadarajan and Odobez, 2009). Using the topics learned,

– we perform some analysis to rank the activities with respect to particular semantics, *e.g.,* direction of motion, speed or size of objects;
– a novel activity based scene segmentation is proposed to obtain regions of similar activity content;
– we study various abnormality measures in the given context and propose a novel measure based on document reconstruction error.

Chapter 5. In a topic model framework, the temporal information in the video is normally ignored. To model the word dynamics within an activity, we propose a novel topic model based approach

6. Here, we expect that our model generated in an unsupervised fashion will reduce human effort at consequtive stages.

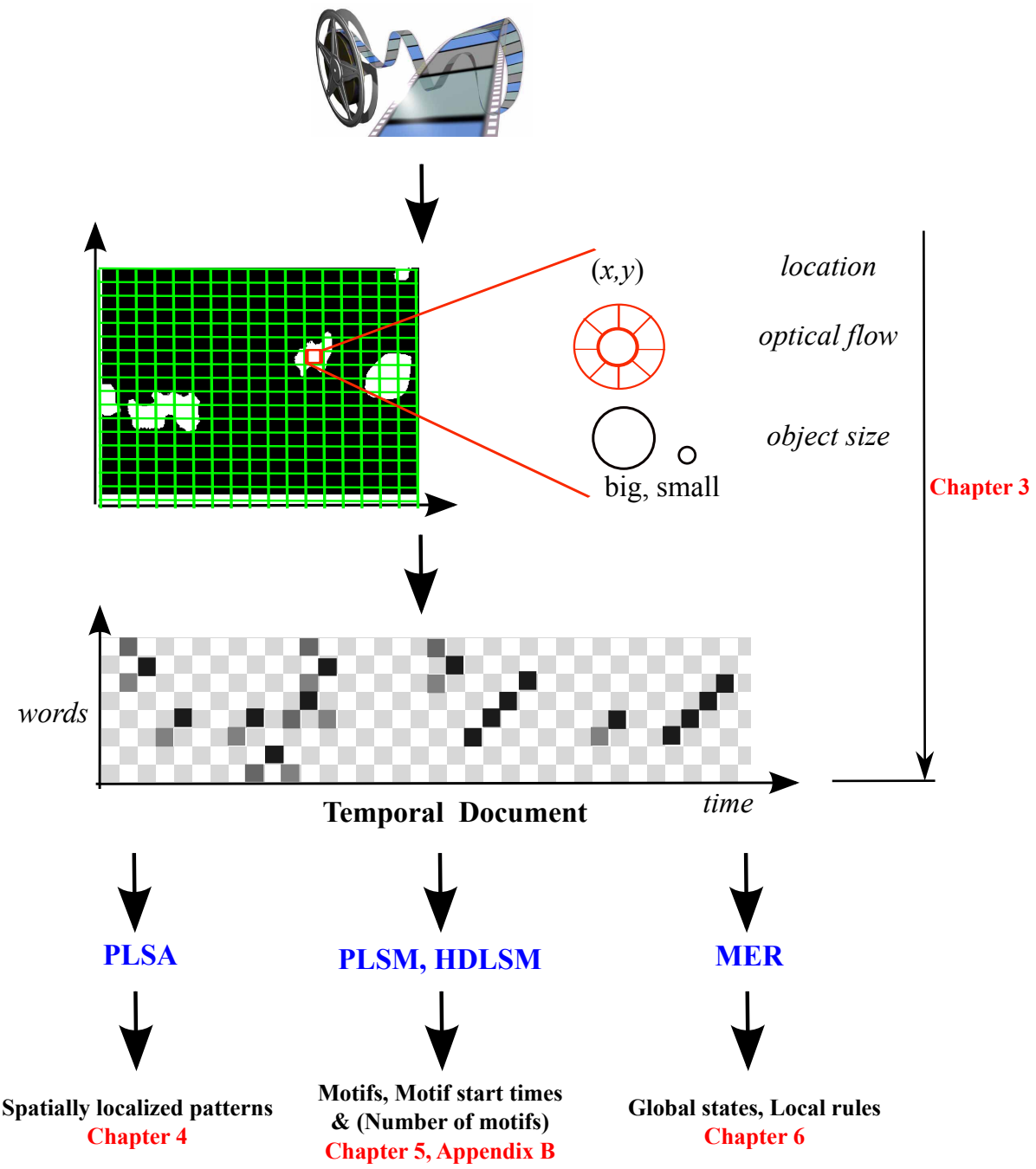


Figure 1.2. A pictorial overview of this thesis. Chapter 3 deals with deriving a video representation, from visual features to discrete word counts In Chapter 4, word counts within overlapping time windows are concatenated, ignoring all temporal information and used with PLSA to discover activities. In chapter 5, PLSM uses the temporal document to discover motifs and their start times. In Appendix B, the number of motifs for a dataset is derived automatically in a non-parametric PLSM like model. In chapter 6, the motif occurrences are used to learn behaviors and global scene states.

called *Probabilistic Latent Sequential Motifs* **(PLSM)** (Varadarajan *et al.*, 2010), for discovering dominant sequential activity patterns called *motifs* from word×time counts or *temporal documents*. The model has the following features:

- motifs not only capture the co-occurrence of words in a temporal window, but also the *temporal order* in which the words occur within this window;
- parameter estimation in this model is formulated by *jointly inferring motifs as well as their starting times*, allowing us to implicitly align the occurrences of the same pattern during learning;
- a regularized EM procedure to encourage *sparse distributions*, is proposed. We see that when documents are corrupted by noise, imposing the sparsity constraint on the motif occurrence distribution improves motif recovery;

Results from exhaustive experiments involving synthetic and real-life datasets of single and multi-camera views (Emonet *et al.*, 2011b), quantitative evaluations on event detection and prediction tasks and experiments on Time Delay of Arrival (TDOA) data prove the effectiveness of the model.

Chapter 6. We propose a novel model called the *Mixed Event Relationship* **(MER)** model (Varadarajan *et al.*, 2012), that takes as input binary event matrix, whose entries indicate the start of a fixed set of short-term temporal activities over time, and outputs both local and global scene level rules. The generative model has the following properties:

- it assumes that events can occur either independently or depending on other events, decided by a binary random variable in the generative model.
- there are global scene states that regulate which of the activities can spontaneously occur;
- there exist local rules that link past activity occurrences to current ones with temporal lags.
- model parameters are efficiently inferred using a collapsed Gibbs sampling scheme.

Experiments on various datasets from the literature show that the model is able to capture temporal processes at multiple scales: global scene level and local inter-activity level.

Chapter 7. We conclude the thesis with a brief summary of the important contributions made and outline the potential directions for future work.

Appendix. In **Appendix A**, the Expectation Maximization equations for PLSM inference are explained. In **Appendix B**, the *Hierarchical Dirichlet Latent Sequential Motifs* model, a non-parametric Bayesian technique based on Dirichlet process to decide on the number of sequential motifs and locate them is presented. In **Appendix C**, Gibbs sampling steps for the MER model are explained. In **Appendix D**, some basic concepts and formulations from Bayesian statistics that are used in this thesis are presented.

Chapter 2

Literature review

In this chapter, we present a brief overview of the main components of various activity analysis methods. A chart of the topics covered in this chapter is presented in Figure 2.1. Due to the high volume of research work done in this domain, we will restrict this overview to methods pertinent to this thesis. Therefore, the main focus of discussion in the rest of this chapter will be on the following subtopics.

- **Video representation**, which deals with how and with which features we represent video content and more particularly activities;
- **Learning approaches**, which deals with how activities are inferred or learned from low-level feature representations;
- **Probabilistic Topic models**, a class of graphical models to obtain meaningful patterns from discrete data using co-occurrence analysis; given its importance in this thesis, we will dwell into more details on how it is being adapted and applied for activity analysis;
- **Evaluation**, which deals with estimating the goodness of a proposed method for tasks under activity analysis.

For more elaborate reviews we refer to survey papers on visual surveillance (Hu *et al.*, 2004), tracking (Yilmaz *et al.*, 2006), action recognition (Turaga *et al.*, 2008) and trajectory based activity analysis (Morris and Trivedi, 2008b).

2.1 Video representation

Let us consider an artist's sketch of a natural scene, a façade of a building or a person. The sketch contains simple outlines of the drawing, but still convey sufficient information to distinguish one from the other. In fact, there is an increasing evidence that the human visual system perceives and interprets all that it sees by first processing low-level visual features like corners and edges (Lowe, 2000; Pasupathy and Connor, 2002) and object boundaries at the lower levels of the visual cortex. Our objective in video representation is to extract such concise representations of the video. Since videos usually come with enormous information, it is computationally efficient and often sufficient

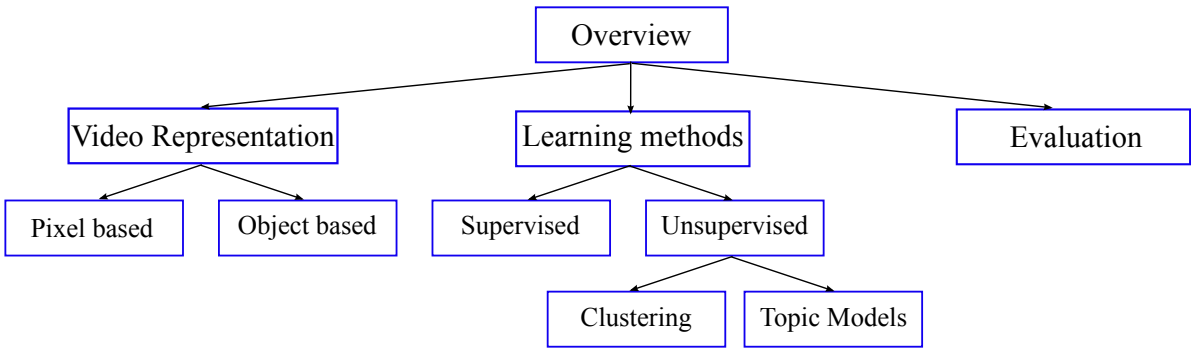


Figure 2.1. A pictorial overview of the topics covered in this review chapter.

to work with such low-level visual features. Thanks to the wide research interest in computer vision over many years, we now have established methods to extract these low-level features.

Video representation methods can be broadly classified into two categories: i) pixel based features and ii) object based features. Pixel based features are obtained by processing every pixel in the image and do not involve any object level semantics. Usually, these methods involve computation of pixel similarities, image derivatives over space and time or applications of mutliple filters. Object centric features on the other hand, work with features that represent a semantic object like a human or his face in the video. They typically involve segmenting individual objects in the frame and establishing their correspondence over time. In the following sections we will briefly review the most popular methods adopted for representing video content.

2.1.1 Background subtraction

Detecting foreground objects and segmenting them from a sequence of frames captured from a static camera is a good clue of activities occurring in a scene. For example, by following foreground blobs over a sequence of frames as in Figure 2.2(e–h), one may easily perceive activities like a car moving or a person standing at a location. Typically, background subtraction methods work by first learning the background of the scene and then classifying each pixel in the frames as either belonging to the background or foreground regions. The extracted foreground blobs are then used to derive cues such as object size, aspect ratio and object silhouettes that are used in further video processing steps. For instance, temporal activity templates were created for action recognition by aggregating foreground blobs obtained from background subtraction in (Bobick and Davis, 2001). Similarly, the ratio of foreground pixels in a region is used to represent the amount of activity in a multi-camera setting with poor video quality in (Loy *et al.*, 2009). Background subtraction also serves as an important pre-processing step for many human detection and tracking tasks (Yao and Odobez, 2008a). Today, we have many reliable background subtraction methods that work in real-time, but they are still affected by illumination variations, shadows cast by moving objects, dynamic backgrounds like waving tree branches, moving escalators, animated advertisements in the background and stationary foreground objects. To address these issues, many adaptive background learning methods were

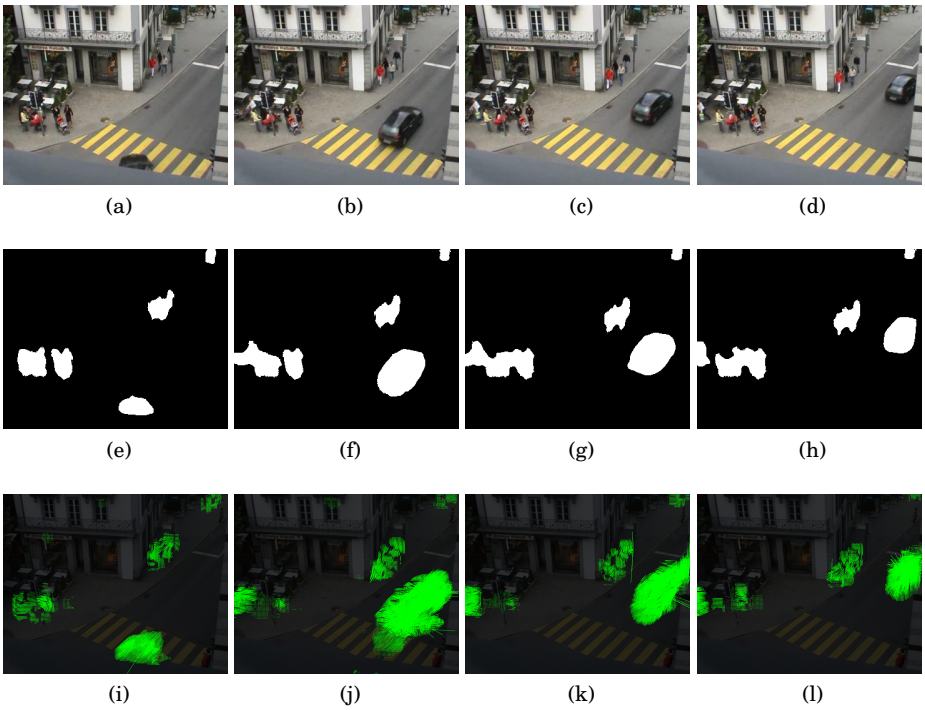


Figure 2.2. Background subtraction from a sequence of frames indicates a car moving from bottom to the right. The white blobs in images (e–h) correspond to foreground regions of the frames in (a–d). Images in (i–l) show optical flow vectors for the corresponding foreground regions

also proposed in the literature (Stauffer and Grimson, 1999; Yao and Odobez, 2007).

2.1.2 Optical flow and motion detection

Most actions are characterized by some specific movement or motion patterns. Therefore, most activity analysis approaches rely heavily on some form of motion detection. A simple method to detect motion is by computing frame differences. A more reliable approach is to compute optical flow features, which is the apparent motion of individual pixels over successive frames. The basic assumption underlying optical flow computation is brightness constancy. According to this, pixel intensities are assumed to remain constant under displacement from one frame to another. One way of getting optical flow is by block matching, where for every block in the first image the algorithm looks for a corresponding block in the second image. Different metrics can be used to measure similarity or difference between blocks: *e.g.,* cross correlation, squared difference or robust error norms. Another class of methods are called differential methods, which include popular algorithms by Horn and Schunk (1981) and Tommasi and Kanade (1991). They differ in the regularization terms used in the computation. While Horn and Schunk (1981) use a global smoothness on the optical flow vectors, Tommasi and Kanade (1991) in their KLT method, obtain the weighted least squared fit of the brightness constancy constraints implemented over multiple resolutions. Working

on muti-resolution pyramids gives a better estimate of the flow especially when objects move faster in the video. See Figure 2.2(i–l) for the optical flow features extracted for the frames in Figure 2.2(a–d) using the KLT method.

Without sophisticated object detection or segmentation, optical flow methods provide a good estimate of the regions undergoing motion as well as their direction. This has been successfully used in many vision tasks like motion segmentation (Odobez and Bouthemy, 1995), point tracking, image registration (Tommasi and Kanade, 1991) and action recognition from a distance by Efros *et al.* (2003). Recently, they have also been used to study higher level activities. For instance in (Hospedales *et al.*, 2009; Kuettel *et al.*, 2010; Wang *et al.*, 2008b), quantized optical flow features were used to derive scene behaviors from videos of urban traffic. In (Andrade *et al.*, 2006; Mehran *et al.*, 2009; Wu *et al.*, 2010), they propose methods to understand crowd behaviors and detect abnormal situations by studying dynamics of optical flow features.

In practice, optical flow vectors are preprocessed before they are put to use. In some cases, vectors with magnitude beyond a certain threshold are selected to filter out spurious flow vectors due to acquisition noise. In cases, where background subtraction is also used, it is more efficient to extract optical flow vectors only on foreground pixels. Though we have many efficient off-the shelf optical flow implementations, they suffer from clutter, low frame rates and relatively large motion in contrast to the object size resulting in poor motion estimates.

2.1.3 Spatio-temporal features

Many human activities involve only certain parts of the body. For instance, a human walking involves distinct movements of the hands and legs and little or no action of the torso. Spatio-temporal features aim to capture such informative subvolumes that provide strong cues about activity in the the video. There are several methods adopted for this purpose in computer vision distinguished by their sparse or dense nature, the characteristics they extract from the subpatches and so on.

One approach to obtain spatio-temporal features is to simply consider dense 3D subvolumes and compute image statistics withing the volumes. Another approach is to use large filter banks such as oriented Gabor filters, Gaussian kernels and their derivatives and extract regions of high response. Extracting sparse space-time interest points instead of 3D volumes is an alternative approach, inspired by 2D image corner detectors like (Harris and Stephens, 1988; Tommasi and Kanade, 1991).

Various features from subvolumes were proposed for activity analysis. One way is to collect statistics of dense space time volumes and match them with template volumes from a training set. For example, Boiman and Irani (2007) used a database of spatio-temporal patches built from normal behaviors and consider abnormal actions in a video as those patterns that cannot be composed by the learned patches from the database. Ke *et al.* (2007) used spatio-temporal shapes from the patches for a similar application. Kim and Grauman (2009) use flow patterns from subvolumes with probabilistic PCA and enforce global consistency constraint using a MRF model. This was applied to detect abnormal events like loitering, skipping payments and zig-zag movements in a metro

station. In (Kratz and Nishino, 2009), gradients within spatio-temporal volumes are modeled using 3D Normal distributions. The spatial relationships among the spatio-temporal patterns are learned using a coupled HMM. This is employed in an anomaly detection task in dense crowd situations. In case of sparse interest points, a K-means clustering is used by (Niebles *et al.*, 2008; Schuldt *et al.*, 2004) to build a code-book and recognize human actions like walking, running and punching.

Apart from this, there are also features based on image appearance like histogram of oriented gradients, motion history images, dynamic textures that have also been used for activity analysis. Appearance-based features usually demand a relatively better quality frame resolution, where object appearances are distinct and clear. Due to this, they are not widely used in surveillance applications where videos generally come with poor resolution and low frame rates.

So far, we reviewed various activity representation methods given by low-level features. Pixel level features are interesting as they can be extracted without any a-priori knowledge about the video content. However, in some pixel based methods, we get rid of the object identity and therefore, a particular object and its activities cannot be separated from others when multiple objects occur in the scene simultaneously.

2.1.4 Object trajectories

An alternative to pixel based approaches is to use object centric features from tracking. Object tracking can be defined as the problem of estimating the location of an object in the 2D image plane or in 3D space as it moves around the scene. This implies that objects appearing in successive frames are linked or associated with consistent labels. Tracking based approaches are preferred for activity analysis since they directly allow us to perform object level analysis. For example, by using a tracking based surveillance system monitoring a parking lot, activities indicating potential theft such as abnormal trajectories of driving or people loitering around cars could be detected. Similarly, trajectories can help us understand how humans behave and interact in a multi-person scene. See Figure 2.3(a) for some examples of human tracking for surveillance and behavior analysis.

Object tracking methods can be categorized based on:

- tracking task: single or multiple objects; physical setup: where and how many sensors are used; scenario: indoor or outdoor and crowd density;
- object representation used for tracking: feature points, contours, silhouettes and bounding box;
- formulation of the tracking problem: optical flow based methods, Kernels, Kalman filters, Particle filters and other dynamic Bayesian networks and detection based trackers;
- features used for association: color histograms, corners, edges, optical flow and texture features.

There is usually a tradeoff between ease in extracting the features and their viability to be tracked well. For example, point features are easy to extract, but tracking over a long period of time is harder due to their poor discriminative power. On the other hand, a human is tougher to detect in a frame while being relatively easier to track continuously due the distinct identity. Tracking mul-

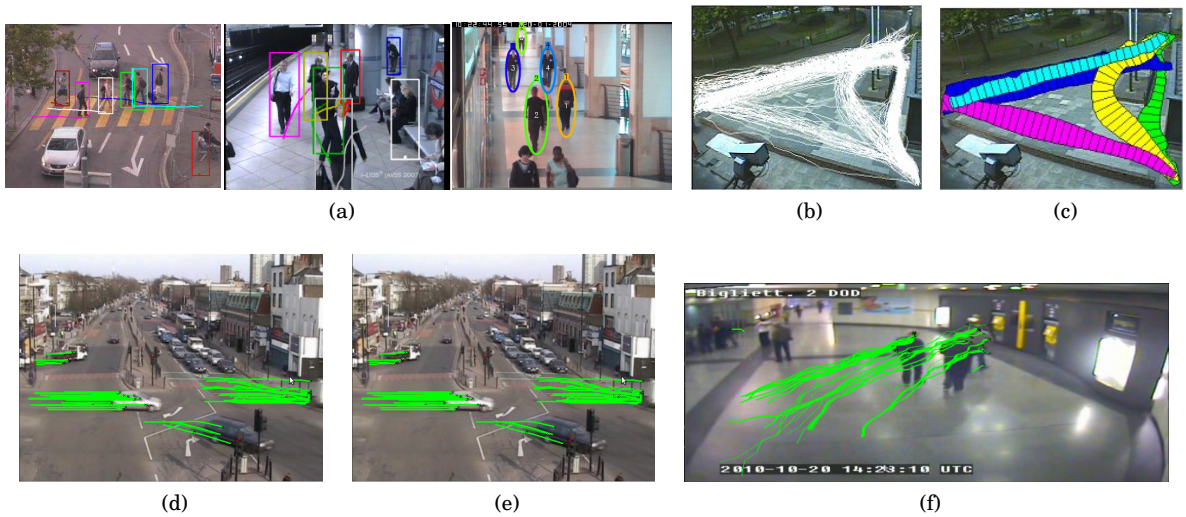


Figure 2.3. Tracking and tracklet based approaches for activity analysis. (a) Tracking humans for behavior analysis (Breitenstein *et al.*, 2009b; Heili *et al.*, 2011). (b) Trajectories from long term observations. (c) Custering trajectories in (c) to infer scene semantics, taken from (Makris and Ellis, 2003). (d–f) Tracklets from outdoor and indoor scenes, courtesy (Jouneau and Carincotte, 2011a,b)

tiple objects in a scene simultaneously is a challenging task. Trackers need to be initialized dynamically, and there is severe object occlusion due to complex interactions. To solve this, researchers have used several alternate strategies like using multiple cameras as in (Yao and Odobez, 2008b). or tracking by detection methods as in (Andriluka *et al.*, 2008; Heili *et al.*, 2011).

After successful tracking, object specific information such as its position, velocity, acceleration and curvature from the trajectory can be used for higher level tasks like activity detection, scene analysis and abnormality detection (Andriluka *et al.*, 2008; Fu *et al.*, 2005; Hervieu *et al.*, 2008; Makris and Ellis, 2003; Stauffer and L.Grimson, 2000; Wang *et al.*, 2004, 2008b). One such example of detecting paths in a scene is shown in Figure 2.3(b,c). The object tracking research domain has grown into a matured area with a vast literature over the past few years due to its importance in several applications. Please see the review by Yilmaz *et al.* (2006) for more details. While tracking gives good results in uncluttered scenarios, it is sensitive to background clutter, small object size, complex object shape appearance and motion, illumination changes, object occlusion and inaccurate camera calibration. The problem becomes more pronounced in the case of crowded scenes as frequent occlusions make reliable tracking almost impossible. Most importantly, while dealing with surveillance tasks there are varieties of objects, *e.g.*, varieties of vehicles and pedestrians with varying size based on the view point appearing in the scene, which need to be modeled and tracked. Also, the visual features such as velocity and curvature from the trackers depend on the distance and the angle of the viewed action. Thus most of the feature representations derived from trajectories are not translation, rotation and scale invariant at the same time. Therefore, it is tough to deploy a generic tracker to obtain successful results in surveillance settings.

2.1.5 Tracklets

Tracklets are intermediate representations, meaning, they have the advantages of both pixel level features like optical flow and object level features *e.g.,* see Figure 2.3(d–f). Object trajectories from a scene are a rich source of activity information, but suffer from several challenges. On the other hand, pixel based features can be readily extracted but provide less object centric information. Typically in multi-object tracking scenarios, there is severe inter-object occlusion creating short term trajectories of objects. Tracklets can also come from feature point tracking methods like the KLT tracker (Tommasi and Kanade, 1991). Their temporal support can vary depending upon the specific circumstances. Using tracklet extraction methods in uncluttered scenes can result in consistent, long tracklets approximating the object trajectory quite well. In cluttered and crowded scenes however, tracklets can be discontinuous extending not more than a second.

Traditionally, researchers have used tracklets by merging them to create longer trajectories (Singh *et al.*, 2008). But in some cases, tracklets themselves have been used as features for higher level processing. For example Chan *et al.* (2006) propose a method that jointly performs tracklet linking and event detection in an airport refueling setup. Jouneau and Carincotte (2011b) analyzed tracklets through a cascade of HMM and HDP-HMM models to detect abnormal events in a traffic junction. In (Zhou *et al.*, 2011), tracklets are linked using an Markov random field topic model and further used for identifying regular paths, sources and sinks from the crowded New-York central station.

2.1.6 Vocabulary design

In practice, many activity analysis methods quantize features into discrete bins called words. Then, their raw occurrences or accumulated counts (through histograms) called "bag-of-words" are used in successive analysis steps. For quantization or dictionary learning, two main strategies are used: *Predefined*, where the vocabulary is defined *a priori* or *Adaptive*, where the vocabulary is learned for instance, through clustering.

Predefined vocabulary. The quantization steps are decided based on some *a priori* knowledge such as frame size or flow directions. For instance, pixel position in an image can be quantized into 10×10 non-overlapping grids and flow directions can be quantized into the four or eight cardinal directions. A predefined vocabulary is simple to obtain and allows an easy interpretation, but an increase in the feature dimension increases the vocabulary size drastically. But increased vocabulary can be allocated to background regions in the image that do not have any activity. Further, this will also accommodate some feature combinations that are never observed in the data *e.g.,* a high velocity in a pedestrian area. A traditional way to deal with this is to use some training data and keep only words that appear a minimum number of times.

Adaptive vocabulary. This is a data driven approach to creating vocabulary. The quantization steps are learned from the features by usually employing a clustering approach like K-Means, Gaussian Mixture Model (GMM), or Probabilistic Latent Semantic Analysis (PLSA). The advantage here

is that the vocabulary is decided based on the distribution of the data and thus concise enough to represent the data. However, the quantization depends on the density of the feature space: high density regions are given many words, while low density regions are not represented well which is typical of clustering approaches. A more important question is the choice of the vocabulary size; while the vocabulary size should depend on the complexity of the data, it is usually hand picked with some *a priori* knowledge. Furthermore, whenever the training data is updated, it has to be followed by a fresh round of learning to update the vocabulary.

2.2 Learning Methods in activity modeling

Learning methods are used to derive patterns from the low-level features. These patterns are then used to identify an activity or tell apart an unusual behavior from the rest. Learning methods differ based on the amount of labels required to learn the patterns. Conventionally, learning methods are broadly classified into unsupervised and supervised methods. Based on the quality and quantity of labels and when the labels are sought during the learning process, there are several variants of supervised learning methods namely: weakly supervised, semi-supervised, active learning and online learning. We will restrict our review to supervised and unsupervised methods in the coming sections.

2.2.1 Supervised activity modeling

Supervised learning methods require that the labels for the different classes are well defined and all the training samples are labeled before the learning phase. This is usually suitable when the events are well defined as in cases such as detecting unattended luggages (Smith *et al.*, 2006), queue formations (Naturel and Odobez, 2008) and vandalism in public places (Sacchi and Regazzoni, 2000). Many earlier methods for activity analysis adopted a supervised learning strategy. Brand *et al.* (1997) trained a Coupled Hidden Markov Models (CHMM) for gesture classification, where each gesture sample has a class label. The likelihoods coming from the learned CHMM were then used to classify test gesture samples. Brand and Kettnaker (2000) and Jiao *et al.* (2004) used a set of normal trajectories to train HMMs for each event class and used this to detect abnormal trajectories. Gong and Xiang (2003) used a Multi linked Hidden Markov Model, where each state is made to represent one of the event classes known *a priori*. Actions of daily living like drinking, walking and punching (Laptev and Lindeberg, 2005; Laptev and Pérez, 2007) were recognized by training classifiers on labeled and segmented action samples obtained from movies. Other flavors of supervised learning such as logic or rule based learning, where domain knowledge is provided in terms of well defined logical statements were also used for activity analysis (Shet *et al.*, 2005; Tran and Davis, 2008).

While supervised methods can perform quite well when provided with accurate labels and a suitable classifier such as Boosting, SVMs, Random Forest, it is usually hard to define positive and negative classes clearly in surveillance settings. In complex scenes, multiple activities occur simul-

taneously. Therefore, the same clip can be associated with for instance, a normal event class and an abnormal event class. Moreover, many supervised activity analysis and recognition methods similar to object recognition, assume that actions can be segmented (temporally) and detected (detecting the action performer) from the rest of the scene. While tracking based approaches, by nature, separate objects and their activities from the scene, action recognition methods such as (Laptev and Pérez, 2007) and (Patron *et al.*, 2010), compile datasets with detailed annotations about when and where activities occur in the video. But, labeling huge quantities of video with precision on time and space is a laborious, time consuming and error prone task. Furthermore, in many surveillance setups, it is difficult to come up with a precise definition of normal and abnormal classes due to the large number of actions and their variations.

2.2.2 Unsupervised activity modeling

In contrast to supervised methods, unsupervised methods do not use labels of the data points during the learning process. Unsupervised activity analysis have been mostly based on clustering algorithms. Clustering is the process of grouping similar data points into a number of subsets called clusters such that samples within a cluster are close to each other with respect to a distance measure. Clustering methods should address three main issues to obtain valid outputs: first, the choice of distance or similarity measure; second, the choice of the clustering algorithm (*e.g.,* iterative, hierarchical, co-occurrence based and online adaptive) method; and third, the choice of model size (number of clusters).

Both pixel level features like optical flow or object centric features like trajectories have been treated with clustering methods to obtain higher level information about the scene. For example, Xiang and Gong (2005) use statistics like centroid, width, height, filling ratio from foreground blobs and motion features with a Gaussian Mixture Model (GMM) clustering to derive event classes. The posterior probabilities from the GMM is then used with a multi-observation HMM to derive composite behavior models for the clips. Yang *et al.* (2009) proposed a co-occurrence based approach to cluster quantized optical flow features and derive motion patterns from traffic scenes. Alternatively, Zen and Ricci (2011) use optical flow direction and magnitude, percentage of foreground pixels in a given region along with convex optimization procedure based on Earth mover's distance to discover behavior prototypes and salient activities from traffic scenes and basketball shows. Due to the convex optimization procedure, it is not prone to local minima. In (Saleemi *et al.*, 2010), they argue for the benefits of unquantized optical flow features and use this with a hierarchical clustering approach to derive motion patterns.

Trajectory clustering methods were also quite popular to derive scene semantics. Trajectories are represented as time series, where the observations for each point in the trajectory come from features such as velocity, curvature, blob size and color. Given that trajectories are of variable lengths, most methods focused on using time warping methods like Dynamic Time Warping (DTW) and Hidden Markov Model (HMM) to calculate trajectory similarities (Brand and Kettnaker, 2000; Hervieu *et al.*, 2008). Porikli (Porikli, 2004) modeled trajectories using an HMM, whose param-

eters were used as features for clustering. A Hausdorff distance based trajectory clustering was used in (Wang *et al.*, 2004) to learn common vehicle activities. A co-occurrence based hierarchical approach to cluster trajectories was proposed in (Stauffer and L.Grimson, 2000) to learn activity patterns. Alternatively, in (Fu *et al.*, 2005), they addressed the time warping problem of trajectories by re-sampling them at equal space intervals. A spectral clustering was then used to group trajectories into dominant paths and lanes. (Zhang *et al.*, 2006) also compare various trajectory similarity measures for clustering and surveillance in outdoor scenes. The results of trajectory clustering could have many applications. The clustered trajectories can be used to learn common activity classes. When a new trajectory is presented, it can be classified into one of the learned classes or an abnormal one if it diverges from every known class beyond an acceptable threshold.

Apart from clustering approaches, unsupervised methods also include non-parametric or model free methods (Boiman and Irani, 2007; Breitenstein *et al.*, 2009a). These methods usually work by building a database of common patterns from normal observations and detecting irregularities or novelties by looking at their similarity to the database of objects.

Labeling data being a prime concern in activity analysis, unsupervised methods have been more popular in the recent days. However, as mentioned earlier, the choice of the clustering algorithm, the similarity measure and the number of clusters are some important parameters that have implications in the performance of the system. Moreover, since clustering methods run typically without any human intervention, some amount of human interpretation and analysis of the clusters is involved at the post-processing stage.

2.2.3 Probabilistic Topic Models

Probabilistic Topic Models (PTM) are a class of Bayesian belief networks that deal with discrete data. They were initially proposed for text data mining to automatically discover main themes or topics from large corpus of text documents. The text corpus is usually given as a set of documents, represented by simple unordered word counts or bag-of-words, ignoring the order in which they occur. Through the analysis of word co-occurrences in documents, PTMs discover the topics and their importance in the document composition. Models like Probabilistic Latent Semantic Analysis (PLSA) (Hofmann, 2001), Latent Dirichlet Allocation (LDA) (Blei *et al.*, 2003a) and, Hierarchical Dirichlet Processes (HDP) (Teh *et al.*, 2006) are some well known examples of PTMs.

With a domain specific design of vocabulary, they have been applied to mine patterns from a wide range of data logs such as, natural images (Fei-Fei and Perona, 2005; Quelhas *et al.*, 2005), action videos (Niebles *et al.*, 2008), cell phone logs (Farrahi and Perez, 2008) and wearable sensors (Huynh *et al.*, 2008). Applications of topic models in video scene analysis started as a niche area, but has received considerable attention recently due to its success in discovering semantically meaningful patterns from simple low-level visual features in an unsupervised fashion (Hospedales *et al.*, 2009; Kuettel *et al.*, 2010; Li *et al.*, 2008; Wang *et al.*, 2008b). Additionally, it brings in the powerful tools of probabilistic generative models enabling us to model complex real life phenomena. For example, Wang *et al.*, (Wang *et al.*, 2008b) introduced the use of location and optical flow features along with

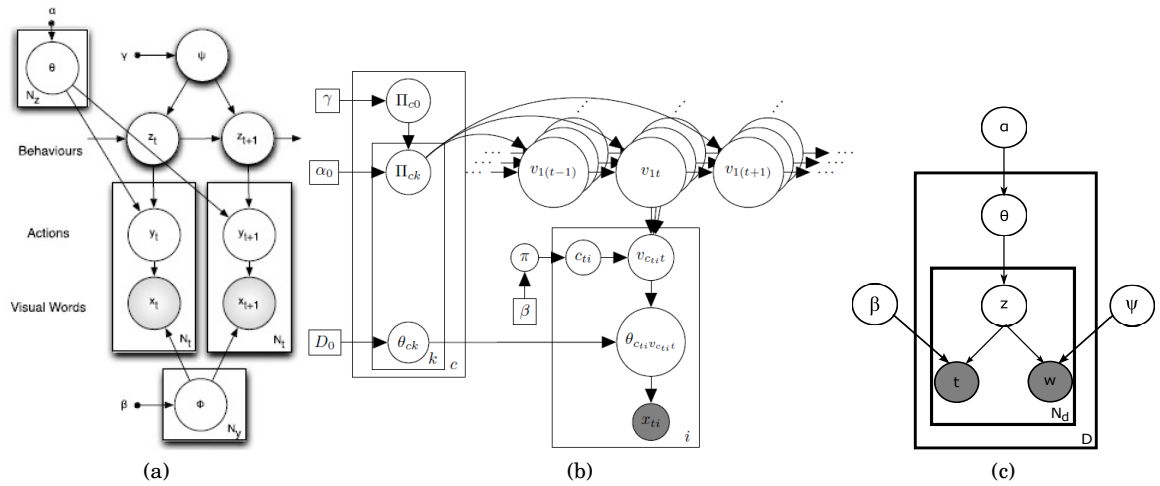


Figure 2.4. Introducing time in PTMs. (a) Markov clustering topic model: uses time at a higher level of the scene,(b) Dependent Dirichlet Processes- Hidden Markov Model - infinite mixture of infinite HMMs, (c) Time LDA model: concatenates features and their time of occurrence within a manually segmented clip as a word.

a hierarchical Bayesian approach to identify activities and their interactions from video of a four road junction. Li *et al.*, (Li *et al.*, 2008) used PLSA on clustered low-level features to infer activities in local regions at the first level, and global activity correlations at the next level. This was then used to detect anomalies in a traffic scene.

In our own work (Varadarajan and Odobez, 2009), we used PLSA to extract dominant patterns of activities from low-level features and sort the patterns in terms of dominant features in the topic. The inferred activities were then used to segment the scene into regions of homogeneous activities. We used a richer feature set compared to (Wang *et al.*, 2008b) by considering also static pixels and foreground blob size.

2.2.4 Temporal modeling with PTMs

Although PTMs are able to discover scene activities, the actual modeling of temporal information remains an important challenge. By relying only on the analysis of unordered word co-occurrence (due to the bag-of-words approach) within a time window, most topic models fail to represent the sequential nature of activities, although activities are often temporally ordered. For example, in traffic scenes, people wait at zebra crossings until all vehicles have moved away before crossing the road, giving rise to a temporally localized and ordered set of visual features. Concatenating word (feature) counts over a large temporal window and representing activities as "static" distributions as is done in PLSA implementations may be concise but not complete, as it does not allow us to distinguish it from an abnormal situation where a person crosses the road while vehicles are still moving.

Recently, several approaches have been proposed to include sequential information in text modeling. This was done either to represent single word sequences (Gruber *et al.*, 2007; Wallach, 2006),

or at a higher level, by modeling the dynamics of topic distributions over time (Blei and Lafferty, 2006b; Gohr *et al.*, 2009; Wang *et al.*, 2008a). For instance, (Wallach, 2006) introduced word bi-gram statistics within a LDA-style model to represent topic-dependent Markov dependencies in word sequences, while in the Topic over Time method of (Wang and McCallum, 2006), topics defined as distributions over words and time were used in a LDA model to discover topical trends over the given period of the corpus.

Many of these temporal models have been adapted for activity analysis. For instance, (Hospedales *et al.*, 2009) introduced a Markov chain on scene level behaviors. As seen in Figure 2.4(a), the chain relates the behaviors, but activities are still considered as a mixture of unordered (activity) words. More recently, (Kuettel *et al.*, 2010) used the HDP-HMM paradigm shown in Figure 2.4(b) (Hierarchical Dirichlet Process, HDP, and Hidden Markov Model HMM of (Teh *et al.*, 2006)), to identify multiple temporal topics and scene level rules. Unfortunately, for all four tested scenes only a single HMM model was discovered in practice, meaning that temporal ordering was concretely modeled at the global scene level using a set of static activity distributions, similar to what was done in (Hospedales *et al.*, 2009). Another attempt was made in (Li *et al.*, 2009), which modeled topics as feature$\times$time temporal patterns, trained from video clip documents where the timestamps of the feature occurrences relative to the start of the clip were added to the feature. However, in this approach, the same activity has different word representations depending on its temporal occurrence within the clip, which prevents the learning of consistent topics from the regularly sampled video clip documents. To solve this issue of activity alignment with respect to the clip start, in (Faruquie *et al.*, 2009) (see Figure 2.4(c)), manually segmented clips were used so that the start and end of each clip coincided with the traffic signal cycles present in the scene. This method has two drawbacks: firstly, only topics synchronized with respect to the cycle start can be discovered. Secondly, such a manual segmentation is time consuming and tedious. From this we observe that none of the existing approaches addressed the issue of recovering activities with their exact temporal structure without any manual segmentation of the videos.

To this end, we propose a novel topic model called ***P**robabilistic **L**atent **S**equential **M**otifs* **(PLSM)**, that recovers the temporal order in which words occur within an activity motif, solving the challenging case of temporal overlaps of several activities in the scene.

2.2.5 Model Selection

As mentioned in section 2.2.2, the size of the model is an important parameter to be determined in unsupervised learning methods that are akin to clustering. Usually in real-life scenarios, we have some rough *a priori* knowledge of the number of clusters. This is the case, for instance, in our video activity analysis scenarios, where this number qualitatively depends on the scene complexity and the features used. Still, being able to adapt the model size as a function of the actual data is desirable. In our problem, the model selection issue translates to identifying an appropriate number of topics or motifs. Activity analysis methods that relied on trajectory clustering have dealt with this problem in several ways. For instance, Morris *et al.*, in (Morris and Trivedi, 2008a), first over cluster

trajectories followed by an agglomerative merge procedure. Minimization (maximization) of some optimality criterion like cross validation performance, ratio between inter-cluster and intra-cluster distances, were also used in number of works (Atev *et al.*, 2006; Figueiredo and Jain, 2002; Hu *et al.*, 2006; McLachlan and Peel, 2005; Porikli, 2004). Information theoretic methods like the Bayesian Information Criterion (BIC) (Chen and Gopalakrishnan, 1998; Jiao *et al.*, 2004; Schwarz, 1978) in essence seek models that find a compromise between likelihood fitting and model complexity (Xiang and Gong, 2006). Using a Bayesian view, some methods explicitly compute models of different sizes and store them. A prior on the model size is then used along with all the different models in the inference stage (Richardson and Green, 1997). But this method is time consuming and involves marginalizing over all the different models and their prior values. Additionally, the prior on the model size should be carefully modeled. A non-informative prior could do more harm than using a selected model. Many works on topic models evaluate a set of models using the perplexity measure or predictive likelihood on held-out data (Blei and Lafferty, 2006a; Corduneanu and Bishop, 2001) and keep the best performing model for future use.

Recent developments in non-parametric methods have revealed an elegant approach to model selection, where one can have in theory an infinite number of topics, but with a finite amount of data, select an appropriate number. The term non-parametric means that the model complexity is never fixed and can grow with the data. Non-parametric models can automatically infer an adequate model size from the data, without the need for an explicit model comparison step.

One example is a Dirichlet process (DP), defined formally as follows: given a probability measure G_0 over a measurable space Θ, and a positive real number α, $G \sim DP(.|G_0, \alpha)$ means that for any partition $\{A_1, A_2, \ldots, A_k\}$ of Θ,

$$\{G(A_1), G(A_2), \ldots, G(A_k)\} \sim \text{Dirichlet}\left(\{\alpha G_0(A_1), \alpha G_0(A_2), \ldots, \alpha G_0(A_k)\}\right) \tag{2.1}$$

The two parameters of a DP are: a base distribution G_0 and a positive real number α, called the concentration parameter. $DP(G_0, \alpha)$ has the same support as G_0 and a draw from DP results in a discrete distribution even if G_0 is continuous. Intuitively, a DP can be thought of as an infinite dimensional Dirichlet distribution or a distribution over distributions. The Stick breaking construction by Sethuraman (1994), the Chinese Restaurant Process and the more general Pitmann-Yor process are some popular methods for realizing a DP.

The Hierarchical Dirichlet Processes (Teh *et al.*, 2006) by Teh *et al.*is a non-parametric topic model that makes use of Dirichlet processes at multiple levels to select the number of topics automatically for a document corpus. A similar model was explored for automatically discovering static activity topics and interactions from traffic video clips in (Wang *et al.*, 2009) and selecting the model order.

Inspired by these non-parametric methods for model selection, (in collaboration with Rémi Emonet) we propose a new model called the ***H**ierarchical **D**irichlet **L**atent **S**equential **M**otifs* (HDLSM) (Emonet *et al.*, 2011a), a non-parametric Bayesian improvisation over the PLSM model.

HDLSM incorporates a new generative model using the concepts of DP at multiple levels: a DP at a higher level to select the number of motifs, and another DP at a lower level to select the motif occurrences. The model parameters are inferred using a Collapsed Gibbs sampling scheme [1].

2.3 Inferring scene semantics

In this section, we briefly review some of the methods proposed in the literature to infer higher level scene semantics such as scene cycles, causality and periodicity.

Many attempts have been made to identify global scene states [2] . Infering global scene states and their sequence is of interest because they help us understand the presence of global traffic rules, cyclic patterns and furthermore, provide context to detect aberrations in scene behaviors. Recently, few methods have attempted this task. In methods similar to PLSA which do not model time explicitly, the dominant topic or cluster at the time instant is taken to be the scene state (Faruquie *et al.*, 2009; Li *et al.*, 2008; Zen and Ricci, 2011). When a DBN like model is used, this is given by the hidden states of the Markov model (Hospedales *et al.*, 2009; Kuettel *et al.*, 2010).

In the case of modeling activity dependencies and interactions, most methods rely on some form of DBNs with complex links between the hidden states to capture this information. Coupled HMM (Brand *et al.*, 1997), parallel HMM (Vogler and Metaxas, 1999) and multi-observation HMM (Xiang and Gong, 2008) are some examples of this. In all these cases, the Markovian assumption constrains the dependency of a current event to an event in the immediate past, while in reality, it can be caused by some event, farther back in time. In dynamic topic model based methods (Hospedales *et al.*, 2009), the Markov chain runs at a higher level between distinct global scene behaviors, correlating only the global states and not associating the local activities. Additionally, since the states are usually a mixture of local events, it is not clear as to which local event in the past triggered the current one. Kuettel *et al.* (2010) therefore discover both local and global rules separately. For local rule finding, they rely on an exhaustive exploration of activity combinations and on a comparison with predefined Markov templates which is both hard to compare to and not scalable.

Among non-Markov style methods, Wang *et al.* (Wang *et al.*, 2009) infer multi-object interactions from scenes using an improvisation of the Hierarchical Dirichlet Processes model. But their method uses static distributions for topics and interactions and does not model the dynamics within the interactions. In both single and muti-camera setups, researchers have proposed models to infer causal relationships among events and regional activities by considering the observations over time as a time series and computing measures to assess their pairwise relationships. Prabhakar *et al.* (2010) consider a point process representation of the individual time series' and infer causality by analyzing the pairwise relationships of the process spectral representation. (Loy *et al.*, 2009) use

1. Please see section 6.2.4 for more details on collapsed Gibbs sampling.
2. A global scene state can be defined as a rule of the scene that dictates what activities can occur for the duration of the state. For example, in traffic controlled scenes, the traffic lights turning red or green determines which activities can occur and when they occur in the scene as a whole (vehicles will stop in one side of the road, while stopped vehicles start to move on the other side).

canonical cross correlation and time delayed mutual information methods to correlate time series that represent events in local regions of a multiple camera setup. Since these methods are free from the first order Markovian assumption, they capture arbitrary temporal lags among individual local events. But associating a causal dependence to every event is not necessary since there could be independent events that are triggered by a global scene state.

Our work presented in chapter 6 addresses this issue of inferring global rules while not compromising on local activity dependencies. In a novel model called ***M**ixed **E**vent **R**elationship* (MER) model, we consider that activities can be either independent or dependent. Independent events are triggered by global behavior states while dependent events can be associated to any past event, not just limited to event in the immediate past.

2.4 Performance Evaluation

Several quantitative methods exist to objectively evaluate activity analysis methods. But there is little agreement on a common set of tasks and performance measures. This is mainly due to several problems such as: i) unavailability of public datasets due to privacy concerns; ii) variations in video data; iii) variety of tasks attempted and iv) incompatibility of methods and their outputs. Due to some recent efforts (Li *et al.*, 2008; Wang *et al.*, 2009), more and more video datasets of public settings are available for activity analysis research, while simultaneously bringing in some convergence of ideas.

Supervised methods. As it is common in supervised methods, test data contains both positive and negative samples, and their labels are estimated using the learned model. The detection or classification accuracy on the test set is then used to evaluate the model performance.

Unsupervised methods. Several evaluation methods are used when the learning strategy is unsupervised. Quality of the clusters learned can be estimated objectively using measures such as ratio of average intra-class distances *vs* average inter-class distances, cluster compactness and cluster homogeneity (Zen and Ricci, 2011). When trajectories are clustered into different paths of the scene, clusters can be used to classify test trajectories into one of the learned paths. Outlier trajectories and clusters with very small support can give information about possible abnormalities. This can be evaluated easily, as in the case of supervised learning, if we have labels for test trajectories (Morris and Trivedi, 2008b). While using clustering methods or PTMs on low-level visual features, a similar approach can be followed where test clips can be labeled as normal or abnormal and compared with the ground truth. This was used to detect typical traffic violations like jay-walking, vehicles passing out of turn and vehicles skipping red lights (Hospedales *et al.*, 2009; Li *et al.*, 2009; Wang *et al.*, 2009). In the above cases, we observe that though an unsupervised learning strategy is adopted, it reduces to a one class classification problem where the training class is carefully selected with predominantly normal data samples, calling for the need of labels. Inevitably, we resort to creating a test set with some subjective interpretation of an abnormality.

In some cases, the learned clusters provide labels that have a real-world interpretation. For

instance, in (Saleemi *et al.*, 2010; Wang *et al.*, 2009), each video clip can be labeled into one of several interaction classes. The labels from clustering are then compared with the labels from human annotations to evaluate the model's effectiveness.

A similar evaluation method can be adopted when patterns from PTMs can be associated with real world events such as vehicle taking a right turn or people crossing the road. This serves as an unsupervised event detector, whose performance can be evaluated on a test set with objective human labels on event occurrences. The results returned by the model are evaluated using precision and recall measures.

Online prediction task. Here, the learned model (*e.g.,* from PTM) can be used to infer the parameters on partial observations up to time t, and predict the activity at time $t+1$. The predictions can then be objectively compared with observations at time $t+1$. A well learned model is expected to give a relatively high predictive likelihood. This evaluation method has several advantages. Firstly, it requires no human labeling and uses the observation directly for evaluation. Secondly, it can be extended for checking online abnormal occurrences, where a low prediction at any time could potentially indicate an unusual occurrence and thirdly, this could be extended to evaluate time series model in general.

2.5 Summary

In this chapter we surveyed the essential parts of a typical video activity analysis method namely, video representation, activity modeling, inferring higher level scene semantics and evaluation. In the video representation part, we discussed pixel-based and object-based approaches and their pros and cons in representing video content. Following this, the discussion on learning methods for activity analysis revealed that coming up with a clear definition of all possible normal and abnormal events is difficult. This makes a strong case for unsupervised supervised or semi-supervised methods over supervised methods. A recent trend in unsupervised methods for activity analysis called PTMs was discussed in detail with its accompanying aspects like model selection, and higher level scene semantics inference. Finally, we also listed some of the well known evaluation tasks adopted in this domain.

After this careful study, we opt for pixel level features: a combination of the foreground pixels, blob size and optical flow vectors. We detail our initial experiments with the PLSA, LDA models to discover and analyze dominant activities from the videos in chapter 4. Our study revealed that PTM's were not properly adapted for recovering activities from videos, meaning, the temporal aspect of activities were not modeled at the individual activity level. We address this issue by proposing the PLSM model in chapter 5. Advancing this further, the PLSM model is reformulated using non-parametric Bayesian methods in Appendix B. In order to infer higher level scene semantics like global behavior states and event associations that are not necessarily Markovian in nature, we propose the MER model in chapter 6. Furthermore, we evaluate the proposed methods on different tasks including abnormality detection, event detection and online prediction.

Chapter 3

Datasets and Feature Extraction

It is often said that data in hand determines the research problem. While it is arguably true, the idea is less disputed in many computer vision and machine learning research areas. In this thesis too, datasets play an important role and hence deserves a special chapter, especially to illustrate its content. In this chapter, details about the datasets used in this thesis are presented in section 3.1. In section 3.2, details on the feature extraction and representation procedures followed in this thesis are presented.

3.1 Datasets

In this thesis, we validate the proposed methods through experiments on a variety of video and audio datasets. All the video datasets used here are from static cameras. However, this does not preclude the use of our models on other types of videos obtained from moving cameras. By adapting the visual features and words (*e.g.,* trajectories, spatio-temporal interest points) accordingly one can still apply the proposed models. The datasets used here vary in the following aspects:

- videos of simple to complex scenes of urban traffic, having free as well as signal controlled movements;
- videos of indoor scenes from crowded metro stations spanning single and dual views;
- Time Delay Of Arrival (TDOA) data of a traffic scene recorded using binary microphones.

We will present each dataset in the following sections.

3.1.1 Outdoor traffic scenes

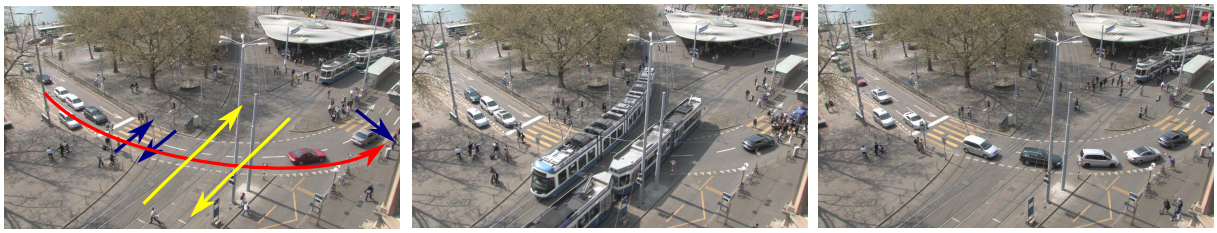
MIT data. The MIT scene (Wang *et al.*, 2009) is a two-lane and four-road junction captured from a distance, where there are complex interactions among vehicles arriving from different directions, and few pedestrians crossing the road (see Figure 3.1(a) for sample frames and common activities). This has a duration of 90 minutes, recorded at 30 frames per second (fps), and a resolution of

(a) Dominant activities from MIT data (Wang *et al.*, 2009) along with some sample frames

(b) Dominant activities from Far-Field data (Varadarajan *et al.*, 2010) along with some sample frames

(c) Dominant activities from Traffic Junction data (Varadarajan and Odobez, 2009) along with some sample frames

(d) Dominant activities from Junction data (Li *et al.*, 2008) along with some sample frames



(e) Dominant activities from ETH-Zurich data (Kuettel *et al.*, 2010) along with some sample frames

Figure 3.1. Outdoor urban scene datasets used in this thesis.

480×756 which was down-sampled to half its size.

Far-Field data. The Far-field scene (Varadarajan *et al.*, 2010) depicts a three-road junction captured from a distance, where typical activities are moving vehicles (see Figure 3.1(b) for sample frames and common activities). As the scene is not controlled by a traffic signal, activities occur at random with large variations in their speed. The video duration is 108 minutes, recorded at 25 fps and a 280×360 frame resolution.

Traffic Junction. This is a video captured from a portion of busy traffic-controlled road junction (Varadarajan and Odobez, 2009). The video is 44 minutes long recorded at 25 fps sampling rate, and has a frame size of 288×360. Few sample frames with common activities are shown in Figure 3.1(c). The scene has multiple activities that include people walking on the pavement, people waiting for vehicles to cross, people crossing on zebra crossings, vehicles moving in and out of the scene in different directions etc. Some unusual activities such as people crossing the road at the wrong place (out of zebra crossing), vehicle parked at the pedestrian path and vehicles stopping ahead of the stop line are observed here.

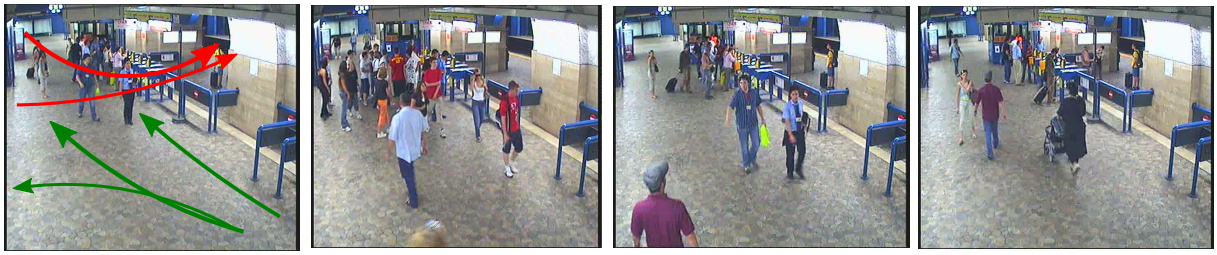
QMUL Junction data. The QMUL Junction data, with sample frames in Figure 3.1(d) due to (Li *et al.*, 2008), is filmed from a four road junction scene in London. The video is 1 hour (90000 frames) long, recorded at 25 fps and a resolution of 360×288. The scene is regulated by traffic lights and dominated by four types of traffic flows. There is a continuous flow of traffic in the scene for the entire duration of the video. Instances of jay-walking, unusual traffic flows caused by fire engine and ambulance are observed in this dataset.

ETH Zurich data. This dataset with some sample frames and activities in Figure 3.1(e) is taken from a junction, where activities due to pedestrians crossing the road, cars and trams moving along the road are seen. This scene is also controlled by traffic lights. The video is 1 hour long, recorded at 25 fps with a resolution of 900x560.

3.1.2 Metro indoor scenes

Rome metro station. The **Metro station** data (Figure 3.2(a)) is captured from a static camera looking at a hallway. As colored arrows show some dominant movements: people arriving there from several directions, buying tickets, staying in the hall, or going through turnstiles leading to the train platform. The scene is usually crowded with a high degree of unstructured movement by people. More importantly, due to the low view point, motion at a given image location can be due to people moving at different depths in the scene, making the low level image measurements highly ambiguous. The video is 120 minutes long and captured at 5 fps with a frame resolution of 576×720.

Torino metro station. We also tested our model by combining data from multiple synchronized views of Torino station (Emonet *et al.*, 2011b). We considered two kinds of binary views: 1) non-overlapping views and 2) overlapping views. The first scene (Figure 3.2(b)) is made of two non-overlapping views, recording two neighboring areas: a stairs/escalator area and a walking area. In the bottom view of Figure 3.2(b), we can see people taking the escalator or the stairs to go up or get



(a) Rome metro station

(b) Torino metro station: Non overlapping views

(c) Torino metro station: Overlapping views

Figure 3.2. Indoor, unstructured scenes from Rome and Torino metro stations. a)Rome metro station b) orthogonal (overlapping) views of a ticketing hall fused into one montage. c) Non-overlapping views showing an stairs and escalator (below) and a passage hall (top) fused into one

down respectively. The staircase leads to the left side corner of the passage hall that is in on the top view of Figure 3.2(b). Some common motion patterns are given using the colored arrows.

The second scene (Figure 3.2(c)) is from two overlapping views of a ticketing hall of the metro station. The hall is connected to the two station entrances, contains two vending machines and has a row of turnstiles used as entry or exit points to the metro network. Due to the overlapping views, the activities that can be seen from both the cameras are marked with same colors. People enter through the turnstiles and use the escalators that are at the rear-end of Figure 3.2(c) bottom view. Exiting people (marked in cyan) walk out of the sides of this view and appear at the rear-end of

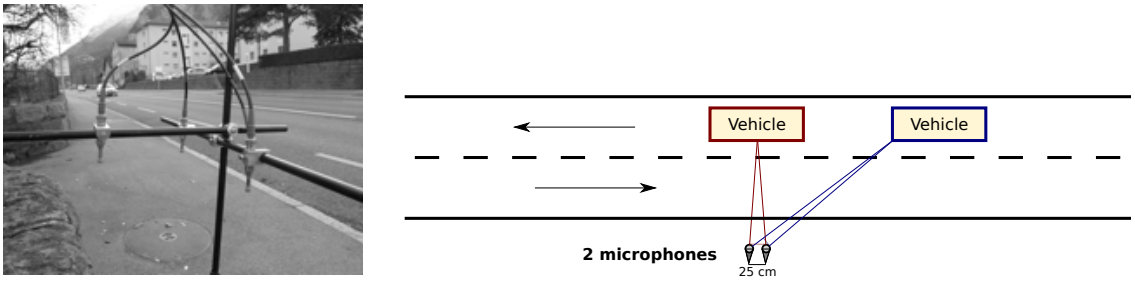


Figure 3.3. Audio scene analysis setup. a) A snapshot of the physical: An array of two microphones is located on a road side. b) A graphic showing the physical setup more clearly. The audio time difference of arrival (TDOA) between these microphone is measured and provides information about the azimuths of sound sources.

Figure 3.2(c) top view. Each view is recorded at 5 fps and has 704x288 frame resolution.

3.1.3 Data from micro-phone arrays

To test the generality of the model, we used it for analyzing a scene using acoustic data. The recording setup is described in Figure 3.3.

The recording was done using two microphones located on the side of a two-way road where the main activities are essentially vehicles either going from left to right or from right to left, at different speeds. This dataset has 41 recordings of 20 seconds each. In this experiment, our activity feature characterizes the sound source locations, and relies on the time difference of arrival (TDOA) principle: a sound generated by a source located at an azimuth angle θ relative to the microphone pair arrives at the microphones with a time difference of $\tau(\theta)$ between them. We use dense TDOA information extracted from the microphones to build temporal documents. At each time instant, we compute on an 80ms temporal window the generalized cross-correlation $GCC(\tau)$ between the two signals for different τ values corresponding to azimuth angles from almost $-90°$ to $90°$. We then normalize the measurements, and further subtract a uniform value from the result. The normalization provides some invariance to car loudness, while the subtraction removes uniform noise that might have been amplified by the normalization step. Finally, the representation is simplified by averaging the measurements on 25 regular intervals $\Delta\tau_i$ to measure the "amount" of sound signal coming from the direction $\theta(\tau_i)$ and construct the word-time frequency matrix $n(w_{\tau_i}, t_a, d)$. Figure 3.4 shows a sample document (a clean one) with multiple vehicles passing: five cars going from left to right (upward ramp) and one car going from right to left (downward ramp).

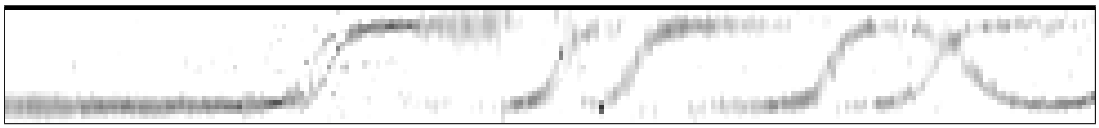


Figure 3.4. TDOA sample temporal document showing multiple occurrences of cars going from left to right (upward ramp) and one occurrence of car going from right to left (downward ramp) overlapping. The horizontal axis is time (one time step is 80ms), the vertical axis is the azimuth angle.

3.2 Feature extraction

As argued in chapter 2, we rely on simple pixel level visual features for video representation due to the efficiency in extraction and their ability to represent various activities in any scene. The video is therefore represented using foreground pixels, optical flow and in a special case the size of foreground blobs too. The adopted topic model approaches use counts of discrete data entities called words, which are collected in sets called documents. Therefore, to discover activity patterns using topic models, we need to define our vocabulary (the set of visual words characterizing the scene content), and how we build our video documents. In our case, a visual activity could be described by three types of features: location, motion, and size features.

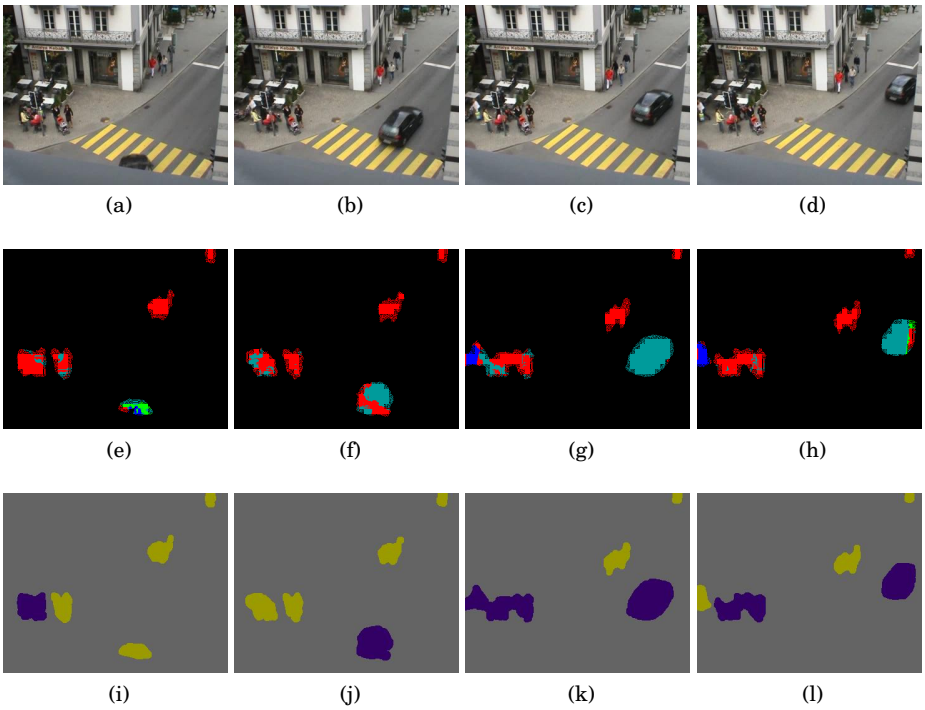


Figure 3.5. Demonstrating quantized features. (a–d) Images from Traffic Junction scene, (e–h) Optical flow vectors of foreground pixels quantized into four cardinal directions and static pixels. shown in five colors: red, light and dark blue, cyan and green. (i–l) Foreground blob size is classified into two classes and displayed with color codes: yellow is small and violet is big. We can see that vehicles and group of people being shown under the "big" category.

Location: In many surveillance scenes, most of the activities are characteristic of the place where they occur. Thus, location has to be taken into account when building our vocabulary, and we divide the image into cells of size $C \times C$ pixels. Therefore, for a video of dimension 280×360, with $C = 10$, we obtain a set of 28×36 cells. The cell size has an impact on the resolution of the learned patterns. Small cells result in large vocabularies, but the learned patterns will have a fine spatial resolution, while large cells shrink the vocabulary size significantly and provide a coarse estimate of the action locations. We experimented with different cell sizes for location quantization *e.g.,* 10×10 and 4×4.

Motion: To identify the relevant parts of the scene, we first perform background subtraction using the algorithm proposed in (Yao and Odobez, 2007) and detect the foreground pixels. See Figure 3.5(a–d) for some sample frames from Traffic Junction dataset. Their corresponding foreground regions are shown in Figure 2.2(a–d). For each of them, we also compute its optical flow using the Lucas-Kanade algorithm. Using a small threshold ($= 0.1$) on the resultant optical flow vector magnitude, foreground pixels are categorized into static pixels (static label) and moving pixels. Moving pixels are further differentiated by quantizing their motion direction into four or eight cardinal direction labels. Thus, in total, we have 5 or 9 possible motion words at each location giving a vocabulary of $28 \times 36 \times 5 = 5040$ words or $28 \times 36 \times 9 = 9072$ words respectively for a video of dimensions 280×360. Quantized optical flow with 5 bins is shown with color codes in Figure 3.5(e–h). As in quantizing location, the number of bins used for optical flow quantization increases or decreases the vocabulary size. But a fine quantization gives a precise interpretation of the results in terms of the direction of movement.

Size: To further characterize foreground objects, we can associate with each foreground pixel the size of the connected component it belongs to. The foreground blobs can be roughly classified into two categories based on foreground blob size. The first one consists of small blobs corresponding mainly to pedestrians and the second one consists of large blobs corresponding to vehicles or group of pedestrians. This categorization can be obtained by applying a simple K-Means clustering on the extracted blob sizes with $K = 2$, and use the cluster number as a size word describing roughly the size of objects in the scene. Please see Figure 3.5(i–l) for the color coded size words. We see that pedestrians (yellow blobs) contribute to small size and vehicles and pedestrian groups (violet blobs) contribute to the bigger size.

Size features are interesting when, a) the scene has objects of different sizes so that each object can be analyzed based on their size, b) when views are not crowded, as crowded scene result in connected blobs and do not meet our need, and c) when the scene is of uniform depth, *i.e.,* object size remains roughly constant irrespective of its position in the scene. In our initial experiments in chapter 4, we used blob size features too since, in the Traffic Junction dataset (see Figure 3.1(c)), there are objects of different sizes with their sizes roughly preserved through out the scene. This is not applicable in cases such as Far-Field (Figure 3.1(b)) or MIT data (Figure 3.1(a)) where mostly only vehicles appear and we need to calibrate object sizes based on their distance from the camera.

3.3 Summary

In this chapter we presented the various datasets used in this thesis. The datasets coming from state of the art papers and projects like CARETAKER and VANAHEIM have a good variety in terms of content. The feature and vocabulary design process illustrates how the video content is simplified for further processing. More specific details about the vocabulary and documents will be provided while we discuss the experimental results in the coming chapters.

Chapter 4

Activity Analysis Using PLSA

After deciding to use probabilistic topic models (PTMs) as our data mining tool, we started with the Probabilistic Latent Semantic Analysis (PLSA) by (Hofmann, 2001). In the rest of this chapter, we will quickly review the basic ideas behind PLSA and LDA and see how they can be used in our application of mining activities from videos. We will then present some analysis on the topics extracted followed by their use on two applications: activity based scene segmentation and abnormal event detection.

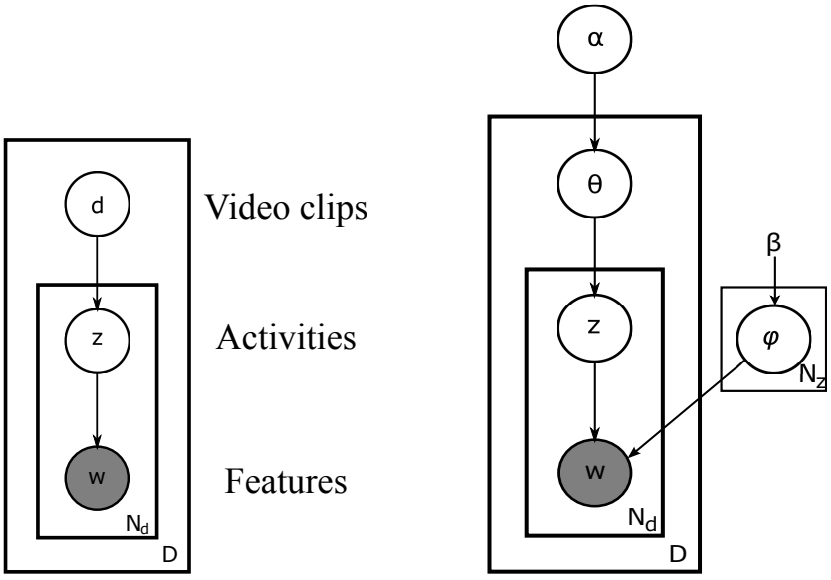
4.1 Introducing PLSA

PLSA and LDA are generative models, meaning, they are based on probabilistic sampling rules that describe how words in a document are generated. To get an intuition of the generative process of PLSA, let us consider that a columnist for *Wall Street Journal* decides to write an article on the Global Economic Crisis. He would first plan his article based on some sub-topics that could possibly be *Economy*, *Stocks* and *Banking* for example. Then, he might decide the importance to be given for each of the sub-topics possibly reflected by the number of words or paragraphs for the subtopics. For example he might decide to write about each of the above topics in about $\{5, 5, 7\}$ paragraphs of the same size respectively. Then, for each topic, he would choose the most appropriate words to convey his ideas on the subject. Now, let us consider for a moment that a computer, ignorant of language grammar and word order is assigned a job to generate a number of such articles using an algorithm. Then if each word is indicated by the variable w, each topic by z and document by d, perhaps it might have the method given in Algorithm 1 in its RAM for drawing a "bag of N_d words" for each document d,

Graphical Model. The procedure described in Algorithm 1 is called a generative process and its pictorial version in Figure 4.1(a) is called the PLSA graphical model. In this notation, the circular nodes represent random variables. Shaded circles indicate observed variables and transparent circles represent latent variables.

Algorithm 1 The PLSA generative model

for $d = 1$ to D; **do**
  for for $j = 1$ to N_d; **do**
    draw a topic $z \sim p(z|d)$
    draw a word $w \sim p(w|z)$
  end for
end for



(a) PLSA (b) LDA

Figure 4.1. The PLSA and LDA topic models. See text for the explanation.

In the case of Figure 4.1(a), w and d are observed and z is called a latent variable which needs to be estimated. The directed edges indicate conditional dependencies. Here, we have w depending on z and the presence of z introduces a conditional independence: a word w and document d are conditionally independent given the topic z, indicated as $w \perp\!\!\!\perp d|z$. Intuitively, this means that words depend only on the topic and not on the document for which it is generated. The plates indicate repetition of the sampling process, where the variable in the bottom right of the plate indicates the number of samples. In Figure 4.1(a), the plate surrounding w and z indicates that z is sampled N_d times, each time followed by a w sample. In other words, for each document d, there are N_d (z, w) pairs.

Distributions. The importance given to each topic is given by the probability distribution $p(z|d)$. In the example taken, this would simply be the proportion of the three topics in the article given by $\{5/17, 5/17, 7/17\}$. Similarly, the number of times each word occurs under a topic gives the distribution $p(w|z)$. This would mean that words like *fiscal deficit, banks* and *GDP* will have high probability under the topic "Economy" and, words like *profit booking, NASDAQ, LSE* and *banks* may occur more frequently under the "Stocks" topic. Note that the term *banks* occurs in more

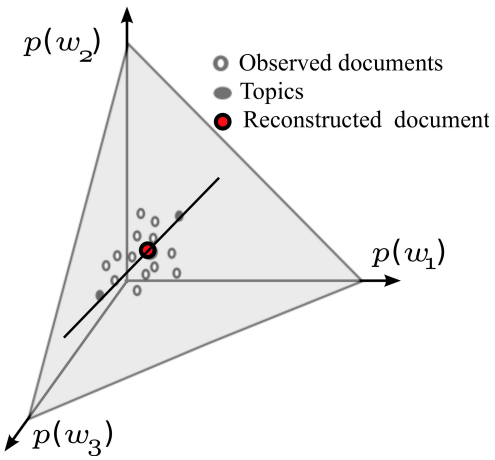


Figure 4.2. A 2 dimensional simplex defined in a three dimensional word space. Topics and documents lie in the simplex. Documents can be expressed as a convex combination of topics. By Hofmann (2001); Steyvers and Griffiths (2007)

than one topic.[1] The conditional independence assumption in the model can be used to split the joint distribution of the model into smaller factors. More precisely, the joint distribution of all the variable triplets (w, z, d) can be written as

$$P(w, z, d) = P(d)P(w|z)P(z|d) \tag{4.1}$$

Furthermore, the probability of an observation pair (w, d) can be obtained by marginalizing out the topic variable in the joint distribution:

$$P(w, d) = \sum_{z=1}^{N_z} P(w, z, d) = P(d) \sum_{z=1}^{N_z} P(z|d)P(w|z) \tag{4.2}$$

A closer look at equation (4.2) reveals that the model decomposes the conditional probabilities of words in a document $p(w|d)$ as a convex combination of the topic specific word distributions $p(w|z)$, where the weights are given by the topic distribution $p(z|d)$ in a document.

4.1.1 Geometric Interpretation and relation to other models

Geometrical Interpretation. The vocabulary of N_w words defines an N_w dimensional space. All multinomial distributions over this vocabulary lie on the $N_w - 1$ dimensional simplex, illustrated in Figure 4.2, where every point sums to 1. The topics as well as the documents lie in the same $N_w - 1$ dimensional simplex. However, the topic distribution $p(z|d)$ lies in the $N_z - 1$ dimensional simplex. By representing a document as a mixture of topics, we seek a lower dimensional ($N_z \ll N_w$), decomposition of the documents.

Matrix Factorization. Due to the lower dimensional view of documents that PLSA offers, as

1. Note that the word banks can also occur in documents that talk about rivers and water bodies, which is an example of polysemy.

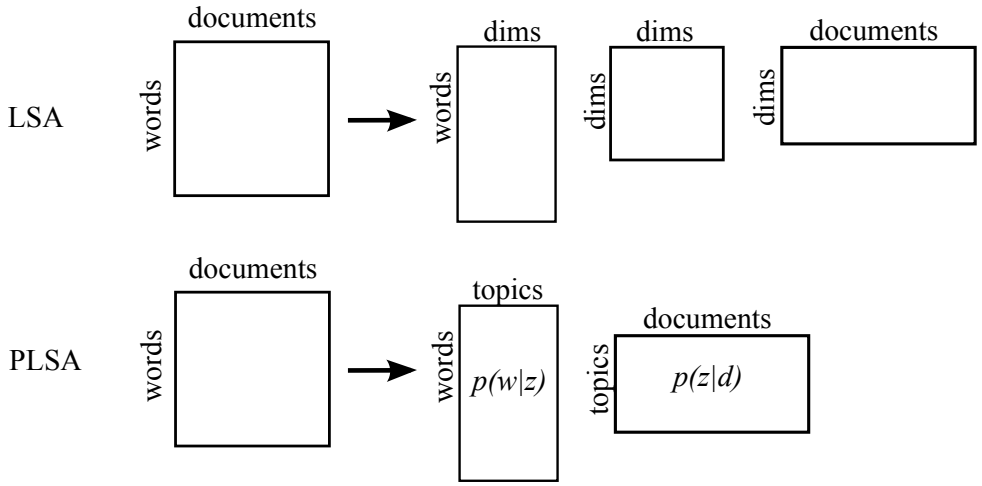


Figure 4.3. Comparing PLSA and LSA as matrix factorization methods. By Hofmann (2001); Steyvers and Griffiths (2007)

pointed out by (Hofmann, 2001), it can be viewed as a dimensionality reduction method too. This compels us to compare PLSA with other matrix factorization and dimensionality reduction methods like Latent Semantic Analysis (LSA) and Non-negative Matric Factorization (NMF). In LSA, a document-word matrix is factorized into two orthogonal matrices: a matrix of document vectors and a matrix of word vectors along with a diagonal matrix of singular values. Figure 4.3 illustrates this decomposition. In case of PLSA, the document-word matrix factorizes into two matrices: a topic distribution matrix $p(w|z)$ and a document distribution matrix $p(z|d)$. The diagonal matrix that we see in LSA can be assumed to be absorbed by either of the matrices. Though conceptually similar, LSA and PLSA differ in many ways. Firstly, the two matrices that PLSA produces contain independent probability distributions that sum to one, whereas in LSA the factorized matrices have orthonormal components, taking any value from the real line $\mathcal{R}$, that are not necessarily positive or summing to one. Having no such constraint on the values creates difficulty in inferring the importance of a word in a topic and the contributions of a topic in a document. NMF (Lee and Seung, 2001) addresses this issue by constraining the vectors to have non-negative values but they still need not sum to one [2].

PLSA, by providing distributions that sum to one, allows us to interpret the importance of words and topics. Furthermore, this sense of proportion and non-exclusiveness of words (*i.e.,* words can participate in multiple topics), helps us to disambiguate multiple meanings of the same word (polysemous) and multiple words meaning the same thing (synonymous) using their weights and the context.

Latent Dirichlet Allocation. The generative model of PLSA is not complete, *i.e.,* it does not say how the topic weights $P(z|d)$ are generated, making it difficult to generalize for unseen documents. Blei *et al.* (2003a) solved this by introducing a Dirichlet prior distribution $\mathrm{Dir}(\alpha)$ on the topic weights, $\theta_d = P(z|d)$ and named it Latent Dirichlet Allocation (see Figure 4.1(b)). The Dirichlet dis-

2. There are also works that have compared different matrix factorization methods Gaussier and Goutte (2005); Girolami and Kabán (2003).

tribution being the conjugate prior for multinomial distributions, makes it an appropriate choice for the prior, simplifying the mathematical complexity in the inference stage. A K dimensional random variable θ is said to follow a Dirichlet distribution parameterized by $\vec{\alpha}$ if:

$$P(\theta|\vec{\alpha}) = \frac{\Gamma(\sum_{i=1}^{K}\alpha_i)}{\prod_{i=1}^{K}\Gamma(\alpha_i)}\prod_{i=1}^{K}\theta_i^{\alpha_i-1} \tag{4.3}$$

where Γ denotes the Gamma distribution and, $0 \leq \theta_i \leq 1, \forall i$ and $\sum_i \theta_i = 1$. Note that $\frac{\vec{\alpha}}{\|\vec{\alpha}\|_1}$ represents the expected values of the parameter θ (where $\|\vec{\alpha}\|_1$ is the L1 norm of $\vec{\alpha}$). When Dirichlet distribution is used as a prior over the parameters θ of a multinomial distribution, $\|\vec{\alpha}\|$ denotes the strength of the prior, and can be viewed as a count of virtual observations distributed according to $\frac{\vec{\alpha}}{\|\vec{\alpha}\|_1}$. Usually, a non informative symmetric Dirichlet prior $\{\alpha_1 = \ldots = \alpha_k\}$ is used. When the strength of the prior is increased, the effect is a smoothed document distribution. Similar to the $\{\theta, \alpha\}$ pairs, Griffiths and Steyvers (2002a,b, 2004) suggested a Dirichlet prior $\mathrm{Dir}(\beta)$ over the multinomial topic distributions $\varphi_z = P(w|z)$ and called it the smoothed LDA.

PTMs vs Clustering. The ability of PTMs in grouping co-occurring words encourages us to compare and contrast them with the conventional clustering methods. PTMs and clustering methods differ mainly on the following aspects: i) clustering methods group data using a distance measure defined on the feature space while grouping by PTMs are induced by feature co-occurrences; ii) choice of the distance measure plays a critical role in determining the shape, size and the quality of clusters, while in PTMs the temporal extent of the document and vocabulary are the key factors; iii) clustering methods can deal with discrete and continuous data, while conventional PTMs such as PLSA and LDA deal with recurring counts of discrete entities. Therefore, by selecting PTMs over clustering methods for activity analysis, we accept a simple but valid assumption that activities leave a trail of co-occurring observations. A more complicated choice on the distance metric is thus avoided.

4.1.2 PLSA Inference

The parameters of PLSA *i.e.,* the distributions $\{p(w|z), p(z|d)\}$ are estimated iteratively using the maximum likelihood principle. More precisely, given a set of training documents $\mathcal{D}$, the log-likelihood of the model parameters Θ can be expressed by:

$$\mathcal{L}(\Theta|\mathcal{D}) = \sum_{d\in\mathcal{D}}\sum_{w} n(w,d)\log\left(P(w|d)\right) \tag{4.4}$$

where, $n(w,d)$ is the word count in documents and $P(w|d)$ is given by the generative model as in equation (4.2). The optimization is conducted using the Expectation-Maximization (EM) algorithm (Hofmann, 2001). The EM procedure starts by initializing the parameters using random values. Then in the Expectation step of equation (4.5) the posterior distribution of the latent variable is calculated as:

$$P(z|w,d) = \frac{P(w,z,d)}{\sum_{z=1}^{N_z} p(w,z,d)} = \frac{P(w|z)P(z|d)}{\sum_{z'=1}^{N_z} P(w|z')P(z'|d)} \tag{4.5}$$

In the Maximization step, by maximizing the Expected likelihood function of PLSA the model parameters are estimated as:

$$P(w|z) \propto \sum_{d=1}^{D} n(w,d)P(z|w,d) \tag{4.6}$$

$$P(z|d) \propto \sum_{w=1}^{N_w} n(w,d)P(z|w,d) \tag{4.7}$$

At test time, we are interested in estimating the weights $P(z|d)$ of the topics for a document d. This is achieved by running the EM algorithm keeping the learned model $P(w|z)$ fixed and maximizing the log likelihood of the words in the document:

$$L_d^u(P(z|d)) = \sum_w n(d,w) \log \left(\sum_z P(z|d)P(w|z) \right) \tag{4.8}$$

The above algorithm for PLSA derives point estimates of the parameters directly and is said to suffer from problems involving local maxima of the likelihood function. In the case of LDA, there are several exact and approximate inference methods to estimate the parameters. Mean field approximation (Buntine, 2002), Variational inference (Blei *et al.*, 2003a), Expectation propagation (Minka and Lafferty, 2002) and Markov Chain Monte Carlo (MCMC) methods (Griffiths and Steyvers, 2004) are some of them.

To discover activities from videos, we experimented with both PLSA and LDA models. PLSA parameters were learned using equations (4.5–4.7) and the LDA parameters were learned using the C implementation provided at (Blei *et al.*, 2003b). Since we have no *a priori* knowledge about which topics are more probable in a document or which words are more probable in a topic, we chose symmetric (non-informative) hyper priors for LDA, *i.e.,* we set $\alpha = 0.1$ and $\beta = 0.1$. With this setting of LDA, we obtained topics that were qualitatively similar to the ones obtained from PLSA. To avoid a cumbersome process of selecting appropriate priors for the LDA model and to save on computational time, we chose the simpler PLSA implementation in all our further analysis in this chapter.

4.2 Activity patterns and scene segmentation

To illustrate the use of the PLSA model on activity analysis, we consider three datasets: The Traffic Junction video, Far-field video and MIT video described in section 3.1 and reminded here in Figures 4.4(a–c). To apply PLSA, we need to define our semantic space; that is, the vocabulary set and the input documents.

Vocabulary. Our vocabulary could be defined as a Cartesian product of the location, motion, and

(a) Traffic Junction (b) Far field (c) MIT data

Figure 4.4. Datasets used for activity analysis using PLSA

size word spaces. Consider for an image size of 280x360, with location quantized into 10x10 bins, flow features quantized into 5 bins and with 2 size words, we would get $28 \times 36 \times 5 \times 2 = 10080$ words. However, while knowing the joint feature (motion, size) for each location might be desirable (for instance to distinguish between cars and people on zebra crossings), this results in a high dimensional vocabulary. As PLSA models word co-occurrences across documents, we expect that topics will capture separately people activity or car activity at a given location since they don't occur simultaneously. In other words, *given* an activity and location, we expect the motion and size to be independent, and thus we can simply concatenate them and define the set of words for a cell c, denoted by V_c, to be the concatenation of the motion and size words,[3] leading to a codebook of $28 \times 36 \times (5+2) = 7056$ words instead of 10080 words. Thus, a word can be denoted by $w_{c,m}$, where c is the location and m are the motion and size characteristic labels. In this chapter, we show results with both 10x10 location grids as well as 4x4 location grids. Similarly, we used 5 and 9 bins to quantize optical flow words.

Documents. Documents are represented as a simple bag-of-words. That is, the video is first divided into short term clips. Then the word counts $n(w, d)$ for the document d are obtained by counting the number of occurrences of a word w in the document. The documents are usually created from overlapping clips. The length of the clip considered for creating documents is an important parameter in PLSA like models. Shorter clips (usually of 1 sec duration) result in spatially localized activities, while longer clips give patterns spanning large spatial extent. For our experiments in this chapter, documents were created using overlapping clips of 5 seconds duration.

We present some topics that were discovered by the approach and show how it can be used to identify activities related to different object sizes or to segment the scene into different semantic regions.

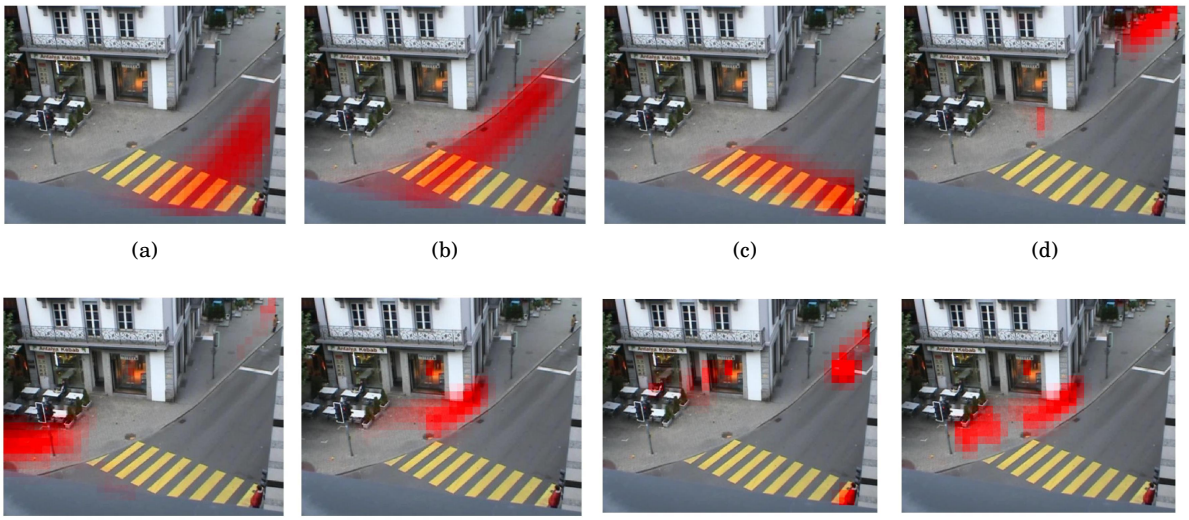
4.2.1 Activity patterns

An activity like a vehicle moving on the road can be described by a set of motion and size features co-occurring over a sequence of locations. Similarly, a pedestrian standing at the foot path can

3. This means that when constructing documents, a pixel will provide two words for the cell it belongs to: a motion word and a size word.

(a) m = right (b) m = left (c) m = up (d) m = down (e) m = static (f) m = small (g) m = big

Figure 4.5. Location activations by word type in a PLSA topic. Given a topic that represents vehicle moving towards top right, the images show the locations where each word type m = right,left,up,down,small,big is active within a PLSA topic.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 4.6. Examples of common activity patterns. Vocabulary created from 10x10 location qunatization, 5 motion bins and 2 size words. (a–b) vehicles passing, (c) pedestrians crossing the road, (d–f) topics involving small objects - pedestrians walking on the foot path, (g–h) the first two topics involving static pixels, (g) partially occluded vehicle waiting for signal (top right) with pedestrians waiting for signal at the bottom-right, (h) pedestrians waiting at the footpath for crossing the road.

be described by a co-occurring set of static pixels and size features. Thus, each activity pattern or a topic is a strongly co-occurring set of visual features represented by $P(w|z)$. To identify the set of locations which are mainly active for a given topic, one way is to look at locations that are active (non-zero value) for each motion and size word given a topic. A example of this for a topic representing "vehicles moving towards top right" is given in Figure 4.5. For this topic, we see that word types: m = *"right"* and m = *"big size"* are active and that the rest of them have insignificant activations.

Another compact way is to marginalize the word distribution with respect to the words that occur at the same location. That is, we can plot the map defined for each cell c by: $P(activity \in c|z) = \sum_{w \in V_c} P(w|z)$. This leads to the concatenation of the location activations from all the word types. For example, Figure 4.6(a) is a concatenation of the Figs. 4.5(a–g). Figure 4.6(a)-(f) show the activity locations of selected topics highlighted in red.

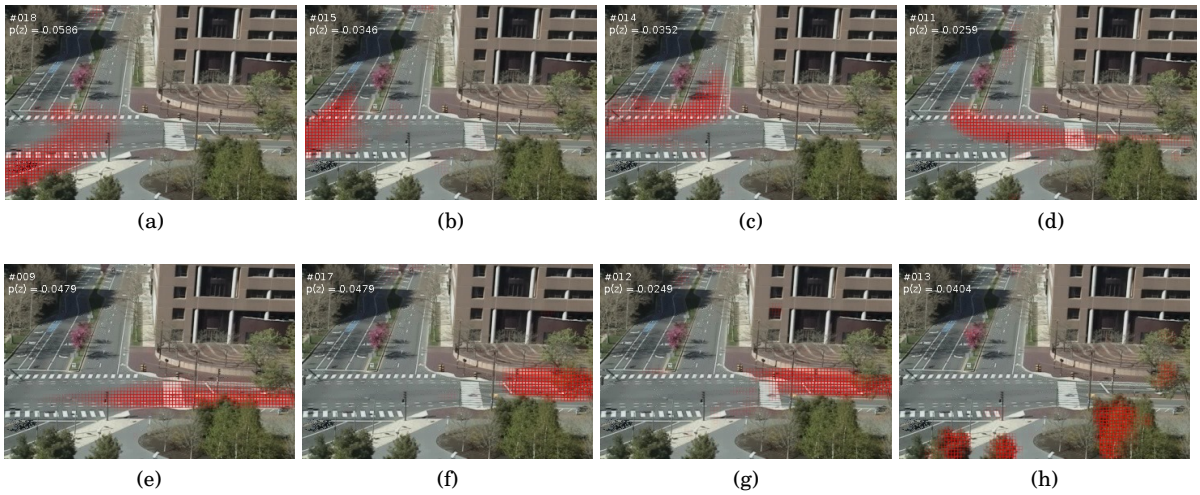


Figure 4.7. PLSA activity patterns from MIT data, Vocabulary created from 4x4 location qunatization and 9 motion bins. The topics in (a–e) represent vehicles moving in different directions. Topics in (f–g) represent activities if vehicles waiting for the signal. Activity generated by trees moving due to wind is seen in (h).

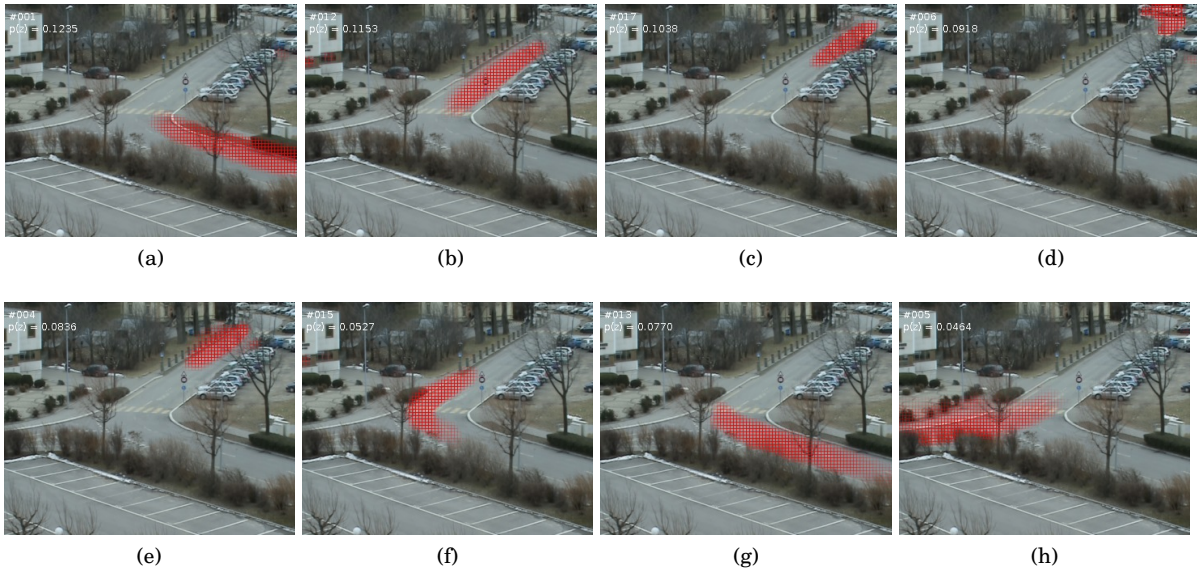


Figure 4.8. PLSA activity patterns from Far-field data. Vocabulary created from 4x4 location qunatization and 9 motion bins. Patterns that form complete object trajectories.(a–d) shows a vehicle moving from bottom right, taking a turn and moving towards top right. (e–h) vehicles moving in the exact opposite direction of (a–d).

Understanding topics by features. We can identify which of the extracted topics are more related to the activities of objects of small or large sizes by ranking the aspect according to the size probability obtained by marginalizing over the word 'small size' of every cell, *i.e.,* by computing $P(size = small|z) = \sum_{w_{c,m}/m=small} P(w|z)$. For instance, Figure 4.6(d,e,f) show the top ranking topics from 10 topics involving small objects that correspond to pedestrians walking on the side-walk.

A similar analysis can be done with static objects, and corresponding topics indicate pedestrians waiting to cross the road and cars waiting at the traffic light (Figure 4.6(g,h)). Interestingly, note that the model was able to discover that during several parts of the junction traffic cycle, both pedestrians (bottom right) and vehicles (top right) needed to wait simultaneously.

We could obtain topics of similar semantic content from other datasets too. These topics are generated by using 4x4 grids for location words and 9 bins for motion, a finer quantization compared to the images in Figure 4.7. In Figure 4.7–4.8, topics obtained from MIT and Far-Field data respectively are shown. In the case of MIT data, Figure 4.7(a–e) show vehicles moving in different directions. For example, in Figure 4.7(a–b) vehicles moving top to bottom of the scene and vice-versa. In Figure 4.7(c), vehicles come from the left side and taking a left turn, move towards the top of the scene. In Figure 4.7(d), vehicles move from the top to the right. Two static topics corresponding to vehicles waiting for the signal are shown in Figure 4.7(f–g) and interestingly, some tree movements are captured in Figure 4.7(h).

In Far-Field data (Figure 4.8), since there are not many static events, we obtain mostly topics that correspond to moving vehicles. The example in Figure 4.8 shows various topics that span the entire trajectory of a vehicle when it appears from either at the bottom right Figure 4.8(a–d) or from the top right in Figure 4.8(e–h).

4.2.2 Scene segmentation

Another way to investigate the learned topic is to segment the scene according to the extracted activities. Knowledge of the semantic scene regions could then provide context to the actions and thus help in understanding the intent of actions in a scene location. For example, in a typical traffic scene as in Figure 4.4, activities like pedestrians walking along the pavement or waiting at the zebra crossings are seen on the pedestrian side while vehicular movements are (in principle) only seen on the roads. An activity based segmentation achieves this by grouping parts of the scene into segments such that each segment corresponds to locations where similar semantic activities take place.

Approach. Scene segmentation works on a representation of each pixel in the scene. Li *et al.* (2008) represents each pixel using quantized spatio-temporal words that are observed at this location in the training data. We propose an alternative method to characterize a location by the set of topics or *activities* that can occur at this pixel. This should lead to a less noisy representation, and implicitly incorporate temporal information as the activities model *observations which co-occur*, unlike raw feature distributions (Li *et al.*, 2008). Activities at the cell location c are represented by the topic distribution at this cell, denoted $P(z|c)$ and defined as: $P(z|c) = P(z|V_c) \propto P(V_c|z)P(z) = \sum_{w \in V_c} P(w|z)P(z)$. In practice, we expect these distributions to smoothly evolve when the location c moves along semantically similar regions (*e.g.,* while moving along the same side of the road), and change abruptly when the location moves across some semantic border (*e.g.,* moving from the road zone to the sidewalk region). Thus, clusters mainly correspond to smooth manifolds which can not be well represented using metric based clustering approaches like K-means. We used a spectral

clustering algorithm by Ng *et al.* (2001), which has been shown to better capture such manifolds. It takes an affinity matrix A as input, which is given by:

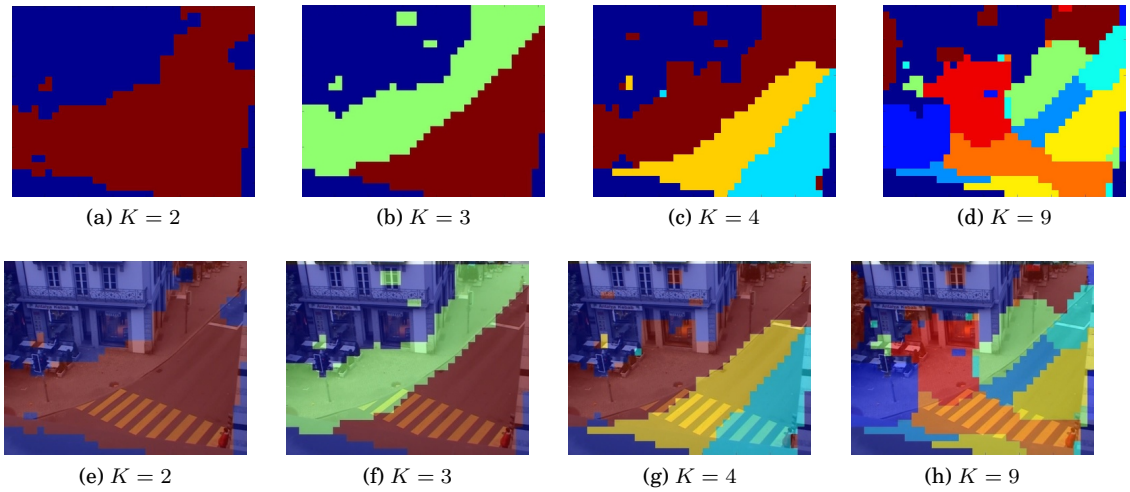
$$A_{c_i,c_j} = exp(\frac{-D^2_{Bhat}(P(z|c_i), P(z|c_j))}{2\sigma^2}) \tag{4.9}$$

where D_{Bhat} denotes the Bhattacharyya distance used to compute the pairwise similarity between the two activity distributions at cell c_i and c_j, and is defined by:

$$D_{Bhat}(P, Q) = \sqrt{1 - \sum_{x \in X} \sqrt{P(x).Q(x)}}. \tag{4.10}$$

The scale σ is taken to be the value that gives minimum cluster distortion (Ng *et al.*, 2001).

Results. Figure 4.9 illustrates the results from Traffic Junction data obtained by applying the



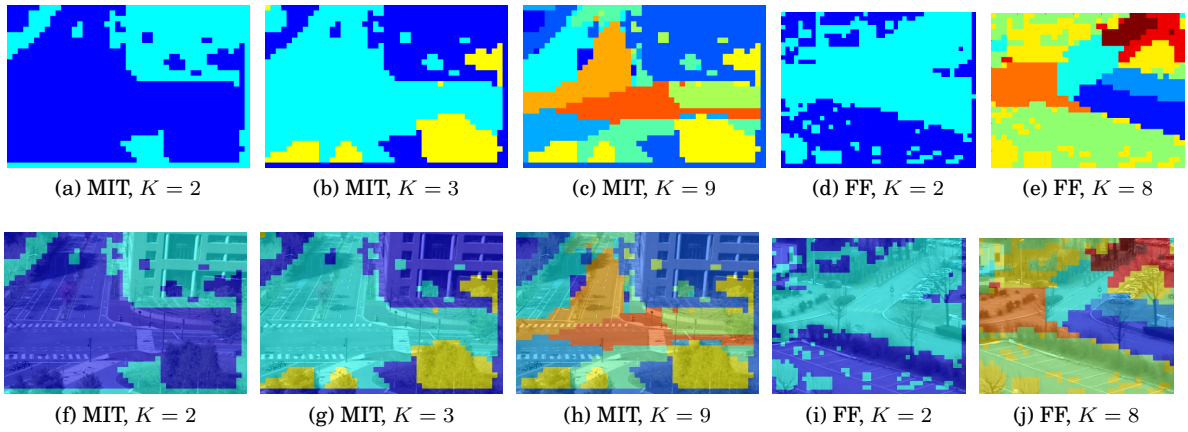
(a) $K = 2$ (b) $K = 3$ (c) $K = 4$ (d) $K = 9$

(e) $K = 2$ (f) $K = 3$ (g) $K = 4$ (h) $K = 9$

Figure 4.9. Semantic Scene Segmentation obtained from 10 PLSA Topics, with the segments on top and the scene below with the clusters superimposed. (a,e) 2 clusters (b,f) 3 clusters (c,g) 4 clusters, (d,h) 9 clusters

spectral clustering algorithm with 2, 3, 4 and 9 clusters respectively, while the number of topics extracted with PLSA was 10. As it can be seen, the results reveal that the number of clusters correspond to different levels of details in interpreting the semantic activities in the scene. When $K = 2$, the algorithm segments the scene into regions of with activity and no activity respectively [4]. When $K = 3$, the activity region is further divided into the pedestrian and vehicle regions. When $K = 4$, the road is split into the different sides of the road. When $K = 9$, further semantic regions like the region corresponding to zebra crossing, where both car and pedestrian motion can occur, or the different regions from where people come to cross the road (and wait) appear. To further demonstrate the methods applicability, we also show scene segments obtained from MIT and Far-field data in Figure 4.10. Figure 4.10 has 2, 3 and 9 segments obtained from 20 topics of MIT

4. Note that the segmentation algorithm does not impose any geometrical or location constraint to obtain connected segments.



(a) MIT, $K = 2$ (b) MIT, $K = 3$ (c) MIT, $K = 9$ (d) FF, $K = 2$ (e) FF, $K = 8$

(f) MIT, $K = 2$ (g) MIT, $K = 3$ (h) MIT, $K = 9$ (i) FF, $K = 2$ (j) FF, $K = 8$

Figure 4.10. Semantic Scene Segmentation obtained from 10 PLSA Topics on MIT and Far-field data. with the segments and their superimposition on the scene. (a,b,c,f,g,h) show segments from MIT data: (a,f) $K = 2$, (b,g) $K = 3$ and (c,h) $K = 9$. (d,e,i,j) show from Far-field data: (d,i) $K = 2$ and (e,j) $K = 8$.

followed by 2 and 8 segments from 20 topics of Far-Field data. When $K = 2$, one segment represents regions of activity and the other represents the background region in both MIT and Far-field data. When K is increased in the MIT case, finer regions corresponding to different motion directions emerge. Interestingly, in Figure 4.10(b,g,c,h), the segment in yellow corresponding to moving trees emerges consistently. In the Far-field case, as K is increased, we see that the lanes of the road that correspond to different motion directions get separated. Here again in Figure 4.10(d,i,e,j), the regions in cyan and yellow are caused by trees moving in the wind. Thus, we see that there is not a single valid value for K, but that each value leads to a scene segmentation with clear semantic interpretation.

4.3 Abnormality detection

There are several methods proposed in the literature that capture abnormal events. In this section, we present those that are pertaining to the topic model that we are using, and evaluate their performance on the Traffic Junction dataset.

4.3.1 Abnormality measures

Modeling using a generative approach gives scope for a variety of measures to identify unusual patterns in the data. But little study has been done in comparing the different measures on the same task. Here, we present various possible measures that can be used based on the approach we consider, and evaluate the measures within the proposed framework to understand their merits and demerits.

Fitting measures. The estimation of the topic distribution $P(z|d)$ of a given clip is obtained by

optimizing the log-likelihood function of equation (4.8). Thus, one natural way to consider if a clip is normal or abnormal is to use this log-likelihood measure $L_d^u(P(z|d))$ at the end of the fitting phase. If the activities happening within the clip correspond to those observed in the training dataset, then the fitting will be able to find a suitable topic distribution explaining the bag-of-word representation of the clip. Thus, normal clips will generally provide high log-likelihood. On the other hand, if an abnormal activity is going on, none of the learned topics will be able to explain the observed words of that activity, resulting in a low likelihood fit. The likelihood expression in equation (4.8) suffers from a severe drawback: it is not normalized and thus, whatever the quality of the fit, the measure is highly correlated with the document size. To solve this issue, we can exploit the average log-likelihood of each word, by dividing $n(w,d)$ by the number of words $n_d = \sum_w n(d,w)$ in equation (4.8), and get the normalized log-likelihood measure:

$$L_d^{nl}(P(z|d)) = \sum_w \frac{n(d,w)}{n_d} \log \sum_z P(z|d)P(w|z) = \sum_w P_o(w|d) \log P_c(w|d) \tag{4.11}$$

where $P_o(w|d) = \frac{n(d,w)}{n_d}$ is called the objective distribution as it is measured directly from the test document, and $P_c(w|d) = \sum_z P(z|d)P(w|z)$ is called the constrained distribution as it lies in the constrained simplex spanned by the topic distribution $P(w|z)$ (cf Figure 4.2).

Distribution reconstruction errors. The goal of optimizing the likelihood function is to fit the constrained distribution to the objective distribution. Thus, one possibility to evaluate the quality of the fitting is to measure the discrepancy between the two distributions. For instance, we could use the Kullback-Leibler divergence to estimate this discrepancy which leads to:

$$\begin{aligned} L_d^{KL}(P(z|d)) &= KL[P_o(w|d)|P_c(w|d)] \\ &= -\sum_w P_o(w|d) \log P_c(w|d) + \sum_w P_o(w|d) \log P_o(w|d) \\ &= -L_d^{nl}(P(z|d)) - H(P_o(w|d)) \end{aligned} \tag{4.12}$$

where $H(P_o(w|d))$ is the entropy of document d, which is a constant specific to each document. From this expression we note that the topic distribution $P(z|d)$ which maximizes the likelihood expression in equation (4.8) is actually the one that minimizes the KL divergence L^{KL}. We can thus interpret the fitting as a document reconstruction process where the error in reconstruction is given by equation (4.12). This interpretation permits us to use a variety of measures capable of estimating the discrepancy between the distributions. The Bhattacharyya distance given by equation (4.10) is one such valid candidate to compare P_o and P_c. Therefore we define the Bhattacharyya error measure as,

$$L_d^{Bh}(P(z|d)) = D_{Bhat}\left(P_o(w|d), P_c(w|d)\right) \tag{4.13}$$

4.3.2 Results and discussion

We first trained a model using 2210 clips containing normal clips. For testing the different measures on the Traffic Juction video, we labeled 320 clips different from the training set containing 140 normal activities, and 180 video clips with abnormal events, where abnormality is defined as: people crossing the road at the wrong place (far away from zebra crossing), vehicle parked at the pedestrian path, or vehicles stopping ahead of the stop line while the stop sign is red. In the following experiments, unless stated otherwise, 20 topics were used to model the scene activities.

Qualitative illustration. The abnormality measures that we have defined allowed us to identify multiple instances of several abnormal events occurring both in isolation or simultaneously with other normal activities in the image. Figure 4.11 shows the first video clips that were retrieved as abnormal using the normalized log-likelihood L^{nl} [5] measure. The object causing the abnormality is marked with a red box for identification. Figure 4.11(a,d) shows the event where a car is parked in the pedestrian foot path. In (d), additionally a pedestrian crosses the road in the wrong place. In Figure 4.11(b,e) a car stops ahead of the stop line, and this stop is not due to stopped cars in front of it. This is a rare occurrence that happened because the car could not cross the road before the red light went on and hence had to stop to let the vehicles on the other directions pass. In Figure 4.11(c,f) pedestrians were crossing the road away from the foot path.

Quantitative evaluation. Precision-Recall (PR) curves were considered to quantitatively assess the performance of the approach and compare the abnormality measures. Figure 4.12(a) shows the PR curves for the Likelihood, Normalized Likelihood, KL-Divergence and Bhattacharyya distance based abnormality measures. We first note that unnormalized likelihood measure does not achieve a good performance. The reconstruction error measure obtained from KL-divergence and Bhattacharya distance shows better performance, but still not as good as the normalized likelihood measure, which achieves the best performance with good detection rates (with a precision of almost 1 for a recall of 50%).

Document size normalization. An analysis of the detection errors made using the likelihood measure, the KL-divergence, and the Bhattacharyya distance, reveal that they are affected by document size or entropy. This is illustrated in Figure 4.12(b), where we plot the Bhattacharya distance error measure as a function of the document size. We can observe from Figure 4.12(b) that smaller documents (with low entropy) tend to have higher reconstruction error, while larger documents (with high entropy) tend to have lower error. The normalized log-likelihood measure directly alleviates this effect by using the average word log-likelihood as abnormality measure. The KL-divergence measure can be normalized by removing the document specific entropy term. When this is done, we are left with the cross entropy term $H(P_o|P_c)$ given by:

$$H(P_o|P_c) = \sum_w P_o(w|d) \log P_c(w|d) \tag{4.14}$$

5. The Adaptive Bhattacharyya measure that we will describe below produced the same results.

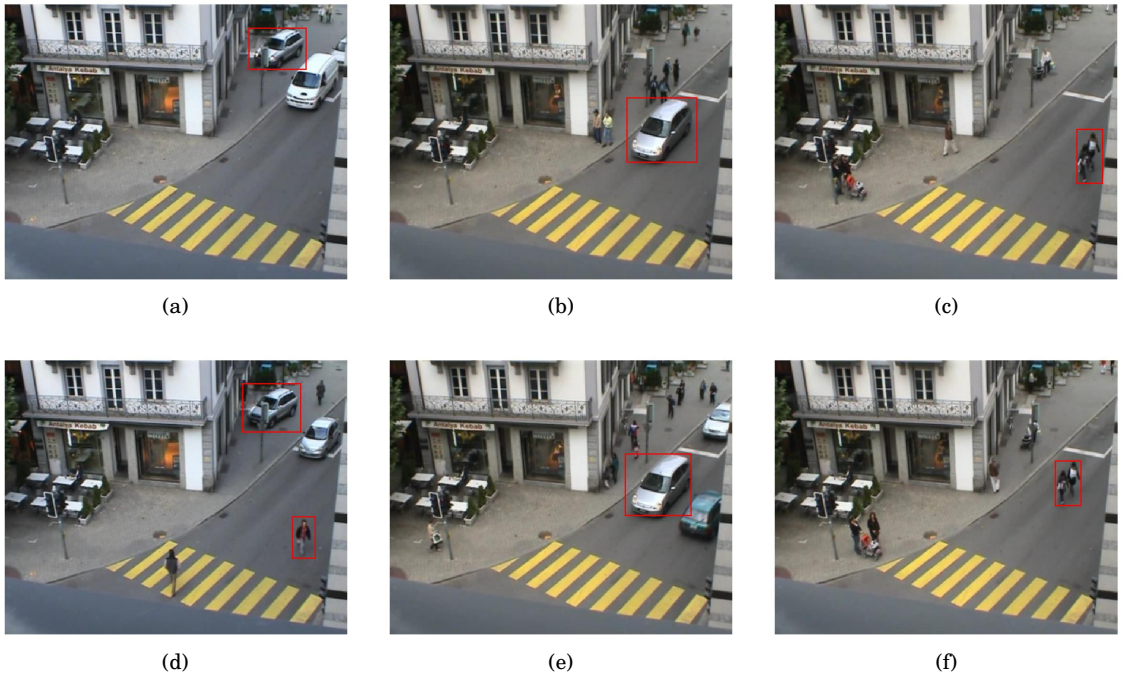


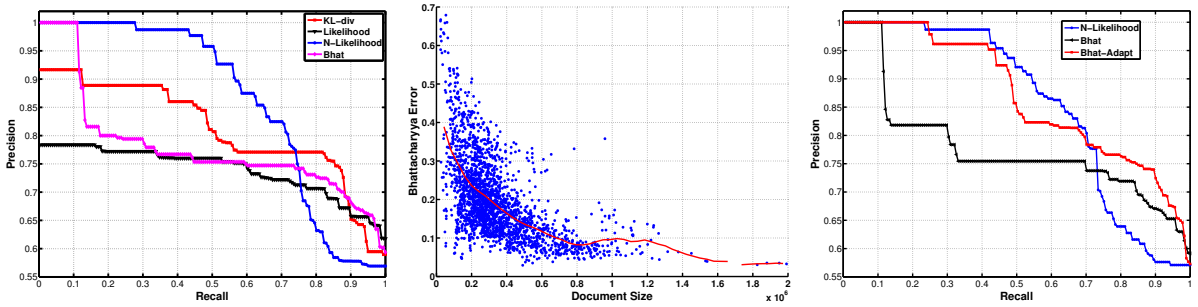
Figure 4.11. The top abnormal events retrieved using the the Adaptive Bhattacharyya measure, equation (4.15). Note that, for illustration purposes, several abnormal documents corresponding to the same already displayed events have been omitted. (a,d) shows the event where a car is parked in the pedestrian foot path. (d) pedestrian crossing the road in the wrong place, (b,e) a car stopping ahead of the stop line. (c,f) pedestrian crossing the road away from the foot path.

which is simply the normalized log-likelihood measure.

In the case of Bhattacharyya distance, such a direct normalization is not possible. Therefore we treat this bias by performing an adaptive normalization based on document size which can be learned from the training data. For this, we construct a histogram of document size in the training set, and calculate for each bin the expected error measure for documents belonging to that bin (please see the red curve in Figure 4.12(b)). Then, for a test document, its reconstruction error using Bhattacharyya distance is normalized with the expected error according to its size before being compared with the abnormality threshold. In other words, we consider the measure,

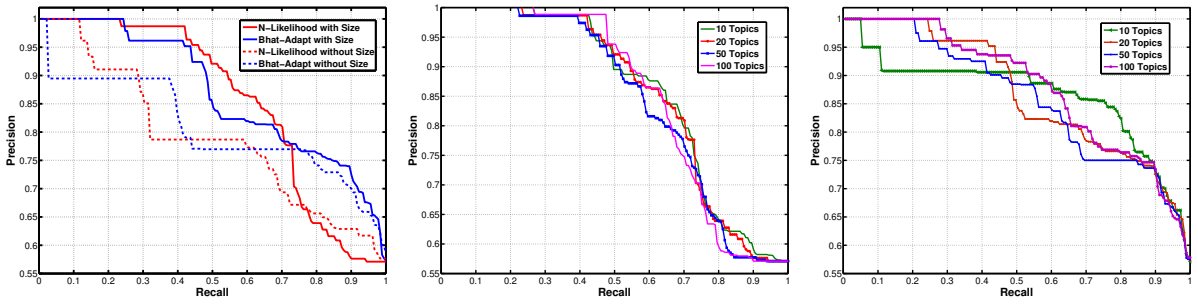
$$L_d^{Bh-ad}(P(z|d)) = L_d^{Bh}(P(z|d))/L_d^{Bh}(n_d) \tag{4.15}$$

as our normalized or *Adaptive* Bhattacharyya abnormality measure. Figure 4.12(c) shows the results obtained after removal of the document size bias. As it can be seen, this leads to a considerable improvement, and the Adaptive Bhattacharyya measure performs now as good as the normalized log-likelihood measure, although with a different behavior. While the latter one performs better at medium recall, the Adaptive-Bhattacharyya measure succeeds to keep a precision significantly



(a) (b) (c)

Figure 4.12. (a) Precision-Recall(PR) curves for Likelihood: equation (4.8), Normalized Likelihood: equation (4.11), KL-Divergence: equation (4.12) and Bhattacharyya distance: equation (4.13). (b) Bhattacharyya abnormality error measure vs Document Size: Scatter plot showing the relation between the Document size (number of words) and Bhattacharyya distance abnormality measure. The superimposed curve in red shows the expected Bhattacharyya error for a given size computed from the training data. (c) PR curves for Normalized Likelihood, Bhattacharyya distance and Adaptive Bhattacharyya measure: equation (4.13,4.15).



(a) (b) (c)

Figure 4.13. (a) PR curves for Normalized Likelihood: equation (4.11), Adaptive Bhattacharyya measure: equation (4.13,4.15), with and without using the size words. (b) Effect of varying the number of topics: PR curves for Normalized Likelihood (left), and (c) Adaptive Bhattacharyya measure (right) when using 10, 20, 50 and 100 Topics.

higher than random for very high recall.

Video Features. We also evaluate the effect of adding the size words in our description of activities, as compared to using just optical flow words as used by Wang *et al.* (2008b). This is shown in Figure 4.13(a), where the normalized log-likelihood and Adaptive Bhattacharyya measure abnormality PR curves are plotted with and without object size words. These curves show that the detection rates improve significantly when object size words are used as compared to just optical flow words. The size features help in distinguishing activities due to pedestrians vs vehicles, this helps in identifying some abnormalities like pedestrians crossing the road at the wrong place characterized by a small sized object in a location with predominantly vehicles.

Number of topics. Finally, Figure 4.13(b–c) plots the PR curves for our two best measures when 10, 20, 50 and 100 topics were used (20 topics were used in the other curves) to model the different scene activities. As it can be seen, the number of topics does not affect the results significantly.

This is particularly true for the normalized likelihood measure. In case of Adaptive Bhattacharyya measure, we observe this as we increase the number of topics beyond 20.

4.4 Summary

In this chapter, we first introduced the topic model framework and explained the parameter estimation procedure. This will be a precursor to the methods to be discussed in future chapters. In our initial application of PLSA for activity analysis task, the activity patterns discovered in a strictly unsupervised setting with simple features, produced convincing results. The patterns obtained could also be used to segment the scene and detect abnormal events. However, there are some limitations. The PLSA or the LDA models ignore the ordering of words in a video clip resulting in topics without any temporal order. It is also difficult to infer scene level rules by directly using PLSA topics. In the following chapters, we will address these issues by proposing a more powerful approach that incorporates temporal information in the model.

Chapter 5

Probabilistic Latent Sequential Motifs

Temporal order is an intrinsic nature of most human activities. Consider an online sensor monitoring mouse clicks and their locations from browsing activity of individuals. Mining this sensor log, one may discover recurrent sequential patterns of mouse clicks such as compose-send, compose-spellcheck-send, reply-send or even fwd-send clicks. Extending this to physical human activities like walking or running, we see that they also adhere to a particular temporal order in moving the limbs. Therefore, it is essential to discover activities with their temporal structures, for benefits of better understanding and completeness in representation.

But in reality recovering such sequential patterns is made difficult due to one important property of the observations. In most real-life scenarios, multiple activities occur simultaneously leading to a mixed observation over time. To understand this problem, consider that we have two sensors recording water and electricity usage in an apartment. We could expect to spot activity patterns (motifs) like a short duration-high water consumption followed by short duration-high electric consumption (filling a kettle and then starting a boiler). We could also expect patterns like one hour long alternating water and electric consumption (due to a washing machine). Due to the presence of multiple persons, multiple occurrences of these two motifs can occur at the same time and with no particular synchronization. Note that the two motifs share the same basic actions but differ only in the order of execution. Similar examples can be found in many other domains such as spatio-temporal brain activity modeling and weather pattern mining to name a few.

In this chapter we propose a novel graphical model called Probabilistic Latent Sequential Motifs (PLSM), that discovers sequential patterns called motifs from sensor logs of the nature described above. Our particular interest is to discover such motifs from surveillance video logs, where the motifs represent dominant activities in the scene. In the previous chapter, we saw how the PLSA topic model could be applied to mine activities from video scenes. One important limitation of this approach is the bag of words assumption which leads to activities with no particular temporal struc-

ture. Modeling word dynamics within topics in the presence of multiple activities can be achieved if we can idenitify each instance of the activities and their observations respectively from the rest of the scene observations. But unlike object trajectories, pixel level features lose the object identity and observations from individual activities cannot be separated easily. The model introduced in this chapter solves this problem of discovering motifs and identifying their precise time of occurrences. To constrain the model so that we obtain sparse and peaky activity starting time distributions (almost resembling an a-periodic dirac comb function) the model inference also incorporates a novel sparsity constraint based on KL-divergence.

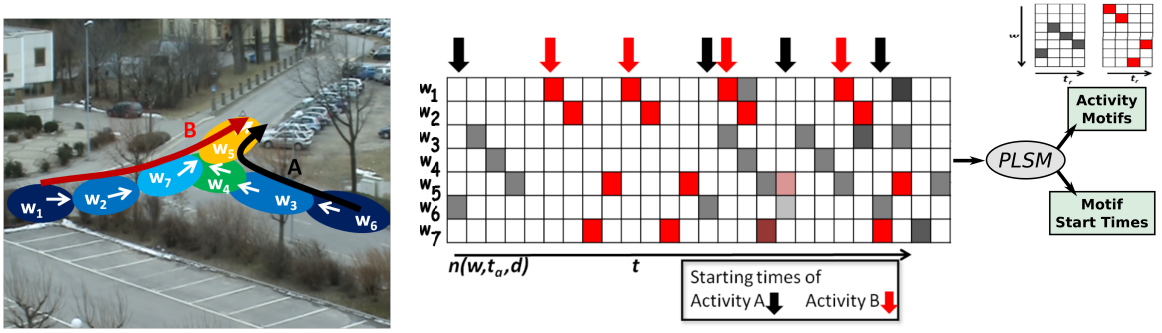
The rest of this chapter is organized as follows. We will first provide an intuition of the PLSM model, its application to video activity analysis, its input and outputs in the beginning of section 5.1. Then, we will introduce the PLSM model with details, including the inference procedure. Experiments on synthetic data are first conducted in section 5.3 to effectively demonstrate various aspects of the model. Application of the PLSM model to extract recurring activities from surveillance videos is explained in section 5.4, which includes experiments on three different video datasets from state of the art papers and three video datasets of single and multi-camera views from crowded metro stations. The captured PLSM motifs are shown and discussed in section 5.5, with quantitative experiments on an activity prediction task and on a comparison with ground truth labeled data. The generality of the PLSM method is further demonstrated in section 5.6, which presents its application to audio traffic localization data captured by microphone array sensors. Finally, section 5.7 concludes the chapter with a discussion and a summary.

5.1 Probabilistic Latent Sequential Motif Model

Before formalizing our approach, we illustrate the activity discovery algorithm on a simple video case, as shown in Figure 5.1. Assume that in this scene, only two activities can occur, and that we have a vocabulary of $N_w = 7$ words, where each word characterizes the motion activity happening in some local regions (the colored blobs in Figure 5.1(a)). The method to automatically define these scene specific words is described in section 5.4.1. The main idea of the PLSM model illustrated in Figure 5.1 is that each occurrence of a scene activity leaves a noisy and variable trace in the word$\times$time count matrix. The count matrix in Figure 5.1(b) shows a simple case with observations from multiple occurrences of the two activities, making clear that activities can overlap in time, share the same vocabulary, occur without any particular synchronization, and are often accompanied with noise. Our goal in this difficult scenario is thus to recover the latent structure by learning the activity temporal patterns called motifs and their time of occurrence as illustrated in Figure 5.1(b).

5.1.1 Notation and model overview

In this section we introduce our notation, a more detailed description of the model's generative process, and the EM steps derived to infer the model parameters, including the handling of sparsity,



(a) Words from two activities shown as colored blobs.

(b) A temporal document d defined as a word×time count matrix $n(w, t_a, d)$, and on the right, the result of the PLSM algorithm (discovered activity motifs and their starting time).

Figure 5.1. Applying PLSM to discover activities from videos. (a) Assume that the scene contains only two activities (Activity A - black and Activity B -red), and that we are able to automatically extract seven low-level activity words labeled $\{w_1, \ldots, w_7\}$ depicted with colored blobs in the image. (b) Each activity occurrence leaves a (noisy) observation trail in a word×time count matrix according to a specific temporal pattern. For instance, in a simple case, activity A could be specified by the particular sequence of word $[w_6, w_3, w_4, w_5]$. Note that these trails can share vocabularies and can be interleaved, *i.e.,* have temporal overlap. The goal of the PLSM algorithm is thus to identify the latent structure characterized by the activity motifs and their start times from the observed count matrix $n(w, t_a, d)$.

the exploitation of priors, and the model selection.

Figure 5.2(a) formally illustrates the process of generating a frequency matrix $n(w, t_a, d)$, which we call a temporal document. A qualitative description of this process is given in Figure 5.1(b). Let D be the number of temporal documents in the corpus, indexed by d. Let $V = \{w_i\}_{i=1}^{N_w}$ be the vocabulary of words that can occur at any given instant $t_a \in [1, \ldots, T_d]$, where N_w is the size of our vocabulary and T_d is the number of discrete time steps of temporal documents d (and thus represents the duration of the temporal documents). A temporal document is then described by its count matrix $n(w, t_a, d)$ indicating the number of times a word w occurs at the absolute time t_a within the temporal document d; a temporal document d thus contains $N_d = \sum_{w,t_a} n(w, t_a, d)$ words in total. According to our model, these temporal documents are generated from a set of N_z (N_z is the number of motifs) motifs $\{z_i\}_{i=1}^{N_z}$ represented by temporal patterns $P(w, t_r|z)$ with a fixed maximal duration of T_z time steps (*i.e.,* $t_r \in [0, \ldots, T_z - 1]$), where t_r denotes the relative time at which a word occurs within a motif. A motif can occur and start at any time instant $t_s \in [1, \ldots, T_{ds}]$ within the temporal document [1]. In other words, qualitatively, temporal documents are generated by taking the motifs and reproducing them in a probabilistic way (through sampling) at their starting positions within the temporal document, as illustrated in Figure 5.1(b) and Figure 5.2(a).

1. The starting time t_s can range over different intervals, depending on hypotheses. In the experiments, we assumed that all words generated by a motif starting at time t_s occur within a temporal document; hence t_s takes values between 1 and T_{ds}, where $T_{ds} = T_d - T_z + 1$. However, we can also assume that motifs are partially observed (beginning or end are missing). In this case t_s ranges between $2 - T_z$ and T_d.

5.1.2 Generative Process

Our data $\mathcal{D}$ is the matrix $n(w, t_a, d)$ containing counts of triplets of the form (w, t_a, d). The actual process to generate these triplets (w, t_a, d) is given by the graphical model depicted in Figure 5.2(b) and works as follows:

Algorithm 2 The PLSM generative model

- draw a temporal document d with probability $P(d)$.

- draw a latent motif $z \sim P(z|d)$; {*where* $P(z|d)$ *denotes the probability that a word in temporal document* d *originates from motif* z.}

- draw the starting time $t_s \sim P(t_s|z, d)$. {*where* $P(t_s|z, d)$ *denotes the probability that the motif* z *starts at time* t_s *within the temporal document* d.}

- draw a word and relative time pair $(w, t_r) \sim P(w, t_r|z)$. {*where* $P(w, t_r|z)$ *denotes the joint probability that a word* w *occurs at time* t_r *within the motif* z. *Note that since* $P(w, t_r|z) = P(t_r|z)P(w|t_r, z)$*, this draw can also be done by first sampling the relative time from* $P(t_r|z)$ *and then the word from* $P(w|t_r, z)$*, as implied by the graphical model of Figure 5.2(b).*}

- set $t_a = t_s + t_r$. {*This assumes that* $P(t_a|t_s, t_r) = \delta(t_a - (t_s + t_r))$*, that is, the probability density function* $P(t_a|t_s, t_r)$ *is a Dirac function. Alternatively, we could have modeled* $P(t_a|t_s, t_r)$ *as a noise process specifying uncertainty on the time occurrence of the word.*}

The main assumption in the above model is that given the motifs, the words within the temporal document are independent of the motif start; that is, the occurrence of a word only depends on the motif, not on the time when a motif starts.

Furthermore, contrasting PLSM with PLSA and LDA models, we find that PLSM performs temporal modeling at two different levels which the latter misses out completely. The temporal modeling is done: a) within motifs to identify when words occur, *i.e.,* at which relative time with respect to the motif beginning, given by the distribution $P(w, t_r|z)$; b) within documents, to identify when a motif actually starts in the document given by the distribution $P(t_s|z, d)$. The term *motif* for $P(w, t_r|z)$ is therefore used to distinguish them from simple word distributions $P(w|z)$ of PLSA/LDA models.

Before going into more details of the model, let us establish the connections between the proposed model and its application to video activity analysis. Our input to the model is the preprocessed video represented by the word count matrix $n(w, t_a, d)$ in Figure 5.1(b) or 5.2(a). The words that appear in this temporal document are spatially localized activities like the blobs shown in Figure 5.1(a) or Figure 5.12. The method used to derive these local activities is detailed in section 5.4.1. The motifs $P(w, t_r|z)$ used in the generative process are the dominant activities that occur in the scene as shown in Figures 5.1(a) or in Figures 5.14, 5.15 or 5.16. The generative process uses the start times of activity motifs in the video represented by the distribution $P(t_s|z, d)$. This is indicated using the red and black arrows in Figure 5.1(b) and is used to detect events in section 5.5.3.

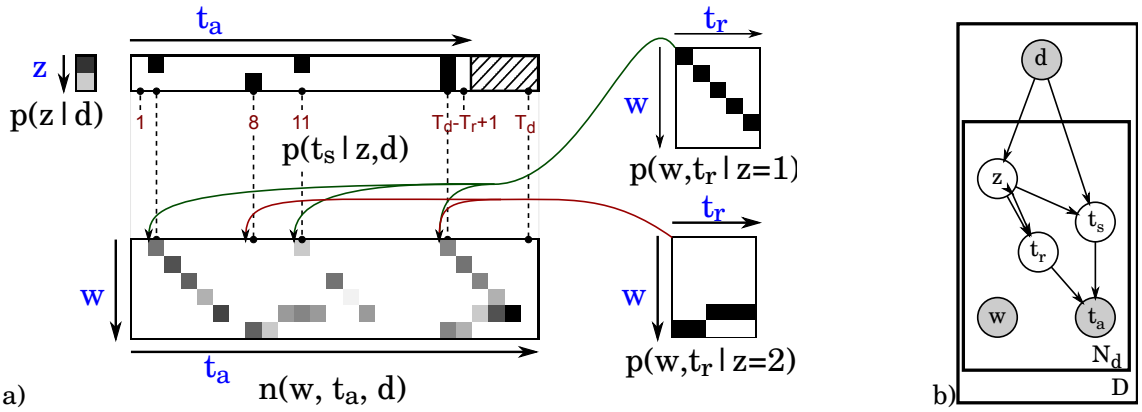


Figure 5.2. Generative process. a) Illustration of the temporal document $n(w, t_a, d)$ generation. Words $(w, t_a = t_s + t_r)$ are obtained by first sampling the motifs and their starting times from the $P(z|d)$ and $P(t_s|z, d)$ distributions, and then sampling the word and its temporal occurrence within the motif from $P(w, t_r|z)$. b) Graphical model (shaded circles represent observed variables and unshaded ones indicate latent variables).

The joint distribution of all variables can be derived from the graphical model. However, given the deterministic relation between the three time variables ($t_a = t_s + t_r$), only two of them are actually needed to specify this distribution. For instance, we have

$$\begin{aligned} P(w, t_a, d, z, t_s, t_r) &= P(t_r|w, t_a, d, z, t_s)P(w, t_a, d, z, t_s) \\ &= \begin{cases} P(w, t_a, d, z, t_s) & \text{if } t_r = t_a - t_s \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{5.1}$$

In the following, we will mainly use t_s and t_a. Accordingly, the joint distribution is given by:

$$P(w, t_a, d, z, t_s) = P(d)P(z|d)P(t_s|z, d)P(w, t_a - t_s|z). \tag{5.2}$$

5.2 Model inference

In this section we will discuss the model inference with details about the EM equations and the sparsity constraint.

5.2.1 Likelihood optimization with sparsity constraint

Our data $\mathcal{D}$ is the set of temporal documents $n(w, t_a, d)$. The likelihood of observing this data is given by the equation:

$$P(\mathcal{D}) = \prod_{d=1}^{D} \prod_{t_a=1}^{T_d} \prod_{w=1}^{N_w} P(w, t_a, d)^{n(w,t_a,d)} \tag{5.3}$$

From these observations, our goal is to discover the motifs and their starting times. This is a difficult task since the motif occurrences in the temporal documents overlap temporally, as illustrated in

Figure 5.2(a). The estimation of the model parameters Θ, *i.e.,* the probability distributions [2], $P(z|d)$, $P(t_s|z,d)$, and $P(w,t_r|z)$ can be done by maximizing the log-likelihood $\mathcal{L}(\mathcal{D}|\Theta)$ of the observed data $\mathcal{D}$. This is obtained by taking log on both sides of equation (5.3) and by marginalizing over the hidden variables $Y = \{t_s, z\}$ (since $t_r = t_a - t_s$, as discussed at the end of the previous subsection):

$$\mathcal{L}(\mathcal{D}|\Theta) = \sum_{d=1}^{D}\sum_{w=1}^{N_w}\sum_{t_a=1}^{T_d} n(w,t_a,d)\log\sum_{z=1}^{N_z}\sum_{t_s=1}^{T_{ds}} P(w,t_a,d,z,t_s) \tag{5.4}$$

Sparsity in topic models. One common issue in non-parametric topic models is that distributions are often loosely constrained, resulting in non-sparse process representations which are often not desirable in practice. Similar to the *sparse coding* representational scheme (Yi Zhang, 2010), what we seek are distributions where most of the elements in a vector are zero while few elements are significantly different from zero. For instance, in PLSA, one would like each document d to be represented by a few topics z with high weights $P(z|d)$, or each topic $P(w|z)$ to be represented by only a few words with high probability. But in practice, nothing guides the learning procedure towards such a goal. The same applies to LDA models despite the presence of priors on the multinomial $P(z|d)$ (Wang and Blei, 2009).

Approaches to this problem have been proposed in areas related to topic models. In Non-negative Matrix Factorization (NMF), a non-probabilistic model close to PLSA, Hoyer (2005) proposed to set and enforce through constrained optimization an a-priori sparsity level defined by a relationship between the L1 and L2 norm of the matrices to be learned. Very recently, Wang and Blei (2009) introduced a model that decouples the need for sparsity and the smoothing effect of the Dirichlet prior in HDP, by introducing explicit selector variables determining which terms appear in a topic. The even more complex focused topic model of Williamson *et al.* (2009) similarly addresses sparsity for hierarchical topic models but relies on an Indian Buffet Process to impose sparse yet flexible document topic distributions.

Introducing sparsity constraints. To address the sparsity issue, we propose an alternative approach. The main idea is to guide the learning process towards sparser (more peaky) distributions characterized by smaller entropy. We achieve this by adding a regularization constraint in the EM optimization procedure that favors lower entropy distributions by maximizing the Kullback-Leibler divergence between the distribution to be learned and the uniform distribution (maximum entropy). This results in a simple procedure that can be applied to most EM like inference schemes where a sparsity constraint on the distribution is desirable.

In our model this is the case of $P(t_s|z,d)$: one would expect this distribution to be peaky, exhibiting high values for only a limited number of time instants t_s. To encourage this, we propose to guide the learning process towards sparser distributions by using a penalized likelihood optimization criterion.

There are several candidates for this penalty term. For instance, we could use a direct approach by minimizing the L_0 or L_1 norm of $P(t_s|z,d)$ considered as a vector of parameters, or use

2. Note that $P(d) \propto N_d$ and is thus not unknown.

an entropy-based penalty term that favors low entropy and hence, sparse-distributions. However, such methods either do not suit well our probabilistic modeling approach (for instance, the L_1 norm of $P(t_s|z,d)$ can not be optimized as it is always equal to 1), or do not lead to simple optimization schemes (Besnerais *et al.*, 1999). Thus, we preferred to achieve this indirectly by adding a regularization constraint to maximize the Kullback-Leibler (KL) divergence $D_{KL}(U||P(t_s|z,d))$ between the uniform distribution U (maximum entropy) and the distribution of interest. Interestingly, in the past regularization approaches using KL-divergence to the uniform distribution have been used in physics (Besnerais *et al.*, 1999), and have been shown to have good properties for sparse approximation, such as differentiable and increased stability of the solution (Bradley and Bagnell, 2008). In our case, the formulation we exploit has also the advantage of leading to a simple modification of the EM inference scheme.

Though such an approach can be applied to any distribution of the model, we demonstrate this approach by applying it to $P(t_s|z,d)$. After development and removing the constant term, our constrained objective function is now given by:

$$\mathcal{L}_c(\mathcal{D}|\Theta) = \mathcal{L}(\mathcal{D}|\Theta) - \sum_{t_s,z,d} \frac{\lambda_{z,d}}{T_{ds}} \cdot \log(P(t_s|z,d)) \tag{5.5}$$

where $\lambda_{z,d}$ denotes a weighting coefficient balancing the contribution of the regularization compared to the data log-likelihood.

EM optimization. As is often the case with mixture models, equation (5.5) can not be solved directly due to the summation terms inside the logarithm. Thus, we employ an Expectation-Maximization (EM) approach and maximize the expectation of the (regularized) complete log-likelihood instead, defined as:

$$\begin{aligned} E[\mathcal{L}] = &\sum_{d=1}^{D}\sum_{w=1}^{N_w}\sum_{t_a=1}^{T_d}\sum_{z=1}^{N_z}\sum_{t_s=1}^{T_{ds}} n(w,t_a,d)P(z,t_s|w,t_a,d)\log P(w,t_a,d,z,t_s) \\ &- \sum_{t_s,z,d} \frac{\lambda_{z,d}}{T_{ds}} \cdot \log(P(t_s|z,d)) \end{aligned} \tag{5.6}$$

The solution is obtained by iterating the equation (5.7–5.10) (see Appendix A for more details on this derivation). In the Expectation step, the posterior distribution of the hidden variables is calculated as in equation (5.7) where the joint probability is given by equation (5.2). In the Maximization step equations (5.8 – 5.10), the model parameters are updated by maximizing equation (5.6) along with the constraint that each of the distributions sum up to one.

In practice, the EM algorithm is initialized using random values for the motif distributions (see also next subsection) and stopped when the data log-likelihood increase is too small. A closer look at the equations shows that in the E-step, the responsibilities of the motif occurrences (z, t_s) in explaining the word pairs (w, t_a) are computed (where high responsibilities will be obtained for informative words, *i.e.,* words appearing in only one motif and at a specific relative time), whereas the M-step aggregates these responsibilities to infer the motifs and their occurrences. In other words, the pos-

E-step:

$$P(z, t_s|w, t_a, d) = \frac{P(w, t_a, d, z, t_s)}{P(w, t_a, d)} \text{ where, } P(w, t_a, d) = \sum_{z=1}^{N_z} \sum_{t_s=1}^{T_{ds}} P(w, t_a, d, z, t_s) \tag{5.7}$$

M-step:

$$P(z|d) \propto \sum_{t_s=1}^{T_{ds}} \sum_{t_r=0}^{T_z-1} \sum_{w=1}^{N_w} n(w, t_s + t_r, d) P(z, t_s|w, t_s + t_r, d) \tag{5.8}$$

$$P(t_s|z, d) \propto \max\left(\varepsilon, \left(\sum_{w=1}^{N_w} \sum_{t_r=0}^{T_z-1} n(w, t_s + t_r, d) P(z, t_s|w, t_s + t_r, d)\right) - \frac{\lambda_{z,d}}{T_{ds}}\right) \tag{5.9}$$

$$P(w, t_r|z) \propto \sum_{d=1}^{D} \sum_{t_s=1}^{T_{ds}} n(w, t_s + t_r, d) P(z, t_s|w, t_s + t_r, d) \tag{5.10}$$

Figure 5.3. The EM algorithm steps.

terior terms $P(z, t_s|w, t_a, d)$ can be interpreted as weights or votes given to motif occurrences which are accumulated in equations (5.9–5.10) to identify the relevant motif occurrences. Furthermore, by associating occurring words (w, t_a) to motif occurrences (z, t_s), this posterior implicitly aligns all the words of a motif instance with its starting time, and as a consequence statistically achieves a soft alignment of multiple occurrences of the same motif, even in the presence of temporal overlap with the same or other motifs.

When looking at equation (5.9), we see that the effect of the additional sparsity constraint is to set to a very small constant ε the probability of terms which are lower than $\lambda_{z,d}/T_{ds}$ (before normalization), thus increasing the sparsity as desired. To set sensible values for $\lambda_{z,d}$ we used the rule of thumb $\lambda_{z,d} = \lambda \frac{n_d}{N_z}$, where n_d denotes the total number of words in the temporal document, and λ the sparsity level. Note that when $\lambda = 1$, the correction term $\lambda_{z,d}/T_{ds}$ in equation (5.9) is equal to the average (over t_s) of the term on the right hand side of equation (5.9) involving sums.

Inference on unseen temporal documents. Once the motifs are learned, their time occurrences in any new temporal document – represented by $P(z|d_{new})$ and $P(t_s|z, d_{new})$, can be inferred using the same EM algorithm, but keeping the motifs fixed and using only equation (5.8) and equation (5.9) in the M-step.

5.2.2 Maximum a-posterior Estimation (MAP)

In graphical models, Bayesian approaches are often preferred compared to maximum-likelihood (ML) ones, especially if there is knowledge about the model parameters. This is the case for methods

like LDA that can improve over PLSA by using Dirichlet priors on the multinomial distributions. However, as it was shown in Girolami and Kabán (2003) and Chien and Wu (2008), LDA is equivalent to PLSA when priors are uninformative or uniform, which is a common situation in practice.

The MAP estimation of parameters Θ can be formulated as follows:

$$\Theta_{\text{MAP}} = \arg\max_{\Theta}(\log P(\Theta|\mathcal{D})) = \arg\max_{\Theta}(\log P(\mathcal{D}|\Theta) + \log P(\Theta)) \tag{5.11}$$

where $P(\mathcal{D}|\Theta)$ is the likelihood term given by equation (5.4), and $P(\Theta)$ is the prior density over the parameter set. In practice, it is well known that using priors that are conjugate to the likelihood simplifies the inference problem. Since our data likelihood is defined as a product of multinomial distributions, we employ Dirichlet distributions as priors.

Application to the PLSM model. Our parameter set Θ comprises the multinomial parameters $P(w, t_r|z)$, $P(z|d)$, and $P(t_s|z, d)$. We don't have any a priori information about the motif occurrences $P(t_s|z, d)$ nor can we obtain an updated prior that is common to all the temporal documents in a general scenario. Moreover, for this term, we employ the sparsity constraint rather than a smoothing prior. Thus, we will use the MAP approach to set priors on the other multinomial parameters. Replacing in equation (5.5) the log-likelihood by the parameter log-posterior probability, the criterion to optimize simply becomes $\mathcal{L}_m(\mathcal{D}|\Theta) = \mathcal{L}_c(\mathcal{D}|\Theta) + \log P(\Theta)$, with the last term given by:

$$P(\Theta) \propto \prod_{d,z} P(z|d)^{\alpha_{z,d}-1} \prod_{z,w,t_r} P(w, t_r|z)^{\alpha_{w,t_r,z}-1}, \tag{5.12}$$

where $\alpha_{z,d}$ and $\alpha_{w,t_r,z}$ denote the Dirichlet parameters governing the prior distributions of $P(z|d)$ and $P(w, t_r|z)$ respectively. As before, $\mathcal{L}_m$ can be conveniently optimized using an EM algorithm, which leads to the same update expression as in Figure 5.3, except that equation (5.8) and equation (5.10) need to be modified to account for the prior.

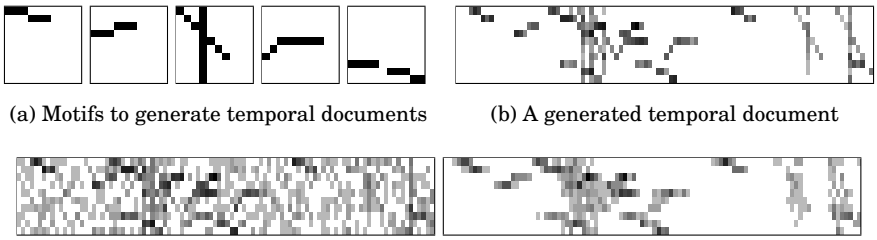
$$P_{\text{MAP}}(z|d) \propto (\alpha_{z,d} - 1) + \sum_{t_s=1}^{T_{ds}} \sum_{t_r=0}^{T_z-1} \sum_{w=1}^{N_w} n(w, t_s + t_r, d) P(z, t_s|w, t_s + t_r, d) \tag{5.13}$$

$$P_{\text{MAP}}(w, t_r|z) \propto (\alpha_{w,t_r,z} - 1) + \sum_{d=1}^{D} \sum_{t_s=1}^{T_{ds}} n(w, t_s + t_r, d) P(z, t_s|w, t_s + t_r, d) \tag{5.14}$$

5.2.3 Model Selection

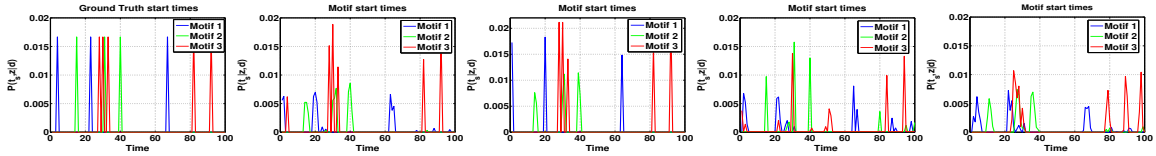
To select the right number of motifs, we use the BIC measure, which penalizes the training data likelihood based on the number of parameters and data points. A general version of the BIC measure of a model M is given by:

$$BIC(M) = -2\mathcal{L}(\mathcal{D}|\Theta) + \lambda_{bic} N_p^M \log(n) \tag{5.15}$$



(a) Motifs to generate temporal documents (b) A generated temporal document

(c) Temporal Document with uniform noise (d) Temporal Document with location noise



(e) Ground truth (f) Clean, $\lambda = 0.0$ (g) Clean, $\lambda = 0.5$ (h) Uni. noise, $\lambda = 0.5$ (i) Loc. noise, $\lambda = 0.5$

Figure 5.4. Synthetic experiments. (a) The five motifs, (b) A segment of a generated temporal document, (c,d) The same segment perturbed with: (c) Uniform noise ($\sigma_{\text{snr}} = 1$), (d) Location noise ($\sigma = 1$) added to each word time occurrence t_a. (e) the true motif occurrences (only 3 of them are shown for clarity) in the temporal document segment shown in (b). (f–i) the recovered motif occurrences $p(t_s|z, d)$ when using as input (f) the clean temporal document (cf b) and no sparsity constraint $\lambda = 0$ (g) or with sparsity constraint $\lambda = 0.5$; (h) the noisy temporal document (c) and $\lambda = 0.5$ (i) the noisy temporal document (d) and $\lambda = 0.5$.

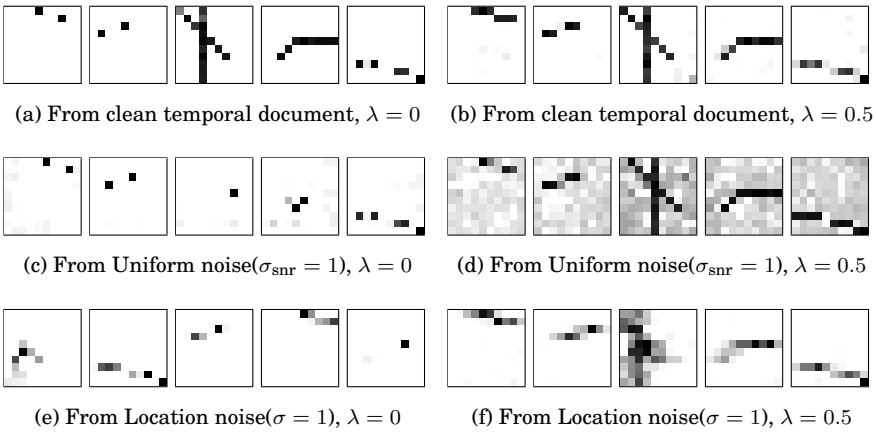
where, $\mathcal{L}$ is the likelihood of the model and is given by equation (5.4), N_p^M denotes the number of parameters of model M, n is the number of data points, and λbic is a coefficient that controls the influence of the penalty. Note that the above equation leads to the standard BIC criterion when λ_{bic}, the weight of the penalty term, is 1. In practice, we followed the approach of (Chen and Gopalakrishnan, 1998) or (Tritschler and Gopinath, 1999) and used a validation set to fix this parameter in the numerical experiments. In essence, this criterion seeks models that find a compromise between likelihood fitting and model complexity. In practice, we conduct optimization for models with different number of motifs according to previous subsections, and finally keep the model with the minimum BIC measure.

5.3 Experiments on synthetic data

In order to investigate and validate various aspects and strengths of the model we first conducted experiments using synthetic data.

5.3.1 Data and experimental protocol

Data synthesis. Using a vocabulary of 10 words, we created five motifs with duration ranging between 6 and 10 time steps (see Figure 5.4(a)). Then, for each experimental condition (*e.g.,* a noise type and noise level), we synthesized 10 temporal documents of 2000 time steps following the generative process described in section 5.1.2, assuming equi-probable motifs and 60 random occurrences per motif. One hundred time steps of one temporal document are shown in Figure 5.4(b), where



(a) From clean temporal document, $\lambda = 0$ (b) From clean temporal document, $\lambda = 0.5$

(c) From Uniform noise($\sigma_{\text{snr}} = 1$), $\lambda = 0$ (d) From Uniform noise($\sigma_{\text{snr}} = 1$), $\lambda = 0.5$

(e) From Location noise($\sigma = 1$), $\lambda = 0$ (f) From Location noise($\sigma = 1$), $\lambda = 0.5$

Figure 5.5. Synthetic experiments. Recovered motifs without (a,c,e) and with (b,d,f) sparsity constraints $\lambda = 0.5$ under different noise conditions. (a,b) from clean data; (c,d) from temporal documents perturbed by Uniform noise, $\sigma_{\text{snr}} = 1$, cf Figure 5.4(c); (e,f) from temporal documents perturbed with Location noise $\sigma = 1$, cf Figure 5.4(d).

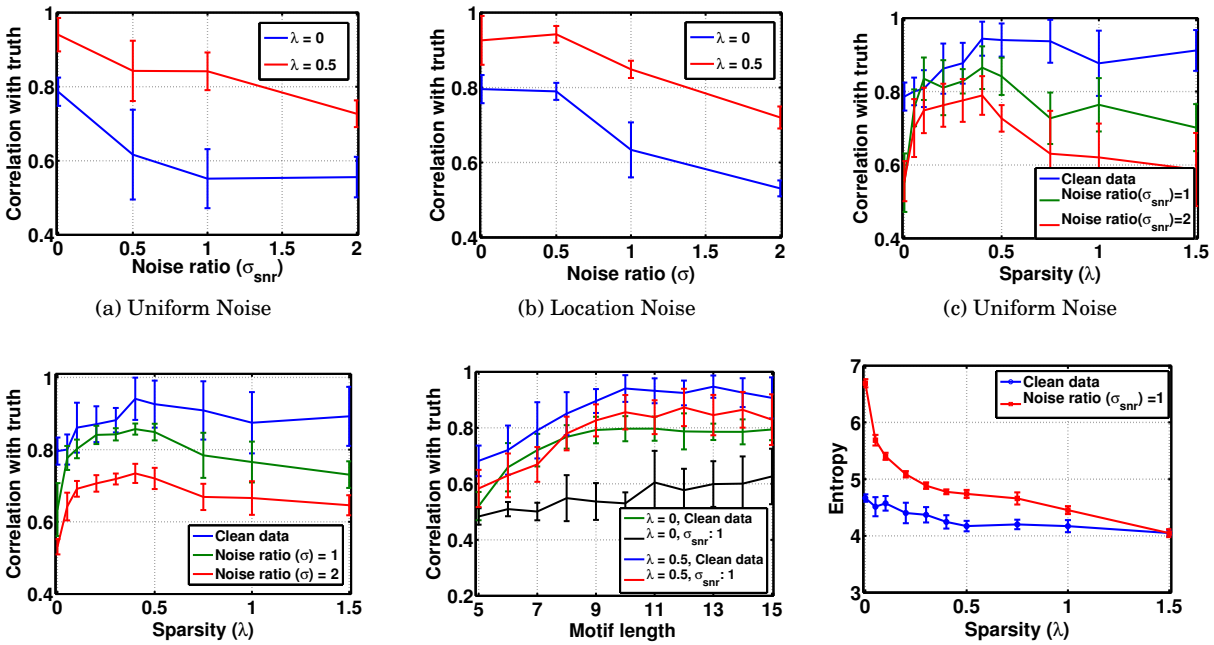
the intensities represent the word count (larger counts are darker). In Figure 5.4(e) corresponding starting times of the first three motifs out of the five motifs are shown for the sake of clarity. Note that there is a large amount of overlap between motifs.

Adding noise. Two types of noise were used to test the method's robustness. In the first case, words were added to the clean temporal documents by randomly sampling the time instant t_a and the word w from a uniform distribution, as illustrated in Figure 5.4(c). We call this *Uniform noise*. Here, the objective is to measure the algorithm's performance when the ideal co-occurrences are disturbed by random word counts. The amount of noise is quantified by the ratio $\sigma_{\text{snr}} = N_w^{noise}/N_w^{true}$ where, N_w^{noise} denotes the number of noise words added and N_w^{true} is the number of words in the clean temporal document. In practice, noise can also be due to variability in the temporal execution of the activity. Thus, in the second case, a *Location noise* was simulated by adding random shifts sampled from a Gaussian distribution with $\sigma \in [0, 2]$ to the time occurrence t_a of each word, resulting in blurry temporal documents, as shown in Figure 5.4(d).

Model parameterization. As we do not assume any prior on the parameter model, we did not use the MAP approach in these experiments, and optimized the penalized likelihood of equation (5.5). For each temporal document, 10 different random initializations were tried and the model maximizing the objective criterion was kept as the result.

Performance measure. The learning performance is evaluated by measuring the normalized cross correlation [3]. Averages and corresponding error-bars computed from the results obtained on the 10 generated temporal documents are reported.

3. The correspondence between the ground truth motifs and the estimated ones is made by optimizing the normalized cross-correlation measure between the learned motifs $\hat{p}(t_r, w|z)$ and the true motifs $p(t_r, w|z)$.



(a) Uniform Noise (b) Location Noise (c) Uniform Noise

(d) Location Noise (e) Changing motif length (f) Average entropy

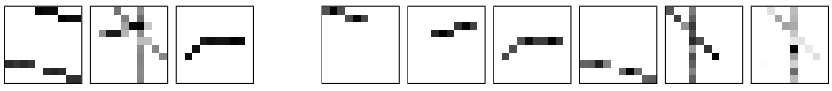
Figure 5.6. (a,b) Average motif correlation between the estimated and the ground truth motifs for different levels of (a) Uniform noise, (b) Location noise. (c,d) Average motif correlation between the estimated and the ground truth motifs for different sparsity weight λ and for different levels of (c) Uniform noise, (d) Location noise on a word time occurrence t_a. (e) Effect of varying motif length T_z from 5 to 15, for two levels of uniform noise. (f) Average entropy of $p(t_s|z,d)$ as a function of sparsity λ.

5.3.2 Results

Results on clean data. Figs 5.5(a) and 5.5(b) illustrate the recovered motifs with and without the sparsity constraint. As can be seen, without sparsity, two of the obtained motifs are not well recovered. This can be explained as follows. Consider the first of the five motifs. Samples of this motif starting at a given instant t_s in the temporal document can be equivalently obtained as a mixture of the first motif in Figure 5.5(a) occurring at three consecutive t_s values with probabilities less than one. This can be visualized in Figure 5.4(f), where the peaks in the blue curve $P(t_s|z=1,d)$ are three times wider and lower than in the ground truth. When using the sparsity constraint, the motifs are well recovered, and the starting time occurrences better estimated, as seen in Figure 5.5(b).

Robustness to noise. Figure 5.5(c) and 5.5(e) illustrate the recovered motifs under noise without a sparsity constraint. We can clearly observe that the motifs are not well recovered (*e.g.,* the third motif is completely missed). With sparsity, Figure 5.5(d) and 5.5(f), motifs are better recovered, but reflect the presence of the generated noise, *i.e.,* the addition of uniform noise in the motifs in the first case, and the temporal blurring of the motifs in the second case. The curves in Figure 5.6(a) and 5.6(b) show the degradation of the learning as a function of the noise level.

Effect of sparsity. We also analyzed the performance of the model by varying the weight of the sparsity constraint for different noise levels and noise types. Figure 5.6(a) and 5.6(b) show that the



(a) Three motifs (b) Six motifs

Figure 5.7. Estimated motifs sorted by their $P(z|d)$ values (given below each motif) when the number of motifs is (a) $N_z = 3$. True motifs are merged. (b) $N_z = 6$. A duplicate version of a motif with slight variation is estimated.

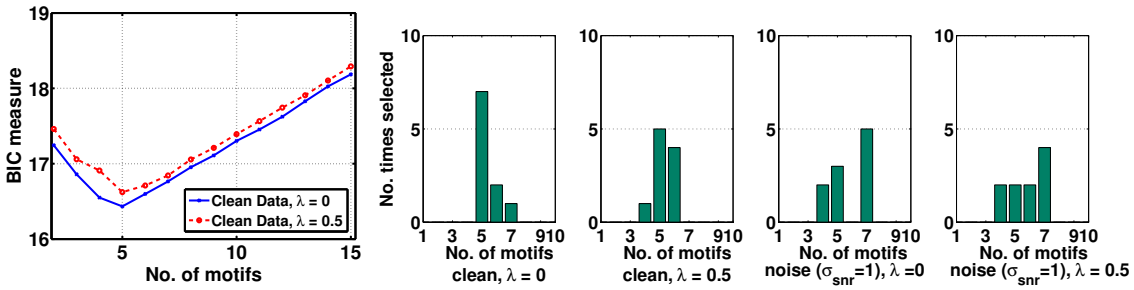
model is able to handle quite a large amount of noise in both cases, and that the sparsity approach always provides better results. While the best results without the constraint gives only a correlation of 0.8, we achieve a much better performance (approximately 0.95) with sparsity. In Figure 5.6(c) and 5.6(d), we see the performance of the method for various values of the sparsity weight λ and for varying noise levels. We notice that as the weight for sparsity increases, the performance shoots up. However, an increase in the sparsity weight beyond 0.5 often leads to degraded and sometimes unstable performance. Finally also note, as illustrated by Figure 5.6(f), that an increase in the sparsity weight λ leads to lowering the entropy of $p(t_s|z, d)$, as desired.

We conclude from these results that we obtain a marked improvement in recovering the motifs from both clean and noisy temporal documents when sparsity constraint is used.

Number of motifs and model selection. We first studied the qualitative effect of changing N_z, the number of motifs. As illustrated in Figure 5.7. When N_z is lower than the true number, we observe that each estimated motif consistently captures several true motifs. For instance, the first motif in Figure 5.7(a) merges the 1st and 5th motif of Figure 5.4(a). When the number of motifs is larger than the true value, like $N_z = 6$ in the example, we see that a variant of one motif is captured, but with lower probability. We observe the same phenomenon as we further increase the number of motifs.

We also tested our model selection approach based on the BIC criteria, as explained in section 5.2.3. To set λ_{bic}, we generated five extra clean temporal documents and used them to select an appropriate value of this parameter. Then, the same value was used to perform tests on other clean or noisy temporal documents. Figure 5.8(a) displays the BIC values obtained for a clean temporal document by varying the number of motifs from 2 to 15. As can be seen, the criteria reaches its minimum for 5 motifs. Histograms in Figure 5.8(b) show the number times a motif size is selected for a set of temporal documents. Although not perfect, the results show that the method is able to retrieve an appropriate number of motifs. In the presence of strong noise with as many noise words as true words, the number of found motifs is usually larger. This is expected as we need more motifs to explain the additional noise in the data.

Motif length. The effect of varying the maximum duration T_z of a motif and in the presence of noise is summarized in Figure 5.6(e). When T_z becomes lower than the actual motif duration, the recovered motifs are truncated versions of the original ones, and the "missing" parts are captured elsewhere, resulting in a decrease in correlation. On the other hand, longer temporal windows do



(a) BIC measure (b) Histogram of number of motifs selected

Figure 5.8. (a) Example of a BIC measure on a synthetic temporal document. (b) Selection histograms (number of times a motif is selected) using BIC on 10 temporal documents, with the following conditions: clean, $\lambda = 0$, clean, $\lambda = 0.5$, Noise ($\sigma_{\text{snr}} = 1$), $\lambda = 0$ and Uniform noise ($\sigma_{\text{snr}} = 1$), $\lambda = 0.5$.

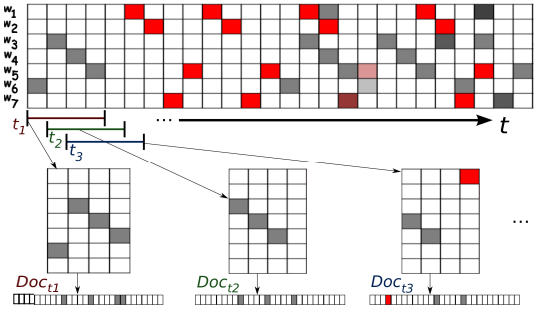


Figure 5.9. Illustration of the method Li *et al.* (2009). Individual TOS Bag-of-Words (BoW) documents $Doc_{t_1}, Doc_{t_2}, Doc_{t_3}, ...$ are created from the video count matrix by sliding a window over time, and considering each pair (word×relative-time-to-the-window-start) in these windows as a TOS word w_{ij}^{TOS}. In the example, the TOS vocabulary is thus of size $7 \times 4 = 28$ (ie documents are defined as counts of these w_{ij}^{TOS} words). Note how with TOS-LDA, the same motif occurrence in the video can result in different set of BoW representation depending on when the motif started w.r.t to the start of the document window, and that all documents are considered to be independent.

not really affect the learning, even under noisy conditions. However, the performance under clean and noisy conditions are significantly worse with no sparsity constraint.

Comparison with TOS-LDA (Li *et al.*, 2009). TOS-LDA works by collecting independent Bag-of-Words (BoW) documents from the video and then by applying an LDA topic discovery method to these documents. Figure 5.9 illustrates on the temporal documents shown in Figure 5.1(b) how the BoW documents of the TOS-LDA method are constructed. These documents are simply obtained from windows of a fixed temporal duration swept over the video, and by associating to each word w_i a time stamp t_j^r relative to the start of each fixed-size window. In other words, the TOS vocabulary is defined as the set of words $w_{ij}^{\text{TOS}} = (w_i \times t_j^r)$. In Figure 5.9, we show three sample documents created from the video by sliding a window of 4 time steps duration over three time steps. The first document is made of the TOS words $\{w_{61}^{\text{TOS}}, w_{32}^{\text{TOS}}, w_{43}^{\text{TOS}}, w_{54}^{\text{TOS}}\} = \{(w_6, t_1^r), (w_3, t_2^r), (w_4, t_3^r), (w_5, t_4^r)\}$, and similarly, the second document contains $\{w_{31}^{\text{TOS}}, w_{42}^{\text{TOS}}, w_{53}^{\text{TOS}}\}$, and so on for other documents. Thus, in this approach, one can clearly see that the same observed activity results in different sets of words for each document depending on its relative time occurrence within these sliding windows (in the example, documents 1 and 2 have orthogonal representations although they contains the same

Figure 5.10. Five temporal topics obtained from the TOS-LDA method Li *et al.* (2009) with clean data.

sub-activity from the video). Said differently, in the learning, *several* motifs (being time shifted versions of each other) will be needed to capture the *same* activity and account for the different times at which it can occur within the window. As pointed out in the literature review chapter 2, section 2.2.4, the method clearly lacks an alignment procedure that indicates when the motif starts, an information that is manually supplied in Faruquie *et al.* (2009) for activities based on traffic cycles.

We applied TOS-LDA to the same set of synthetic temporal documents created as previously described, and Figure 5.10 shows the obtained motifs when using clean data. Since the method does not align the activity motifs, none of the five extracted TOS-LDA topics truly represents one of the five patterns used to create the temporal documents. Rather, they contain parts, blurry and mixed versions of them, with some of the patterns (*e.g.,* the vertical bar) not appearing at all.

5.4 Application to video scene activity analysis

Our objective is to identify recurring activities in video scenes from long term data automatically. In this section, we explain how we can use the PLSM model for this purpose, and describe the video preprocessing used to define the words and temporal documents required by the PLSM model. We then present the datasets used for experiments and finally show three different ways of representing the learned motifs.

5.4.1 Activity word and temporal document construction

To apply the PLSM model to videos, we need to specify its inputs: the words w forming its vocabulary and that define the semantic space of the learned motifs, and the corresponding temporal documents. One possibility would be to define quantized low-level motion features and use these as our words. However, this would result in a redundant and unnecessarily large vocabulary. We thus propose to first perform a dimensionality reduction step by extracting spatially localized activity (SLA) patterns from the low-level features and use the occurrences of these as our words to discover sequential activity motifs using the PLSM model. To do so, we use the approach presented in chapter 4 and apply a standard PLSA procedure to discover N_A dominant SLA patterns through raw co-occurrence analysis of low-level visual words w^{ll}. The work flow of this process is shown in Figure 5.11, and explained below.

Low-level words w^{ll}. The visual words come from the location cue (quantized into 4×4 non-overlapping cells) and motion cue as explained in chapter 3. The foreground pixels are categorized into either static pixels (static label) or pixels moving into one of the eight cardinal directions by

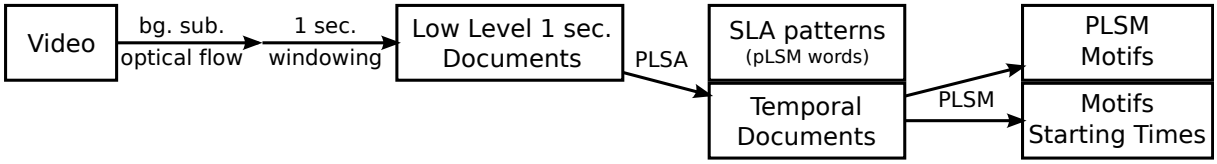


Figure 5.11. Flowchart for discovering sequential activity motifs in videos. Quantized low-level features are used to build 1 second bag-of-words documents, from which Spatially Localized Activity patterns (SLA) are learned and further used to build the temporal documents used as input to PLSM.

thresholding the flow vectors. Note that the static label will be extremely useful for capturing waiting activities, which contrasts with previous works (Wang *et al.*, 2009).

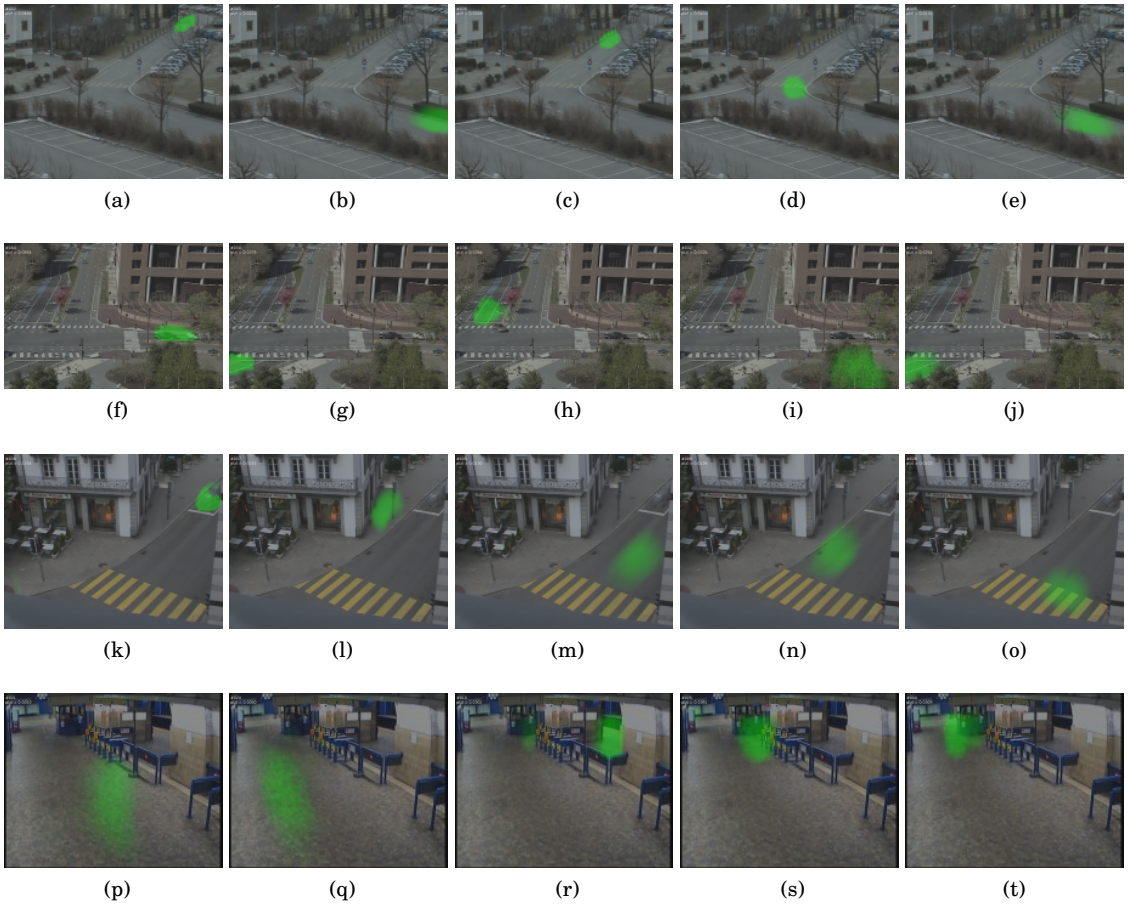


Figure 5.12. Representative SLA patterns obtained by applying PLSA on (a–e) far-field data, (f–j) MIT data and (k–o) Traffic junction data, (p–t) Metro station data.

Low-level SLA patterns z^{ll}. We apply the PLSA algorithm on a document-word frequency matrix $n(d_{t_a}, w^{ll})$ obtained by counting for the document d_{t_a} the low-level words appearing in N_f frames within a time interval of one second centered on time t_a. The result is a set of N_A SLA patterns characterized by their multinomial distributions $P(w^{ll}|z^{ll})$, and the probabilities $P(z^{ll}|d_{t_a})$ provid-

ing the topic distribution for each document. While PLSA captures dominant low-level word co-occurrences, it can also be viewed as a data reduction process since it provides a much more concise way of representing the video underlying activities at a given instant t_a, using only N_A topics, a number much smaller than the low-level vocabulary size. We observed that, with 50 to 100 SLA topics/patterns, we get an accurate description of the scene content. We also ran HDP (Teh *et al.*, 2006), which automatically finds the number of topics, and we obtained between 20 and 60 topics depending on the parameters and the dataset. Following these observations, we used $N_A = 75$ to get both a good representation of the scenes and a reasonable complexity for PLSM processing.

We can visualize the result of this step by superimposing the distributions $P(w^{ll}|z^{ll})$ over the image, indicating the locations where they have high probabilities. This is illustrated in Figure 5.12, which shows representative SLA patterns obtained from each of the four video scenes described below, with their locations highlighted in green [4]. Clearly, the SLA patterns represent spatially localized activities in the scene.

Building PLSM temporal documents. In our approach, we define the PLSM words as being the SLA patterns (*i.e.,* we have $w \leftrightarrow z^{ll}$ and $N_w = N_A$). Thus, to build the temporal documents d for PLSM, we need to define our word count matrix $n(w, t_a, d)$ characterizing the amount of presence of the SLA patterns z^{ll} in the associated low-level document at this instant t_a, *i.e.,* d_{t_a}. To do so, we exploit two types of information: the overall amount of activity in the scene at time t_a, and how this activity is distributed amongst the SLA patterns. The word counts were therefore simply defined as:

$$n(d, t_a, w) = n(d_{t_a})P(z^{ll}|d_{t_a}) \tag{5.16}$$

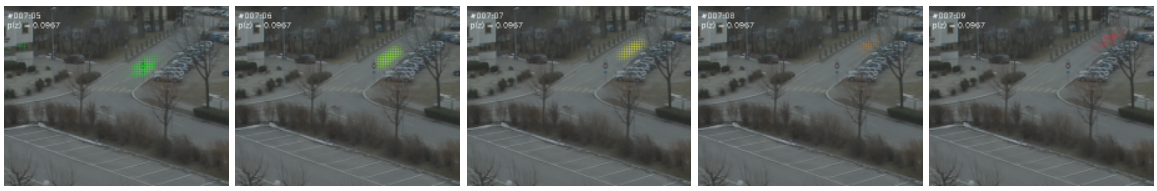
where $n(d_{t_a})$ denotes the number of low-level words observed at a given time instant (*i.e.,* within the 1 second interval used to build the d_{t_a} document). As for the number of temporal documents and their length T_d, this does not affect our modeling. Therefore, one may choose to use the entire video as a single document. However, in practice, we used documents of duration T_d equal to 120 (video clips of 2 minutes) for the urban datasets, or 600 (clips of 10 minutes) for the metro dataset.

5.4.2 Motif representation

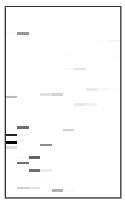
Before looking at the results obtained from the datasets, we explain how learned motifs are represented visually. In Figure 5.13, we provide three different ways of representing a recovered motif of $T_z = 10$ time steps (seconds) duration obtained from PLSM. By definition, a PLSM motif is a distribution $P(w, t_r|z)$ over $w \times t_r$ space. Thus the direct depiction of the motif is that of the $P(w, t_r|z)$ matrix as given in Figure 5.13(k). This shows that the distribution is relatively sparse, that words often occur at several consecutive time steps, and that several words co-occur at each time step. However, this does not provide much intuition about the activities captured by the

4. Note that the topic distributions contain more information than the location probability: for each location, we know what types of motion are present as well. This explains the location overlap between several topics, *e.g.,* between those of Figure 5.12(b) and Figure 5.12(e), which have different dominant motion directions.

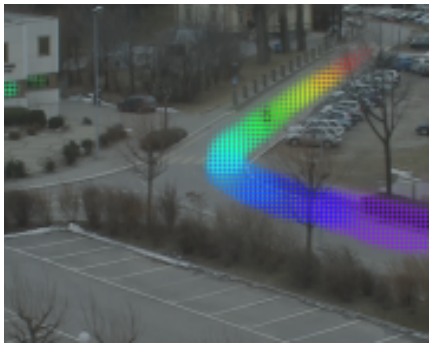
(a) t_r =1 (b) t_r =2 (c) t_r =3 (d) t_r =4 (e) t_r =5



(f) t_r =6 (g) t_r =7 (h) t_r =8 (i) t_r =9 (j) t_r =10



(k) $P(w, t_r|z)$



(l) collapsed t_r =1:10

Figure 5.13. Three different representations of a PLSM motif. (k) Motif probability matrix. The x axis denotes t_r, and the y axis the words. (a-j) For each time step t_r, weighted overlay on the scene image of the locations associated to each word (*i.e.,* the SLA patterns). (l) All time steps collapsed into one image color-coded according to the rainbow scheme, (Violet for $t_r = 1$ to Red for $t_r = T_z$).

motif. The second way of representing the motif is to back-project on the scene image and for each time step t_r, the locations associated with the words (the SLA patterns) probable at this time step, similar to the illustration of the SLA patterns in Figure 5.12. This is illustrated in Figure 5.13(a–j). This provides a good representation of the motif, but is space consuming. An even more realistic representation giving a true grasp of the motifs is provided by rendering them as animated gifs. Some results are hosted in the permanent institution web-page: `http://www.idiap.ch/paper/plsm/plsm.html`.

Due to media and space limitations, we use here an alternative version of these representations that collapses all time step images into a single image using a color-coded scheme, as shown in Figure 5.13(l). Note that the color at a given location is the one of the largest time step t_r for which the location probability is non zero. Hence, the representation may hide some local activities due to the collapsing effect. However, in the large majority of cases, the representation provides good intuition of the learned activities.

5.5 Video Scene Analysis Results

In this section, complementary details about the algorithm implementation are provided. Then, recovered motifs on the four datasets are shown and commented on. We then report the results of quantitative experiments on a counting task and on a prediction task to further validate our approach.

5.5.1 Experimental details

For the low-level processing, 1 second intervals were used to build the low-level documents and then the PLSM temporal document. To reduce the computational cost, optical flow features were estimated and collected in only $N_f = 5$ frames of these intervals. To favor the occurrence of the word probability mass at the start of the estimated motifs, we relied on the MAP framework and defined Dirichlet prior parameters for the motifs [5] as $\alpha_{w,t_r,z} = \tau \cdot \frac{1}{N_z} \cdot f(t_r)$, where f denotes a normalized (*i.e.,* the values of $f(t_r)$ sums to 1) decreasing ramp function as $f(t_r) \propto (T_z - t_r) + c$, T_z is the motif duration and c is a constant term. In other words, we did not impose any prior on the word occurrence probability, only on the time when they can occur. The strength of the prior is given by the term τ and is defined as a small fraction (we used 0.1) of the average number of observations in the training data for each of the $N_z \cdot N_w \cdot T_z$ motif bins. In practice, the prior plays a role when randomly drawing the motifs at initialization, where they are generated from the prior, and during the first few EM iterations. After, given the (low) level of the τ value and the concentration of the real observations on a few motif bins (see an estimated topic in Figure 5.13), its influence becomes negligible. With regards to the sparsity weight, since there is a large range of values from 0.15 to 0.5 that leads to good results, we set λ to 0.5 in all our real data experiments.

5.5.2 PLSM motifs and activities

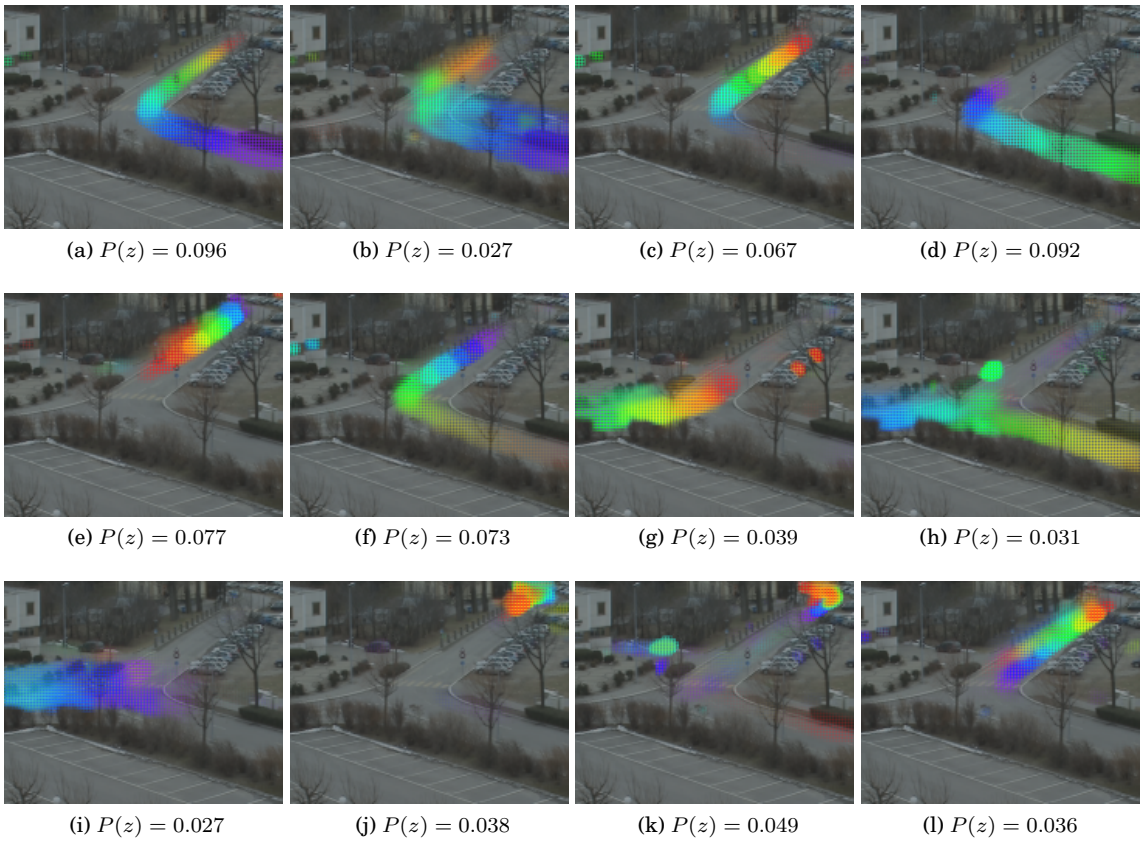
In this section, we show the motifs recovered from outdoor traffic scenes as well as from and single and multi-camera views from metro stations.

Outdoor scenes

We first sought for motifs of maximum 10 seconds duration, *i.e.,* $T_z = 10$. Note that 10 seconds already captures relatively long activities, especially when dealing with vehicles. At the end of this Section, we also show results when looking for 20 second motifs.

The number of 10s motifs selected automatically using the BIC criteria were 20, 26, 16 for the Far-field, MIT, Traffic junction datasets respectively. A selection of the top-ranking representative motifs are shown in Figure 5.14, Figure 5.15, Figure 5.16 and Figure 5.17 using the collapsed color representation (cf Figure 5.13), along with their probability $P(z)$ in explaining the training data. [6]

5. Note that we did not set any prior on the topic occurrences within the temporal document, *i.e.,* we set $\alpha_{z,d} = 0$.
6. In the website `http://www.idiap.ch/paper/plsm/plsm.html`, exhaustive set of results are provided with motifs rendered in animated-GIF.



(a) $P(z) = 0.096$ (b) $P(z) = 0.027$ (c) $P(z) = 0.067$ (d) $P(z) = 0.092$

(e) $P(z) = 0.077$ (f) $P(z) = 0.073$ (g) $P(z) = 0.039$ (h) $P(z) = 0.031$

(i) $P(z) = 0.027$ (j) $P(z) = 0.038$ (k) $P(z) = 0.049$ (l) $P(z) = 0.036$

Figure 5.14. Far-field data. Twelve representative motifs of 10s duration, out of 20. The method is able to capture the different vehicular trajectory segments.

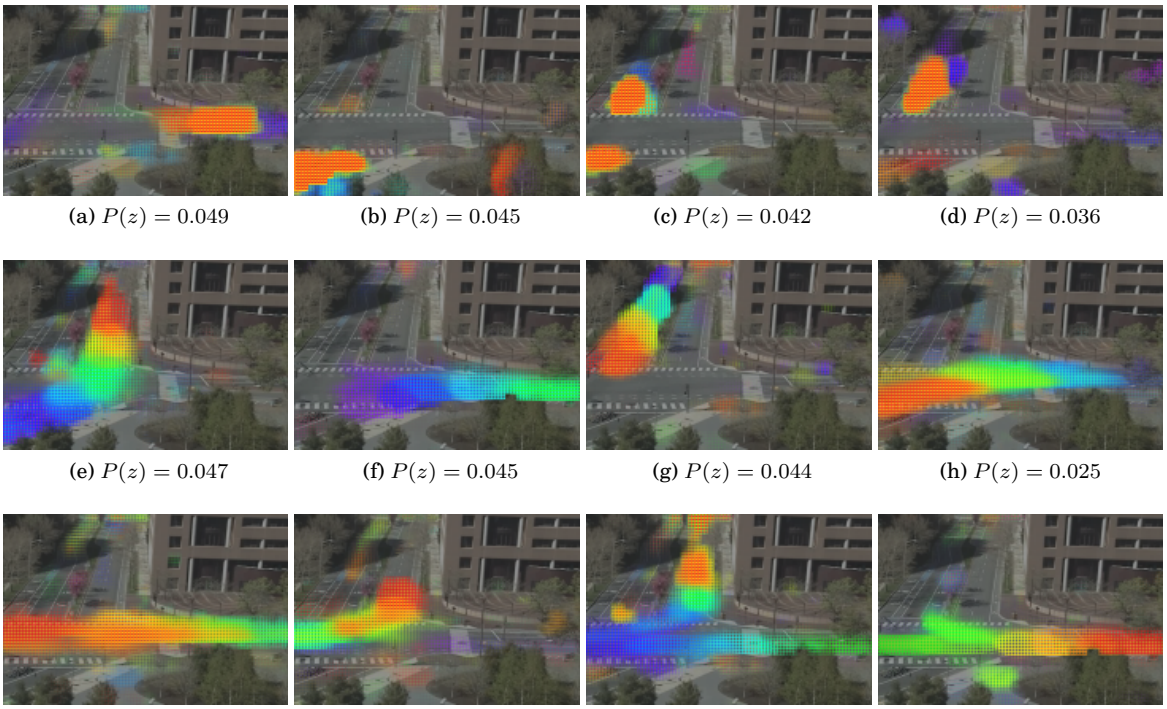
Below we comment on the results.

Far-field data. The analysis of the motifs show the ability of the method to capture the dominant vehicle activities and their variations due to differences of trajectory, duration, and vehicle type, despite the presence of trees at several places that perturb the estimation of the optical flow. For instance, Figure 5.14(a–c) correspond to vehicles moving towards the top right of the image, and Figure 5.14(d–f) to vehicles moving from the top right. Figure 5.14(g) corresponds to vehicles moving from left of the scene to the top right, Figure 5.14(h) to vehicles moving from left to bottom and Figure 5.14(i) to movement towards the left.

Some of the motifs capture the full presence of a vehicle in the scene (e.g, Figure 5.14(h)) but most of the activities are longer than the motif duration (10 seconds). The only solution for the algorithm is thus to split the activities in multiple motifs. For example motifs (g), (c) and (k) together cover the complete trajectory of a car going from the left to the top right of the scene. We observe that the algorithm tend to factor out the common subparts of the trajectories, for example motif (c) is also used for cars coming from the bottom right and going to the top right. The split of trajectories becomes unnecessary when we increase the motif duration, *e.g.,* to 20 seconds as in Figure 5.20.

We also see that vehicle speed has an impact on the recovered motifs. When the possible speed differences are important, multiple motifs are recovered for the different speeds. In Figure 5.14, this is the case for (a) and (b) which differ in the distance crossed by the motifs (and also in the size of the vehicle).

Motifs in Figure 5.14(j,k) represent the activities of vehicles moving in and out of the scene at the top of the scene. Since this location is far from the camera, and vehicles in both directions have to slow down due to a bump in the road, their apparent motion in the image is very slow and all the words are concentrated over a small region for the entire motif duration. Finally, the motif in Figure 5.14(l) represents the activity of two vehicles moving in opposite directions on the top part of the road.



(a) $P(z) = 0.049$ (b) $P(z) = 0.045$ (c) $P(z) = 0.042$ (d) $P(z) = 0.036$

(e) $P(z) = 0.047$ (f) $P(z) = 0.045$ (g) $P(z) = 0.044$ (h) $P(z) = 0.025$

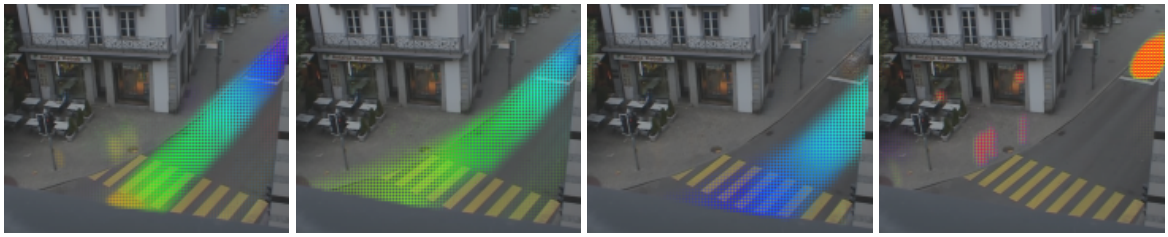
(i) $P(z) = 0.040$ (j) $P(z) = 0.039$ (k) $P(z) = 0.035$ (l) $P(z) = 0.040$

Figure 5.15. MIT data. Representative motifs of 10s duration out of 26. (a–d) Activities due to waiting objects. (e–l) Activities due to motion.

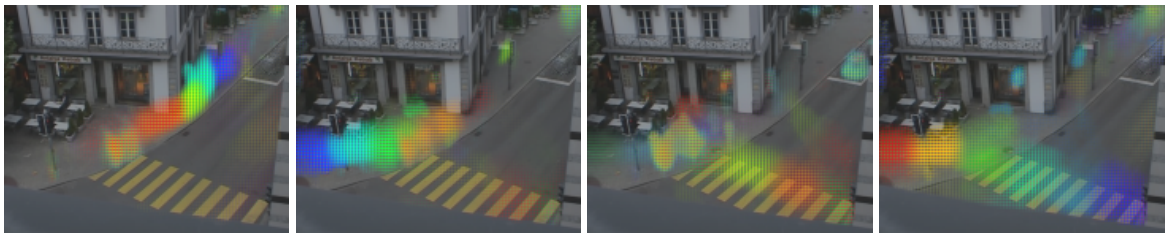
MIT data. This dataset is quite complex, with multifarious activities occurring concurrently and being only partially constrained by the traffic light. Even in this case, our method extracted meaningful activities corresponding to the different phases of the traffic signal cycle, as shown in Figure 5.15. Briefly speaking, one finds two main activity types: waiting activities, shown in Figure 5.15(a-d)[7], and dynamic activities as shown in Figure 5.15(e-l) of vehicles moving from one

7. Waiting activities are characterized by the same word(s) repeated over time in the motif. Thus the successive time color-coded images overwrite the previous ones in the collapsed representation as explained in Section. 5.4.2, leaving visible only the last (orange, red) time instant.

side of the junction to the other after the lights change to green. Note that waiting activities were not captured in previous works like Wang *et al.* (2009), are identified here thanks to the use of background subtraction and of static words.



(a) $P(z) = 0.090$ (b) $P(z) = 0.089$ (c) $P(z) = 0.085$ (d) $P(z) = 0.073$



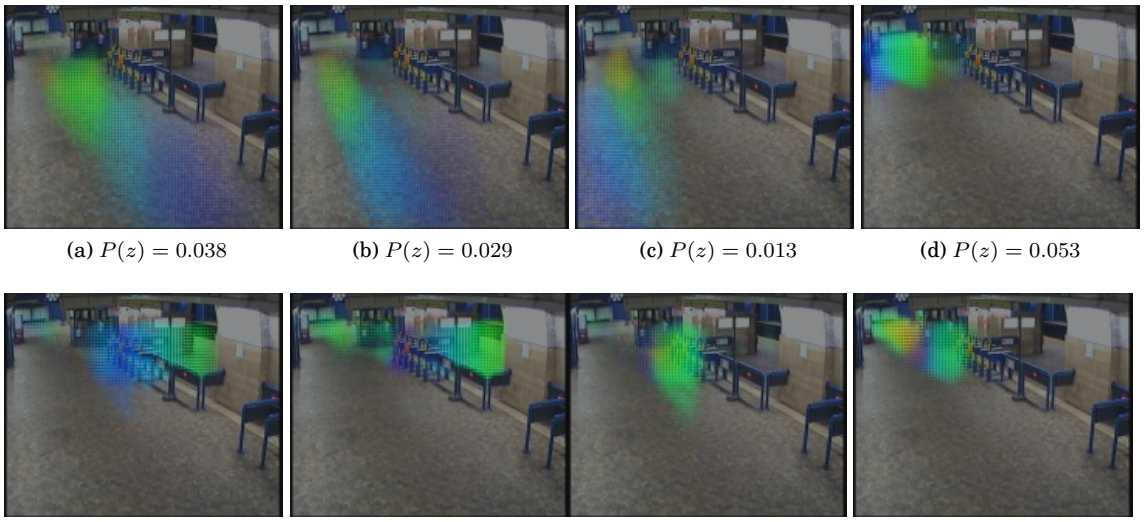
(e) $P(z) = 0.058$ (f) $P(z) = 0.052$ (g) $P(z) = 0.052$ (h) $P(z) = 0.048$

Figure 5.16. Traffic Junction data. Representative motifs of 10s duration. (a-d) vehicle activities. (e-h) pedestrian activities.

Traffic junction data. Despite the small amount of data (44min) and complex interactions between the objects of the scene, the method is able to discover the dominant activities as shown in Figure 5.16. For instance, Figures 5.16(a–c) show some dynamic activities due to vehicles in the scene. These activities usually last for just around 5 seconds, which explains the absence of the whole color range in the images. More specifically, Figure 5.16(a) corresponds to vehicles going straight; Figure 5.16(b) shows vehicles coming from the top right and turning to their right at the bottom and Figure 5.16(c) shows vehicles moving bottom middle to the top right. Waiting activities are also captured, as illustrated in the motif of Figure 5.16(d), which displays vehicles waiting for the signal. Interestingly, another set of motifs capture pedestrian activities, despite the fact that they are less constrained and have more variability in localization, size and shape, timing and dynamics. This comprises people moving on the sidewalk (Figure 5.16(e,f)), but also pedestrians crossing the road on the zebra crossing as in motifs from Figure 5.16(g,h).

Indoor scenes

For the Rome metro station, when the motif length was set to 10 seconds, we obtained 25 motifs using the BIC criteria. For the multicamera views from Torino, to allow motifs to capture any inter-camera transitions a maximum motif length of 15 seconds was used and we obtained around 20 motifs from each case. Note that for these challenging indoor scenes, only optical flow features



(a) $P(z) = 0.038$ (b) $P(z) = 0.029$ (c) $P(z) = 0.013$ (d) $P(z) = 0.053$

(e) $P(z) = 0.072$ (f) $P(z) = 0.070$ (g) $P(z) = 0.048$ (h) $P(z) = 0.091$

Figure 5.17. Metro Station data. Eight representative motifs of 10s duration, out of 25 depicting (a–c) people moving towards top of the scene from bottom right, bottom-middle and bottom left respectively; (d) people leaving the ticket machine towrds the turnstiles; (e–f) people around the turnstiles and crossing them at different places to reach the platform, and (g–h) mixed activities around the information booth and in the tunstile area.

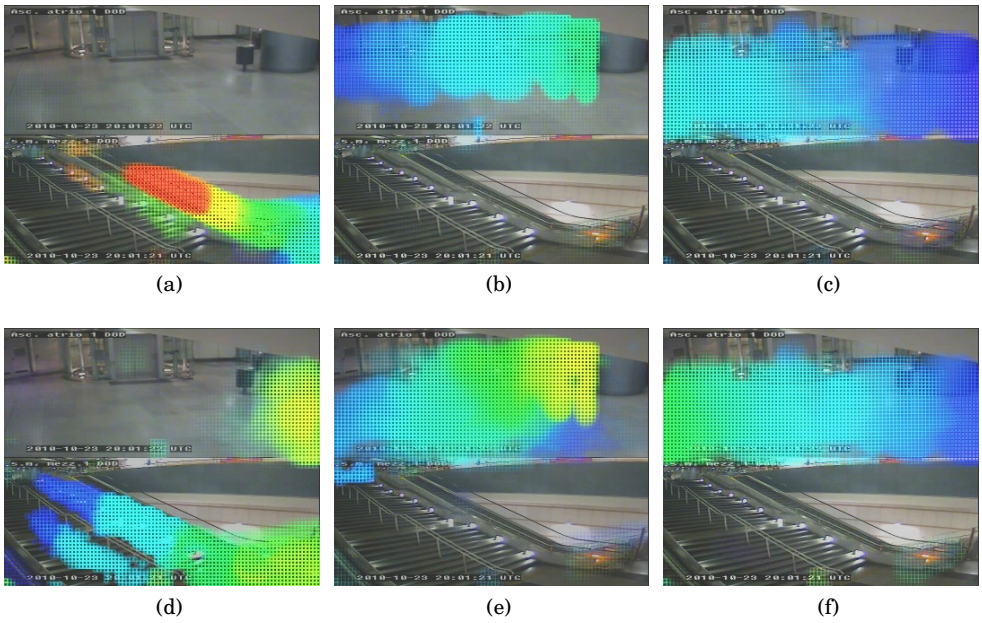
were used for the low-level visual representation.

Rome metro station. Despite the disorderly movements that continuously occur in the scene, the method is able to capture typical structured motion patterns that exist. Typical activities consists of people moving towards the top of the scene from different origins due to their arrival from different places in the metro station. Figure 5.17(a–c) show these movements from bottom right, bottom middle and bottom left respectively. Other typical activities correspond to people leaving the ticket machine towards the turnstiles Figure 5.17(d) and crossing the turnstiles at different places Figure 5.17(e,f), and mixed motion patterns, Figure 5.17(g,h). This clearly demonstrates that our model can successfully extract patterns even from crowded scenes containing unstructured movements.

Multi-camera activities from Torino metro station. Is it possible to obtain activities that span multiple cameras using PLSM and without explicit calibration methods? is an interesting question. To answer this question, we experimented with view pairs as illustrated in section 3.1.2; the first view-pair is from non-overlapping but neighboring metro regions and the second view-pair is from a ticketing hall viewed in orthogonal angles.

The views were combined at the feature level *i.e.,* first, optical flow features were extracted, quantized and words were created individually from the views. The individual words from the two cameras were then concatenated and treated as if from a single view. Then, creating SLA patterns and temporal documents for PLSM were done in the same way as explained in section 5.4.1.

Figures 5.18 and 5.19 show the motifs obtained from the view-pairs. In Figure 5.18, where the views are not overlapping, we find that the motifs are mostly independent. Figure 5.18(a)



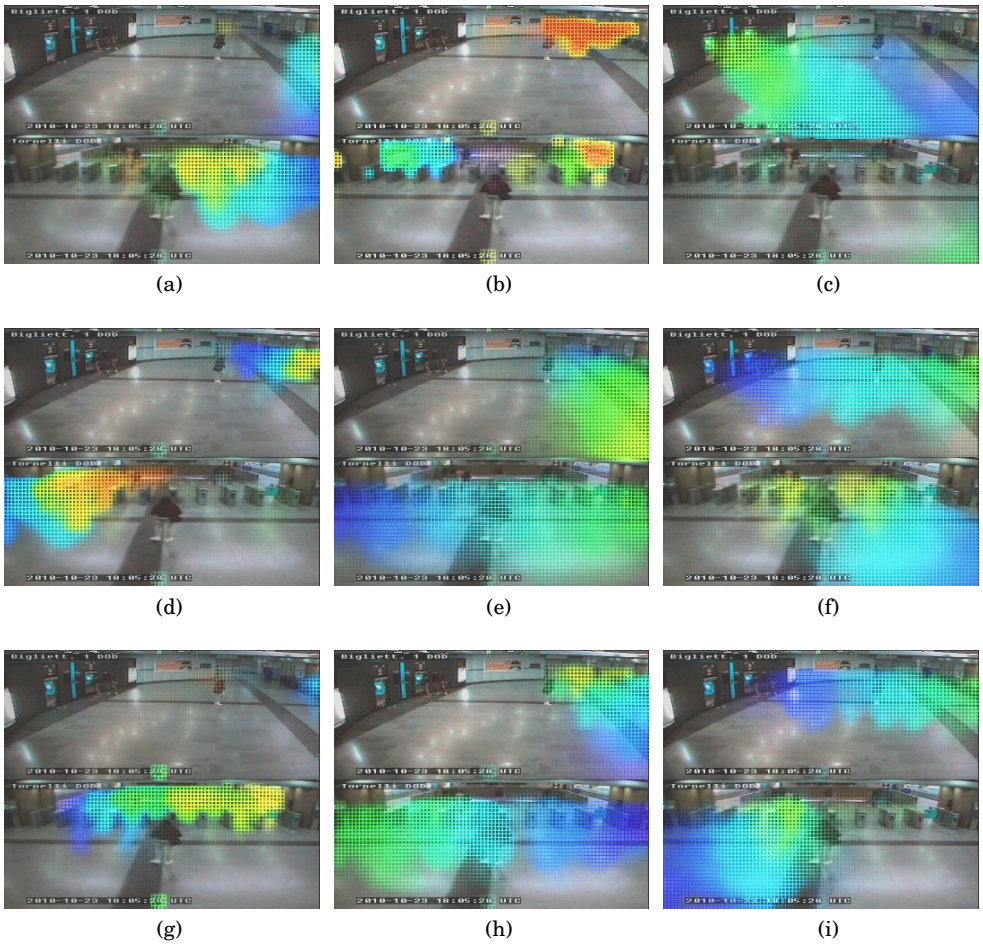
(a) (b) (c)

(d) (e) (f)

Figure 5.18. Representative motifs from non-overlapping view-pairs fused at the feature level. The motifs are mostly independent or restricted to single views, reflecting the non-overlapping nature of the views.

shows people going up the stairs mostly by the escalators. Figure 5.18(b,c) show some typical movements in the passage, *i.e.,* people moving from left to right and right to left respectively. The motifs in Figure 5.18(b,e) are also the exit paths coming either from the (invisible) mezzanine in Figure 5.18(b) or from the stairway in Figure 5.18(e) (with the motif correctly spanning over the two cameras). Motifs also capture people entering the passage (from the turnstiles). There are multiple variations (6 of them) of this activity due to distance from the camera and speeds. One speed variation is shown in Figure 5.18(c,f) which differ only by the color palette: Figure 5.18(f) ends in green meaning it is slower and longer than Figure 5.18(c).

When the cameras are overlapping as shown in Figure 5.19, PLSM motifs help us infer the relationships exisitng across multiple views. Figure 5.19(a,d) show people entering the station and going to the nearest turnstiles. Figure 5.19(g) shows people entering from the left side often go to the right side. This behavior is explained by the presence, on the right, of an elevator and the escalator for going down to the platform. Figure 5.19(b) captures people coming from the escalator (blue in the middle left part of the lower image), and then exiting the station (red in the upper image) after passing behind the pillar. We also see that this behavior is actually correlated with some activity on the right: we see here the two ways of reaching the station exits from the metro platform. This is an example where the motifs explain the entry and exit points in the views. Figure 5.19(e,h) show some typical trajectory of people who use the station as an underground passage without actually going beyond the turnstiles. Figure 5.19(c) shows one of the different ways of going to the vending machine. In Figure 5.19(f,i), the two ways of leaving the vending



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 5.19. Representative motifs from overlapping view-pairs fused at the feature level. Motifs here reflect the relationship between the two views very well. The

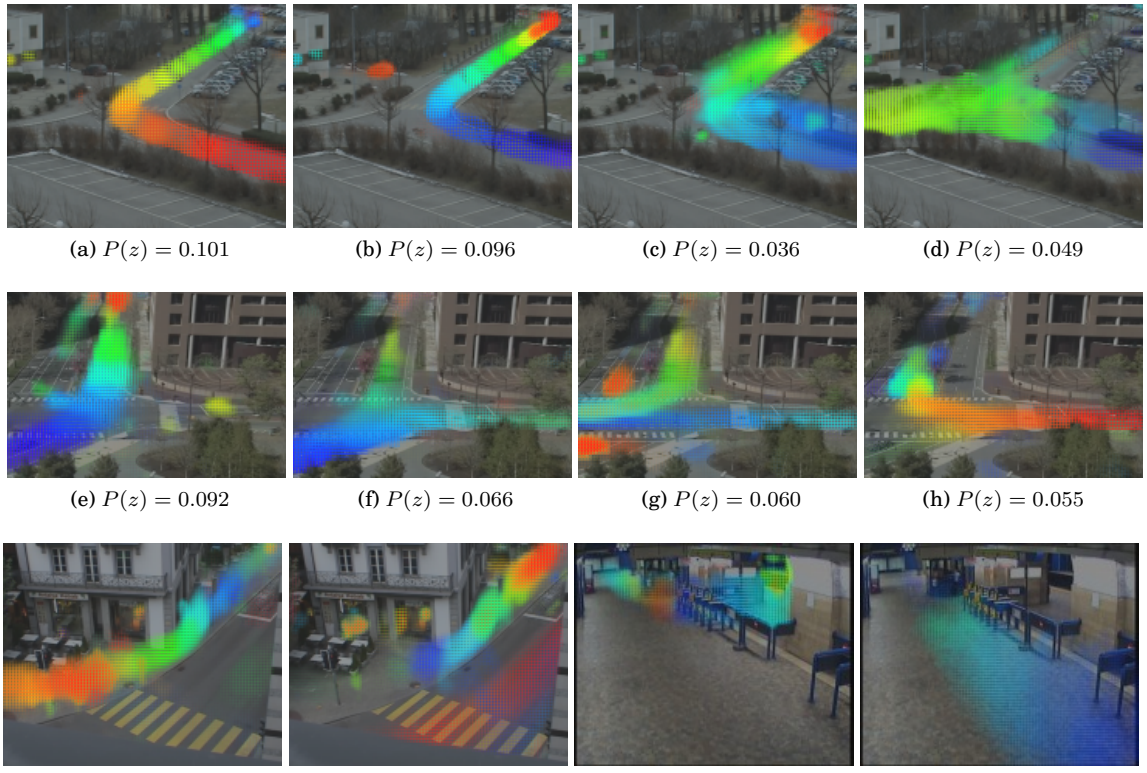
machines are seen: Figure 5.19(f) for the one on the right and Figure 5.19(i) for the one on the left.

From these motifs, we observe that we can extract activities that span multiple cameras by simply fusing the features together. While the cameras are overlapping, most motifs span both the cameras explaining their relationships and providing a *soft camera calibration*. When the views are non-overlapping, motifs are independent and mostly restricted to single views. These examples also highlight the advantage of the temporal aspect of the PLSM motif. That is, the temporal order within motifs has enabled us to explain where or in which camera view an activity starts and where it ends, giving clues about entry and exit points in the scene. Additionally, the temporal order can be used to infer the transit times between the camera views.

Many existing methods for multi-camera activity analysis rely on either a) inter-camera calibration, b) inter-camera object correspondences or c) modeling distribution of transition times between cameras. The work closest to what is presented in this section is by Loy *et al.* (Loy *et al.*, 2009),

where they rely on more robust lower level features (static and moving activities in the foreground) computed over semantically derived region segments. Cross Canonical Correlation Analysis (xCCA) is then applied to extract relations among the regions from multiple views and derive the topology of the camera network. Although the obtained model is used to improve person reidentification across views, the approach does not result in a detailed temporal activity model with automatic soft calibration as we propose here.

Topic Length.



(a) $P(z) = 0.101$ (b) $P(z) = 0.096$ (c) $P(z) = 0.036$ (d) $P(z) = 0.049$

(e) $P(z) = 0.092$ (f) $P(z) = 0.066$ (g) $P(z) = 0.060$ (h) $P(z) = 0.055$

(i) $P(z) = 0.102$ (j) $P(z) = 0.088$ (k) $P(z) = 0.161$ (l) $P(z) = 0.082$

Figure 5.20. Motifs of 20s duration that mainly differ from their 10s shorter counterparts. (a–d) Far-field, (e–h) MIT, and (i–j) Traffic junction data (k–l) Metro station data. All the above motifs capture the full extent of the activities within the scene.

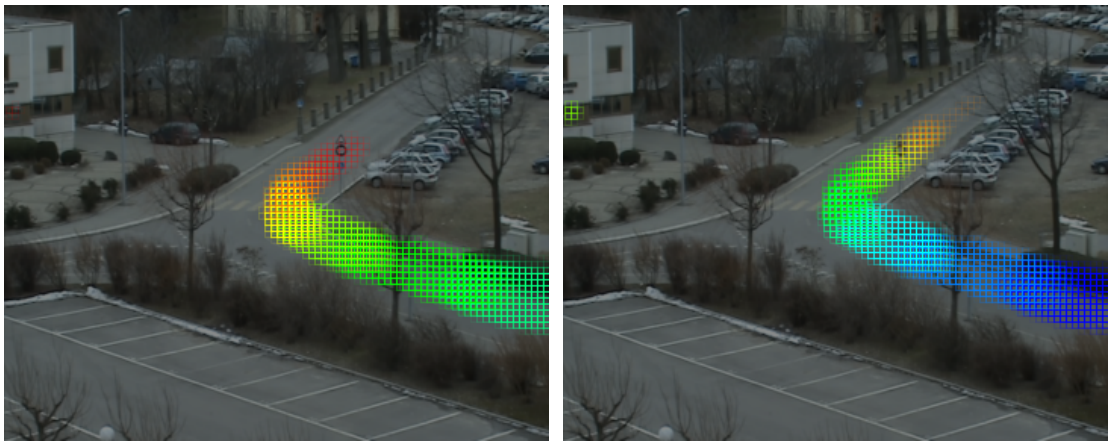
We also experimented with longer motif duration T_z. For instance Figure 5.20 shows motifs of 20 second duration from all the three datasets. Since longer motifs capture more activities, the BIC measure selected only 16, 16, 14 and 12 motifs for the Far-field, MIT, Traffic junction and Metro station data respectively. Broadly speaking, when one extends the motif maximal length beyond the actual duration of a scene activity, the same motif is estimated, as already observed with synthetic data [8]. This is typically the case with the short vehicle motifs in the MIT (Figure 5.15(f,l))

8. Note however that longer motifs increase the chance of observing some random co-occurrences, as the amount of overlap with other activities, potentially unexplained by current motifs, increases as well. This is particularly true when the amount of data is not very large like in the Traffic junction case.

or Traffic junction (Figure 5.16(a-c)) datasets. Still, as activities can often be described with different time granularities, variations or other motifs may appear. For instance, as the travel time of vehicles in the Far-field or MIT scenes usually lasts longer than 10 seconds, vehicle activities are now captured as a single motif as shown in Figure 5.20 rather than as a sequence of shorter motifs of 5 to 10 seconds in length. As an example, the motif in Figure 5.20(a) combines the activities of Figure 5.14(e,d). The same applies with the pedestrian activities in the Traffic Junction case (cf Figure 5.20(i,j)). In case of metro station data, the motif in Figure 5.20(k) captures people moving towards the turnstile and crossing it, one after another. Figure 5.20(l) shows movement from bottom of the scene.

Effect of MAP Prior on time

In this section, we provide an illustration of how the MAP framework is used to improve PLSM results. In section 5.5.1, we proposed to use MAP to favor the occurrences of the motif words at the start of the motif. Since our motifs aim at capturing real world activities, we prefer that the SLA activity starts exactly at the first time instant of the motif rather than after a few time steps. This is explained in Figure 5.21.

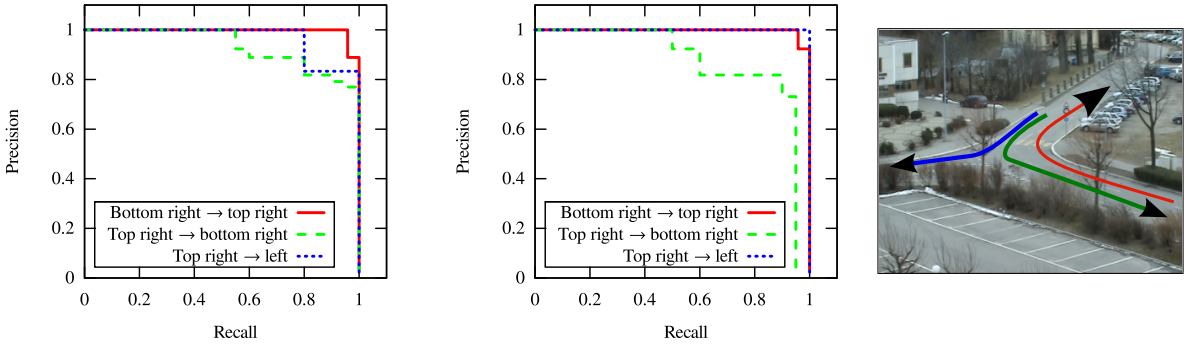


(a) without MAP prior (b) with MAP prior

(c) $P(w, t_r|z)$ of (a) (d) $P(w, t_r|z)$ of (b)

Figure 5.21. Illustrating effect of MAP prior on t_r.

In Figure 5.21, the same activity of vehicle moving from bottom right and taking a right turn



(a) without sparsity (b) **with sparsity** (c) events

Figure 5.22. Precision/recall curves for the detection of 3 types of events mapped onto 3 topics, evaluated on a 12 minute test video. Results are provided both without and with sparsity. F-score at equal precision and recall (three curves): without sparsity (0.958, 0.818, 0.833), with sparsity (0.958, 0.818, 1). Area under the curves: without sparsity (0.995, 0.931, 0.967), with sparsity (0.997, 0.867, 1).

extracted from two runs of PLSM (a) without a prior, (b) with a MAP prior on t_r is shown. The prior used is a decreasing ramp as a function of the time-step and the length of the motif, given by $f(t_r) \propto (T_z - t_r) + c$. Figure 5.21(c and d) show the $P(w, t_r|z)$ matrix of Figure 5.21(a and b) respectively. Darker color indicates a higher probability. The motifs are 10 time steps (10 seconds) long. Without prior (cf. Figure 5.21(a and c)), we can observe that no words occur in the first 3 time steps, which means that the activity captured by PLSM effectively starts after a relative time of 4 seconds from the beginning of the motif. This is also indicated by the rendered motif in Figure 5.21(a) that starts with green color instead of blue. When using the ramp MAP prior on the temporal axis, the PLSM algorithm actually recovers motifs with words occurring in the first time steps of the motifs, as seen in the matrix Figure 5.21(d) and in the image Figure 5.21(b) which starts with the Violet-Blue color. Notice that as a consequence of having words at the motif beginning, PLSM will better capture longer activities, as shown by the longer trailing parts in the matrix (d) which is reflected by a longer extension of the activity towards the top right of the scene in the (b) image.

5.5.3 Event detection

To evaluate how well the recovered motifs match the real activities observed in the data, we performed a quantitative analysis by using the PLSM model to detect particular events. Indeed, as the model can estimate the most probable occurrences $P(t_s, z|d)$ of a topic z for a test document d, it is possible to create an event detector by considering all t_s for which $P(t_s, z|d)$ is above a threshold. By varying this threshold, we can control the trade-off between precision and completeness (*i.e.,* recall).

For this event detection task, we labeled a 12 minute video clip from the Far-field scene, distinct from the training set, and considered all the different car activities that pass through the three

road junction. Activity categories that occurred fewer than 5 times in this test data were discarded, which left us with the 3 activity categories depicted in Figure 5.22 with a total of 51 occurrences.

Given the fully unsupervised nature of our method, we manually associated each ground truth category to one of the discovered motifs (of maximum 10 second duration). This one to one manual association is a very minimal form of supervision and is somewhat suboptimal: the recovered PLSM motifs might not be matching the labelled events, and the one to one matching is somewhat limiting (see results below). The motifs considered for event detection are shown in 5.14(a,d,i). Using the occurrences $P(t_s, z|d)$ of these motifs [9], precision/recall curves were computed. They are shown in Figure 5.22 both without and with sparsity.

From the curves, it is evident that for two out of the three events, we obtain a close to 100% result, especially with sparsity. The worst performance is for the activity "top right to bottom right" which gives a F-score of around 80%. This lower performance is due to the fact that there are two motifs recovered that can explain the same ground truth activity (they may represent minor variations in speed but could be merged to improve results). Overall, the results prove that the discovered motifs match the real activities well, and that motif starting times could be exploited for real event detection.

5.5.4 Activity prediction

The predictive model. Given the set of scenes used here, it is possible to predict the most probable words that can occur in the near future. This is used mainly as a method to evaluated different other models with the proposed PLSM method. We have thus defined our task as estimating the probability $P_t^{pred}(w)$ that a word w appears at time t given all past information, that is, given the temporal document $n(w, t_a, d)$ up to time $t_a = t - 1$.

In our generative modeling approach, a word at time t can occur due to either a motif that has already started at a past time $t_s \in [t - T_z + 1, t - 1]$, or due to a motif that starts at the same time t. Hence, we define the prediction model as:

$$P_t^{pred}(w) \propto (1-\gamma) \sum_{t_s=t-T_z+1}^{t-1} \sum_z \hat{P}(t_s, z|d) P(w, t-t_s|z) + \gamma \sum_z P(z) P(w, 0|z), \tag{5.17}$$

where $\hat{P}(t_s, z|d)$ denotes our estimation that the motif z starts at time t_s given the observed data, γ represents the probability that a topic starts at the current instant, and $P(z)$ represents the motif prior probability estimated (along with the motifs) on training data [10]. To set γ, we have given equal priority to the starting time instants, and set $\gamma = \frac{1}{T_z}$, *i.e.,* a value of 0.1 in the current experiments. To obtain $\hat{P}(t_s, z|d)$ we simply apply our inference procedure to the temporal document $n(w, t_a, d)$

9. To perform the temporal association we allowed a constant offset between the event in the ground truth, and the starting time of a motif learned from PLSM.

10. Note that rather than simply using $P(z)$ as the prior for a motif to start at time t, we could have further exploited the past informations available in the past motif occurrences $\hat{P}(t_s, z|d)$ (*e.g.,* the motif of Figure 5.14(e) is often followed by that of Figure 5.14(d) several seconds later). However, as this is not part of our model, we preferred to go for the simpler case.

using only observations up to time $t-1$.

Evaluation protocol. The prediction performance was evaluated using a standard 10 folds cross-validation approach. That is, each dataset (5500 and 6500 time steps in the MIT and Far-field cases, respectively) was split in 10 folds. Then, for each fold, the complementary 90% of the data was used to train a model that was tested and evaluated on this fold. The reported results are the average over the 10 folds. As performance measure, we used the average normalized prediction log-likelihood (ANL) defined as:

$$\text{ANL} = \frac{1}{N_{test}} \sum_t \frac{\sum_w n(w,t,d) \log(P_t^{pred}(w))}{\sum_w n(w,t,d)} \tag{5.18}$$

It is a standard measure for evaluating the modeling performance of topic models (Wang *et al.*, 2009), and is directly (inversely) related to the perplexity measure that is also commonly used to evaluate the generalization power of topic models (Blei *et al.*, 2003a; Hofmann, 2001). A higher ANL value indicates a better predictive capacity and vice versa. In order to compare the prediction accuracy of our model, we implemented three other methods for prediction.

Temporal Constancy. In this method, we assume that observations do not change over time and hence observations at any time can be approximated by the observations from its immediate past. More specifically, the probability $P_t^{pred}(w)$ of a word appearing at time t is given by:

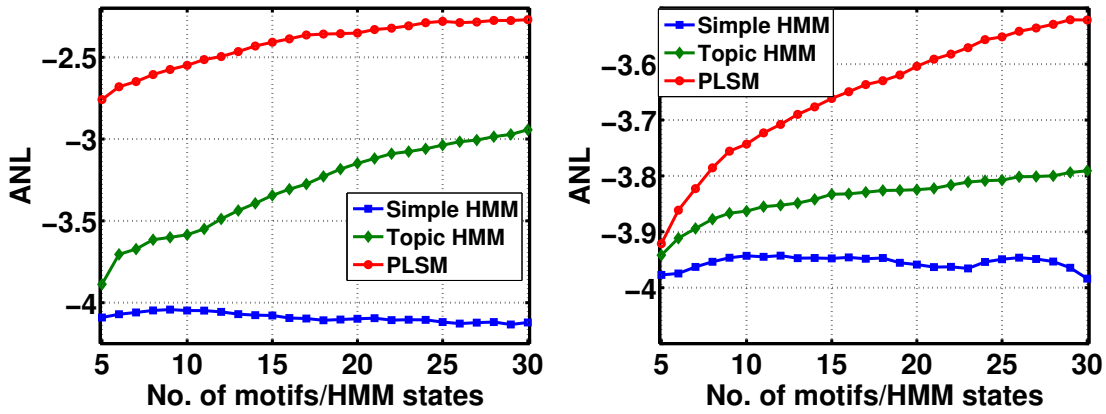
$$P_t^{pred}(w) = \frac{n(w,t-1)+c}{cN_w + \sum_w n(w,t-1)}, \tag{5.19}$$

where the constant c is a smoothing term added to predict a possible future word that does not occur at time $t-1$. A popular terminology for this is Laplace smoothing. We set $c=1$ in our experiments.

Simple HMM. Here, the sequences of observation vectors $o_t(w) = n(w,t,d)$ from the training temporal documents were used to learn in an unsupervised fashion (*i.e.,* by maximizing the data-likelihood) a fully-connected HMM with n states. The emission probabilities were defined as Gaussians with a diagonal covariance matrix. At test time, the trained HMM was used to compute the expected state probability at time t given all observations up to time $t-1$, from which the expected observation vector (and hence a predicted word probability $P_t^{pred}(w)$) was inferred.

Topic HMM. The second model is a more sophisticated approach in line with (Hospedales *et al.*, 2009), wherein the Markov chain models the dynamics of a global behavior state. More precisely, we first apply PLSA (with n topics) to the set of training documents $\{o_t, t \in training\}$. This results in a set of topics $P(w|z)$ and topic distributions $o'_t(z) = P(z|o_t)$. We then learn a HMM with n states using the topic observation sequence o'_t. The HMM states learned with this method capture distinct scene level behaviors characterized by interacting topics and the Markov chain models the temporal dependencies among them. We thus refer to this method as *Topic HMM*. At test time, the expected state, topic and word probability distributions can be successively computed using the learned model.

Results. Firstly, the results from the temporal constancy method (equation (5.19)) were interest-



(a) Far-field data (b) MIT data

Figure 5.23. Average Normalized Prediction log-likelihoods for (a) Far-field data, (b) MIT data. In both plots, the x-axis represents either the number of motifs (PLSM model), or the number HMM states in the two other cases.

ing and helped us in understanding the nature of the two datasets better. The AVL values obtained from this method for Far-field and MIT datasets were -3.5 and -3.3 respectively. In case of Far-field data, this method outperforms the HMM method for all states. It also performs better than Topic-HMM until 10 states. However for the MIT dataset, it also outperforms the PLSM approach [11]. Investigating these results revealed that MIT data is dominated by a huge number of waiting activities, a phenomenon that represents 45% to 55% of the total observations. Since waiting activities produce repeating words, a smoothed replication of the close past makes accurate predictions. This accuracy (for the waiting activities) is difficult to surpass. For instance, PLSM learns motifs that capture co-occurring activities. When only some activities associated to a given motif are appearing in the past, this motif is nevertheless triggered and used to predict the potential occurrence of all the activities in the future, hence loosing some prediction accuracy. In other words, PLSM learns co-occuring activities, such as cars waiting in opposite directions of the junction, but not necessarily isolated events such as a car waiting only in one direction. Hence, when an isolated 'waiting' event happens, the simple replication of the past will be much more accurate. Finally, note that the constancy assumption does not generalize well besides these waiting activities: it performs worse in the Far-Field case, and would perform poorly in the MIT case if one would remove the no-motion words obtained from background subtraction.

Figure 5.23 presents the results of PLSM and the two other competitive methods. We observe that the simple HMM method gives the worst predictions on both datasets compared to the more sophisticated Topic-HMM, whose observations come from the PLSA topics. However, overall, the PLSM model gives a much better performance than the two HMM based methods, showing that the incorporation of temporal information at the topic level rather than at the global scene level is a better strategy. In the Far-field case, where the scene is not governed by any specific rules, PLSM

11. Note that when $c = 0$, the AVL measures dip to -9.2 and -4.6 for the Far-field and MIT datasets respectively, supporting the need for a smoothing term.

performs consistently and significantly better with an average likelihood 200% greater than that of the Topic-HMM when $n = 5$, and 90% greater when $n = 30$. Note that both methods improve as more motifs or states are used, but saturate beyond a value of 30.

On the MIT data, the situation is somewhat different. PLSM and HMM based approaches perform similarly when the motifs/states is $n = 5$. The HMM approaches have an advantage as they are able to model the different phases of the regular cycle governed by the traffic lights that the scene goes through. These distinct global behavior states and the transitions between them are captured explicitly in the Topic-HMM and to a lesser extent in the HMM method whereas our method does not have any prior on the sequences of motif occurrences. Nevertheless, PLSM provides a finer and more detailed description of the activities and its prediction accuracy improves consistently beyond the performance of the other methods that have difficulties to take advantage of the modeling of questionable and unpredictable sub-phase global scene activity patterns. Note however that the difference with the other models is not as high in this case as on the Far-field data, but the PLSM model still performs 35% better than the Topic-HMM. Finally, it is interesting to note that the prediction accuracy of the PLSM method tends to saturate for a number of motif N_z close to that selected using the BIC criterion (20 for the Far-field data, 26 for MIT data).

5.6 Audio Scene Analysis with Microphone array

The PLSM model can be applied to any multivariate time-series that can be described as word$\times$time counts. To test the generality of the model, we used it for analysing a scene using acoustic data. The setup used to capture and the features used to obtain temporal documents were explained in chapter 3. Here, we present the motifs obtained by applying PLSM on the TDOA data.

For the experiments, 30 recordings of approximately 20 seconds each were used, comprising a total of around 120 car passing events. Given the 80ms time step, each recording produced a temporal document of around 250 time instants with 25 possible words (angles). The PLSM approach was applied to these temporal documents, with the same MAP setting and sparsity level ($\lambda = 0.25$) as in the video case. However, given the known expected number of topics (four), we did not use the BIC criterion. The results are shown in Figure 5.24 when using a maximum length of 30 time steps ($\approx$ 2.5 seconds). Despite the noise and variations in vehicle speed (from around 35 to 70km/h), we observe that the dominant patterns are clearly captured: the ramp ones, corresponding to the car passing in front of the microphones; and the almost stationary motifs corresponding to cars approaching or leaving. Indeed, in this latter cases, azimuth angles are around $+90$ or -90 degrees and do not vary much. These activities get captured as separated motifs (from the ramp ones) because the measured duration of the "approach phase" is highly variable and depends on the sound volume of the car: a louder car will be perceived earlier by the microphones. Similar results were obtained when searching for motifs from 30 to 70 time steps. For instance, Figure 5.25 shows the results with a length of 60 time steps($\approx$ 5 seconds).

Figure 5.24. Four sequential motifs of 30 timesteps ($\approx$ 2.5 seconds) from TDOA data.

Figure 5.25. Four sequential motifs of 60 timesteps ($\approx$ 5 seconds) from TDOA data.

5.7 Conclusion

In this chapter we proposed a novel unsupervised approach for discovering activity motifs from multivariate temporal sequences. The PLSM model infers temporal patterns of a maximum time duration by modeling the temporal co-occurrence of visual words, which significantly differs from previous topic model based approaches. PLSM motifs include the temporal aspect and is therefore more complete in its representation. This is made possible thanks to the introduction of latent variables representing the motif start times, bringing the following advantages: a) implicit alignment of occurrences of the same motif while learning, and b) inference of the activity start times. The model parameters can be inferred efficiently using an Expectation-Maximization procedure that exploits a novel sparsity constraint. The effectiveness of our model was extensively validated using synthetic as well as real life data sets from both structured and unstructured scenes. When features from multiple cameras were fused and presented to PLSM, we could obtain motifs that span both the cameras wherever relevant and motifs restricted to single views in other cases. The synthetic data was primarily used to demonstrate several aspects of the model like the effect of noise, the sparsity constraint, the topic length and the number of topics effectively. Qualitative results and quantitative experiments on event detection and prediction tasks showed that the approach was discovering motifs consistent with the scene activities and was resulting in superior performance compared to other state of the art Dynamic Bayesian Network based alternatives. Although the method was demonstrated for activities in a video, we claim that it can have wide applications where sequential patterns need to be extracted. This claim was sufficiently demonstrated using data from a different modality like the TDOA data.

The model offers room for further improvements. For instance, although we have used the Bayesian Information Criteria measure to determine the number of topics, we still observe a few motifs (usually of lower $p(z)$) that are copies or minor variations of other motifs, which could hence be merged. This could be better dealt with by using other data driven approaches like Dirichlet Processes Teh *et al.* (2006) (please see Appendix B for an example). Finally, our model identifies

activities and their starting times, but does not discover higher-level knowledge on the motif occurrences. Modeling these occurrences in terms of dependencies or interactions could enhance the global understanding of the scene through, for instance, the identification of scene level rules (e.g. right of way) or activity cycles due to the presence of a traffic light.

Chapter 6

Mixed Event Relationship Model

6.1 Introduction

Data from sensor logs capturing human activities exhibit complex dependencies. Assumptions made in modeling and mining these long-term data logs should be as realistic as possible. Consider for example, the MERL sensor dataset [1], which is recorded using 200 Passive Infrared Sensors (PIR) over a period of one year. These wireless sensors detect movements of people and generate a single activation when a person or group moves in the neighborhood of the sensor. PIRs could in turn activate other sensors like a video camera to visually record the action or trigger appliances such as lights. Mining common motion patterns from such a dataset is an interesting problem for infrastructure management and understanding human behavior.

A similar case can be made with surveillance data considered in the previous chapters that monitor traffic and people activities in the view. Parallels could be drawn between PIR activations in the sensor log data and event detections from single or multi-camera scene videos. From surveillance videos we would like to understand the different activities going on in the scene, the dependencies between them and possibly the different phases of the scene with their unique characteristics.

In this chapter, we address this issue by proposing a novel model called the **M**ixed **E**vent **R**elationship **(MER)** model, that takes as input a binary event matrix whose entries indicate for instance the start of a fixed set of short-term temporal activities over time, and outputs both local rules (*e.g.,* the implicit rules such as the "right of way" that are followed, or the sequence of trajectory segments that a pedestrian has to follow to reach a destination in a multi-camera set-up) and global scene states for instance, determined by the traffic lights.

The rest of this chapter is organized as follows. In section 6.2, we first present an intuition of the model using an example, and then build the model from the basic ideas of dynamic Bayesian networks. Then we summarize the model with its generative perspective and inference. Following this, we present experimental details in section 6.3. Our results along with analysis and conclusions

1. http://www.merl.com/wmd/details.html

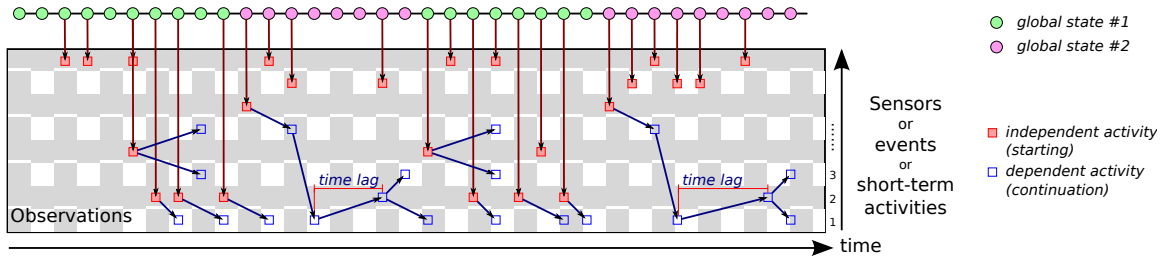


Figure 6.1. Data from sensor logs, an illustration – Observations are binary matrices: can be from event occurrences of short term activities or sensor activations. Each observation occurs either independently or as a continuation of a previous observation. A scene-level state controls the number of activities starting at a given instant, their type as well as whether they are independent or not.

are given in sections 6.4 and 6.5 respectively.

6.2 Model and Inference

In this section, we first introduce some of the important characteristics of activity data that we would like to model and proceed to build the model step by step. Finally, we will formalize the model with the generative process and the inference procedure.

6.2.1 Characteristics of activity data

Figure 6.1 shows a snapshot of a binary matrix of event occurrences. The event occurrences are given by red and blue boxes in the matrix. The figure also illustrates some typical characteristics of the data which are as follows:

- **Simultaneous occurrence of multiple events.** As shown in Figure 6.1, multiple events are activated simultaneously; for instance, in a traffic scene, people can be walking in the pedestrian area while vehicles stop or move in different directions of the scene simultaneously.
- **Scene states.** A scene can go through different states characterized by different activities. In Figure 6.1, the states are represented as colored circles on top and indicate that 2 states can happen. We can also see that the global state 1 mostly triggers activity 2. In a traffic scene controlled by lights, states can represent the different phases of the signal. The different phases can trigger different types of events and possibly a different number of events.
- **Dependent and Independent events.** Events generated could be of two types, which are labeled as independent and dependent. The independent events start on their own due to the nature of the state. In Figure 6.1, they are given as red boxes with an arrow from the state. For example, a static vehicle starting to move after a switch in traffic signal can be considered independent. The dependent events on the other hand are triggered by some event in the past and are indicated with blue boxes. The relation that caused one event to be triggered by another one in the past is indicated with an arrow between the events.
- **Variable temporal lags.** The temporal lag between every pair of related events cannot

be fixed, *i.e.,* a pair of events can co-occur with a variable temporal lag between them. In Figure 6.1, we see that event-2 often occurs after event-1 with a time lag of 3 to 5 seconds We can consider that there is an average temporal lag between a particular pair of events with some minor variability around the average. Therefore, these lags are not necessarily of first order, *i.e.,* they do not necessarily depend on the immediate past.

In our approach, we would like to model these general characteristics of activity data. We will give an idea of how this can be achieved by relaxing the assumptions of the existing models.

6.2.2 Building the model

To build the model that achieves our goals, let us first start by reviewing the basic ideas of a DBNs. Consider a discrete time series, *i.e.,* a sequence of observations where an observation at each instant of time takes a value from one of N_c possible values. Good examples are sequence of words in a sentence, or weather observations over a sequence of days (sunny, rainy and cloudy) as shown in Figure 6.2(a). This can be modeled using a stochastic process called a Markov chain. Here at each time instant t, we have a random variable c^t that takes one of the possible values or *states* depending on the random variables in the past, namely the set $\{c^{t-1}, c^{t-2}, \ldots, ..c^0\}$. Usually, we restrict the dependency to the observation in the immediate past, *i.e.,* $P(c^t|c^{t-1}, c^{t-2}, \ldots, ..c^0) = P(c^t|c^{t-1})$, resulting in a Markov chain of the first order. Here, the jump from one state to another in the sequence, $P(c^t|c^{t-1})$ is probabilistically given by the transition parameters π, a set of N_c multinomial distributions. In a Bayesian formulation, every parameter is sampled from a prior distribution. In this case, since the transition probabilities form a discrete distribution, they are sampled from a Dirichlet distribution with hyper-parameter φ as shown in in the graphical model in Figure 6.2(c).

But often, we do not have access to the true state value but only observe some characteristics of the true state. Such observations and states are modeled using the Hidden Markov Models (HMM) as shown in Figure 6.3(a). Now we need two sets of random variables; let c^t represent the state of the system at time t that is hidden and O^t, the observation from c^t, encircled in a shaded node. Note that the arrows connecting the hidden nodes is similar to the simple Markov chain we saw in Figure 6.2, except that it is now a hidden state capable of generating observations. Invoking the properties of conditional independence, we can say that the observation at time t, O^t is conditionally independent of past states c^{t-1} or past observations O^{t-1} given the current state c^t of the system. The observations are modeled as a function of the hidden state parameters. Here, we will assume that the observations are discrete taking one of the N_z values depending on the hidden state. Similarly, the transitions can be modeled using a Discrete distribution parameterized by θ_k for each state k, and are called the emission distributions. Again to make this consistent with the Bayesian formulation, the emission parameters $\{\theta_k\}_{k=1}^{N_c}$ are sampled from a Dirichlet prior distribution $\mathrm{Dirichlet}(\alpha)$ resulting in the model given in Figure 6.3(b).

In the HMM model, we assumed that there is a single observation generated by the state at an instant. But activity content of a scene can vary over time. We need to accommodate a variable

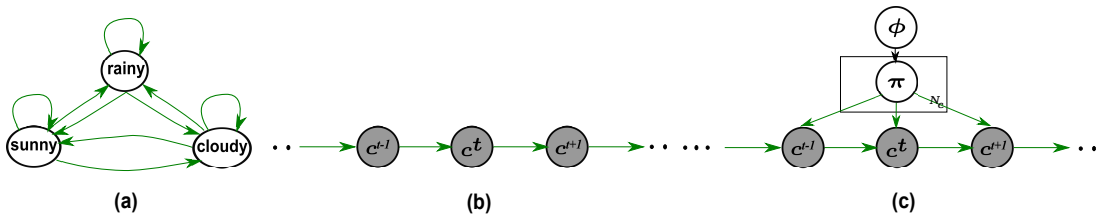


Figure 6.2. Three different representations of a Markov model. (a) Pictorial representation of a chain system. (b) A first order Markov model unfolded in time, (c) The model in (b) with multinomial parameters and Dirichlet priors in Bayesian view.

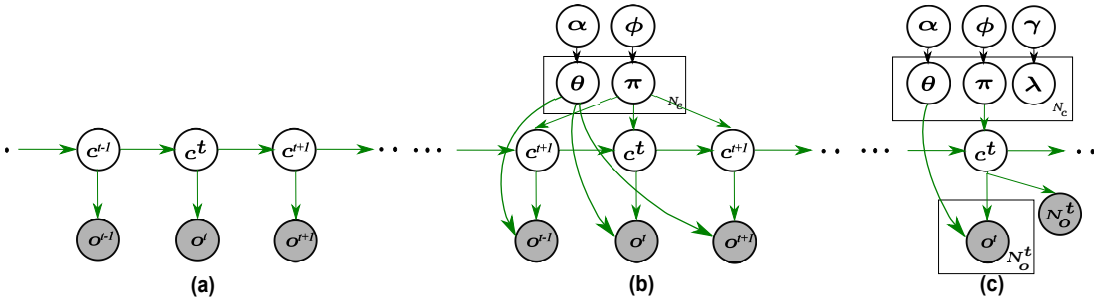


Figure 6.3. Hidden Markov Model variants: (a) Conventional representation of HMM. (b) Bayesian view of HMM with multinomial transitions and emission distributions with their respective Dirichlet priors. c) HMM generating variable number of observations with a Poisson parameter and a Gamma prior.

number of events, and indeed each state might be charecterized by different levels of activities. This variability in the number of observations can be modeled using a Poisson distribution [2] which gives an integer value based on a parameter λ. To accommodate this variability, we add a Poisson parameter λ_k into each HMM state k. Then the observations for any instant are obtained as follows: draw the number of observations $N_o^t|k \sim \text{Poisson}(\lambda_k)$; then draw the observations $\{O_i^t\}_{i=1}^{N_o^t}$ as $O_i^t|k \sim \text{Discrete}(\theta_k)$. This is shown in Figure 6.3(c), where we adopt a Bayesian approach and add a Gamma prior to the Poisson parameter. The Gamma distribution parameterized by $\gamma = \{\gamma_1, \gamma_2\}$ [3] being a conjugate distribution to Poisson distribution, becomes the natural choice for the prior distribution. This results in a mathematically convenient solution for the posterior parameters.

Now we have an HMM that can generate a variable number of discrete observations. As illustrated in Figure 6.1, there are both indepedent and dependent events, where the dependent events are triggered by an event in the past. To model this phenomenon, we associate a selector variable sampled from a Bernoulli distribution to each event. A sample from the Bernoulli distribution takes a value of 1 or 0. Therefore, the sample values are used to decide if an event should be independent or not; in one case, the event is generated from a state-dependent multinomial and in another case, the event is generated from some past event. This entire process is more formally described in the following parts of the section.

2. Poisson distribution is used to express the probability of a given number of events occurring in a fixed interval of time and/or space.

3. The hyper-parameters also have a real-word interpretation that γ_1 indicates the mean count of observations in γ_2 intervals.

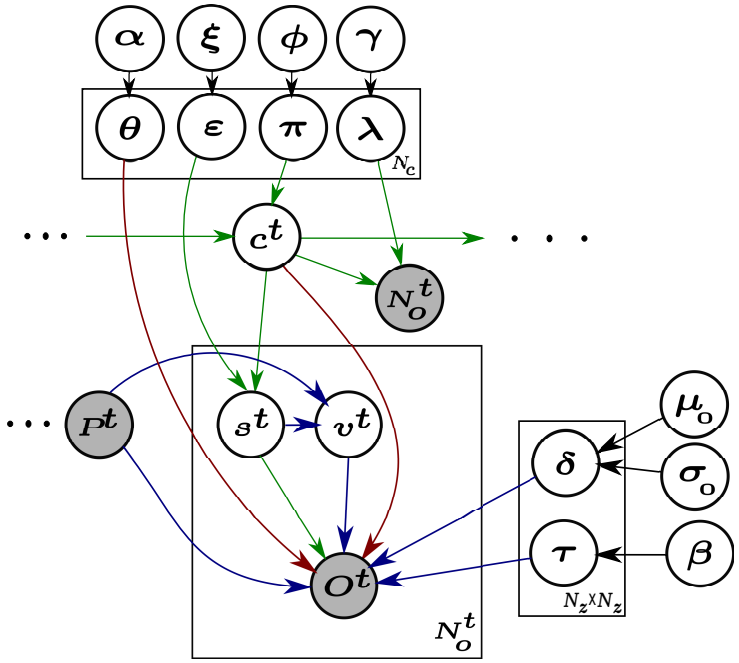


Figure 6.4. Graphical Model – green: links to generate all activities, blue: links to generate dependent activities, red: links to generate independent activities.

6.2.3 Generative Process

In this section we describe the generative process of the model in detail. The notations used in the model are given in Table 6.1 and the graphical model is shown in Figure 6.4. Our data $\mathcal{O}$ is the binary event matrix with T_d columns indicating time, and N_z rows indicating the fixed set of events. Every non-zero entry O_i^t, in the matrix represents an event *i.e.,* the start of an activity i at time t. The generative process consists of three parts: i) generating parameters for the global scene states; ii) generating parameters for event relations and iii) generating the events.

Global scene states. We assume that the scene passes through N_c possible states, *i.e.,* at each instant t, the scene is in a state c^t among the N_c possible ones, depending on the previous state c^{t-1}. The state c^t controls the number of observations that can occur using a Poisson distribution with parameter λ_{c^t}, the proportion of dependent or independent events using a Bernoulli distribution with parameter ϵ_{c^t}, the set of independent events using a Multinomial distribution with parameter θ_{c^t} and and its transitions from c^t to c^{t+1} using the transition parameter π_{c^t}. Poisson, Bernoulli, Discrete and the transition parameters are sampled from prior distributions: $\mathrm{Gamma}(\gamma_1, \gamma_2)$, $\mathrm{Beta}(\xi_0, \xi_1)$, $\mathrm{Dirichlet}(\alpha)$ and $\mathrm{Dirichlet}(\varphi)$ respectively. The choice of the prior distributions is due to their conjugate properties that result in mathematical tractability.

Local event relations. The local event relations determine which event occurs after which other event. Since the dependency can be attributed to an event that occurred not just in immediate past but farther back in time, it needs to be captured using two entities: i) a Discrete distribution

Symbol	Description
α	Dirichlet prior on independent activities
β	Dirichlet prior on motif transitions
μ_0, σ_0^2	Hyper-parameters of Gaussian prior
$\gamma = \{\gamma_1, \gamma_2\}$	Hyper-parameters of Gamma prior
$\xi = \{\xi_0, \xi_1\}$	Hyper-parameters of Beta prior
φ	Dirichlet prior on state transitions
$\tau_{z'}(z)$	Transition from event z' to z
$\delta_{z',z}, \sigma^2$	Mean and a fixed variance on time lag between z' and z
θ_k	Distribution over activities set, for each state k
λ_k	Poisson parameter to select N_o^t, for each state k
ϵ_k	Bernoulli parameter to select s_i^t, for each state k
π_k	Global state transitions, for each state k vsp
N_o^t	Number of activities at time t
P^t	Set of past events for time t
N_p^t	Number of past events for time t
T_k	Number of time instants explained by state k
T_l	Maximum lag for event associations

Table 6.1. Notations used in this chapter

$\text{Discrete}(\tau_{z'}(.))$ for each past observation of type z' to capture the transition between events, and ii) a Gaussian distribution with mean $\delta_{z',z}$ on temporal lags for each possible transition from z' to z. The transition and lag parameters are obtained from $\text{Dirichlet}(\beta)$ and $\mathcal{N}(\delta_{z',z}|\mu_0, \sigma_0^2)$ respectively.

Sampling events. At each time t, the number of events N_o^t is sampled from a Poisson(λ_{c^t}). For each of event O_i^t, we associate a binary selector variable $s_i^t \in \{0, 1\}$ sampled from Bernoulli(ϵ_{c^t}). The selector variable s_i^t decides if the event is generated depending on one of the past occurrences or independently. When $s_i^t = 1$, we rely on the current state to start an independent event and sample O_i^t from Discrete(θ_{c^t}). When $s_i^t = 0$, we decide O_i^t to be dependent and associate it with another variable v_i^t which indicates one of the past events from the set $P^t = \{O_i^{t'}\}_{i=1,t'<t}^{N_p^t}$ Practically, we limit the dependency to a fixed temporal extent in the past $t - T_l \leq t' < t$.

In the generative process, we assume that when the decision variable is 0, there is at least one event in the past that would take responsibility of generating the current event. More precisely, when $s_i^t = 0$, we assume that $P^t \neq \emptyset$ and that there exists a past event z' such that $\tau_{z',z} \neq 0$. When $P^t = \emptyset$, we expect that such a state of the scene is captured in the global state variable hence generating only $s_i^t = 1$. Please see the inference derivation in Appendix C on how this issue is addressed by jointly sampling s_i^t, v_i^t.

A similar use of binary decision variables can be seen in the text mining domain, where it is used to decide if words need to be associated to form phrases (Griffiths *et al.*, 2007; Wang *et al.*, 2007) or not. However, unlike in text, where there is a single observation at any time, videos have multiple event occurrences at any instant with large temporal variations making this association a

1. for each $k = 1, \ldots, N_c$ global states;
   (a) draw Poisson parameter $\lambda_k \sim \text{Gamma}(\gamma_1, \gamma_2)$
   (b) draw Bernoulli parameter $\epsilon_k \sim \text{Beta}(\xi_0, \xi_1)$
   (c) draw Multinomial parameter $\theta_k \sim \text{Dirichlet}(\alpha)$
   (d) draw state transitions $\pi_k \sim \text{Dirichlet}(\varphi)$
2. for each event type $z \in [1..N_z]$, draw transitions $\tau_z \sim \text{Dirichlet}(\beta)$;
3. for each event pair $z', z \in [1..N_z]^2$, draw lags $\delta_{z',z} \sim \mathcal{N}(\delta_{z',z}|\mu_0, \sigma_0^2)$
4. for each t in $1, \ldots, T_d$
   (a) draw $c^t \sim \text{Discrete}(\pi_{c^{t-1}})$
   (b) draw a number $N_o^t \sim \text{Poisson}(\lambda_{c^t})$
   (c) for each i in $1, \ldots, N_o^t$
   – draw a binary value $s_i^t \sim \text{Bernoulli}(\epsilon_{c^t})$
   – if $s_i^t = 1$, draw $O_i^t \sim \text{Discrete}(\theta_{c^t})$,
   – if $s_i^t = 0$, draw $v_i^t|P^t \sim \text{Uniform}(\frac{1}{N_p^t})$, where P^t is the set of past events $P^t(v_i^t) = z'$ is the past event occurring at t' upon which O_i^t depends on.
   – draw $O_i^t \sim \mathcal{N}(t - t'|\delta_{z',z}, \sigma^2) \cdot \text{Discrete}(\tau_{z'})$, where $\delta_{z',z}$ is the mean lag between z' and z. σ^2 is a fixed variance.

Figure 6.5. The Mixed Event Relationship model generative process

complex problem. It is also interesting to note that several types of Hidden Markov Model (HMM) can be derived from our MER model. For instance, when the number of activities N_o^t is 1, it reduces to a kind of HMM where observations at each time instant can be either dependent or independent of the current state. When s_i is always set to 1, it reduces to the standard HMM, where the states generate all the observations from a state specific Discrete distribution.

6.2.4 Model Inference

Hierarchical Bayesian models like LDA, HDP, HDLSM result in mathematical intractable forms for the posterior distributions. Therefore, approximate inference methods like variational Bayes and MCMC methods are popularly used to infer the latent variables of the model. A particular inference method that is of interest to us is the collapsed Gibbs sampling procedure (Griffiths and Steyvers, 2004) due to its simplicity and faster convergence. The collapsed Gibbs sampling method *integrates out* (collapses) the parameter(s) resulting in a smaller set of variables to be estimated. In other words, collapsing implies sampling the latent variables and skipping the steps of sampling the parameter(s). The parameters are later inferred from the latent variable samples after the burn-in period. This collapsing step results in a smaller set of parameters, faster convergence of the posterior distribution and easier implementations (Liu, 1994).

Exact inference for the MER model is intractable. But thanks to conjugate pairs like Poisson-

Gamma, Dirichlet-Multinomial and Beta-Bernoulli and Normal-Normal, it is possible to derive a collapsed Gibbs sampling algorithm by integrating out the parameters $\{\pi, \lambda, \epsilon, \theta, \tau, \delta\}$ (please see Appendix D for more details on their conjugacy properties). The algorithm proceeds by iteratively sampling the decision variable s_i^t, and indicator variable v_i^t for each observation O_i^t conditioned on all other variables, parameters and hyper-parameters. The state indicators c^t are sampled for each time instant conditioned on the rest of the variables. The complete derivation of the sampling equations is given in Appendix C. Below, we give the main idea behind the inference process.

Since each occurrence also gives its occurrence time implicitly, we will drop the t associated with s_i^t and denote it by s_i except in places where time needs to be mentioned explicitly. We will also use O, S, V to refer to the set of occurrences, their corresponding selector variables, and indicator variables. $O_{\neg i}, S_{\neg i}, V_{\neg i}$, will indicate all the occurrences, selector variables and the indicator variables except the i^{th} one and hp refers to the set of hyper-parameters set: $\{\varphi, \gamma, \xi, \alpha, \beta, \mu_0, \sigma_0^2\}$.

For a given observation i, we re-sample s_i and v_i jointly. For the case where $s_i = 1$ (independent event), v_i is not meaningful and we obtain the following sampling probability:

$$p(s_i = 1, v_i | O_i^t = z, c^t = k, S_{\neg i}, V_{\neg i}, O_{\neg i}, C_{\neg i}, hp) \propto \frac{l_{-i,k}^{(1)} + \xi_1}{l_{-i,k}^{(.)} + \xi_0 + \xi_1} \cdot \frac{q_{-i,k}^{(z)} + \alpha}{q_{-i,k}^{(.)} + N_z \alpha} \tag{6.1}$$

In the equation (6.1), $q_{-i,k}^{(z)}$ is the count of motif z occurring with state k when $s_i^t = 1$ removing the current observation. Similarly, $l_{-i,k}^{(1)}$ and $l_{-i,k}^{(0)}$ are the counts of $s_i = 1$ and $s_i = 0$ appearing with state k after omitting the current observation. Qualitatively, the probability of $s_i = 1$ is based on two factors; the first term on the right depends on how often an independent event is selected from the state k, and the second term depends on how often the event z is associated with the state k.

In cases where an observation depends (*i.e.,* $s_i = 0$) on the j^{th} one in the past (*i.e.,* $v_i = j$) that is of type z', we have the following sampling probability:

$$p(s_i = 0, v_i = j | O, S_{\neg i}, V_{\neg i}, c_i = k, C_{\neg i}, hp) \propto \frac{l_{-i,k}^{(0)} + \xi_0}{l_{-i,k}^{(.)} + \xi_0 + \xi_1} \frac{r_{-i,z'}^{(z)} + \beta}{r_{-i,z'}^{(.)} + N_z \cdot \beta} \mathcal{N}\left(f(P^t(j), O_i^t) | \delta_{z',z}, \sigma\right), \tag{6.2}$$

where $r_{-i,z'}^{(z)}$ is the count of event z appearing after z' and $f(P^t(j), O_i^t)$ is the temporal delay between $P^t(j) = z'$ and $O_i^t = z$. The mean temporal lag for each pair of events $\delta_{z',z}$ is the posterior estimate obtained from all associations made between z' and z during the inference, except the current one and σ^2 is a fixed variance in the lag. The equation (6.2) which gives the probability of $s_i = 0$ is based on three terms mainly. The first term on the right side depends on how often a dependent event is observed under state k, the second term gives the number of times a transition from z' to z is made, and the third term gives the probability of the current time lag $t - t'$ for the pair z' to z under the Gaussian distribution with the mean lag $\delta_{z',z}$ and a variance σ^2. These two equations (6.1,6.2) show that there are two competing factors to explain an event. The event becomes independent or dependent based on which one of the two factors explain the observation better.

For re-sampling the global state $c^t = k$ at time t, the sampling depends on four factors coming

from the links in the graphical model. The terms are: a) the number of transitions made from c^{t-1} to c^t and from c^t to c^{t+1}, b) the chances of observing the current number of observations N_o^t under the state k, c) the chances of observing the current proportion of independent and dependent events, *i.e.,* the number of times we see a $s_i = 0$ or 1 under the state k, and lastly d) the number of times the independent event types of the current time instant have co-occurred with the state k. These factors are relatively complicated and provided in Appendix C, equation (C.24).

6.3 Experimental setup

Datasets – To evaluate the MER methods we experimented on 4 different video datasets that were presented in chapter 3. The MIT dataset (Wang *et al.*, 2009), the QMUL Junction dataset (Hospedales *et al.*, 2009), the Far-field dataset (Varadarajan *et al.*, 2010) and the ETH dataset (Kuettel *et al.*, 2010) were used. In the following section we explain how the binary event matrix is obtained and then present the results.

Video to Events. In practice, the MER model can be made to work with count matrices with some adaptation. But in our implementation, we used a sparse binary event matrix as input where the non-zero entries correspond to activity starts (events). To obtain this, we used the PLSM model proposed in chapter 5, that takes a temporal document as input and produces motifs and their start times as output. More precisely, the video was converted to a single temporal document as described in section 5.4.1. By applying PLSM on the temporal document, we obtained dominant activities of the scene as $P(w, t_r|z)$ and their start times as $P(z, t_s|d)$ or simply $P(z, t_s)$ (as we have a single document now). Using a small threshold th on the starting time probabilities $P(z, t_s)$, we create the binary event matrix B where the entries $B(z, t_s) = 1$, if $P(z, t_s) \geq th$ and $B(z, t_s) = 0$ otherwise. The threshold was selected by assuming that confident event detections should be more than the uniform value *i.e.,* $1/(N_z \times T_d)$, which is a small value. We obtained 25, 20, 20 and 15 motifs from the MIT, QMUL Junction, ETH and Far-field datasets, all with a fixed maximal duration of 5 or 10 seconds. Since the motifs from MIT and Far-field datasets were sufficiently demonstrated in the previous chapter 5, we show some sample motifs from QMUL Junction and ETH-Zurich datasets and the binary event matrices obtained from them.

Sample events from QMUL Junction dataset. Figure 6.6 shows 5 sample events out of the 20 motifs obtained from the QMUL Junction video by applying PLSM. These motifs are 5 seconds long. The motifs are represented using the same scheme described in section 5.4.2. The motifs represent traffic activity: a) from left to right, b) from bottom left to top left, c) from top right to bottom right d) from bottom of the scene towards right and, e) from right to left. This is captured from UK where people drive on the left.

In Figure 6.7, we show the binary event matrix from 20 PLSM motifs over a 5 minute segment of QMUL junction video, obtained by the procedure described just before. Dark locations indicate presence of an event corresponding to that row at the time corresponding to the column. In this ma-

(a) (b) (c) (d) (e)

Figure 6.6. 5 sample motifs out of the 20 motifs obtained by applying PLSM on QMUL junction video. Time within the activity is color-coded from violet to red. Locations of violet show the initial positions of the event and red shows the final location of the event.

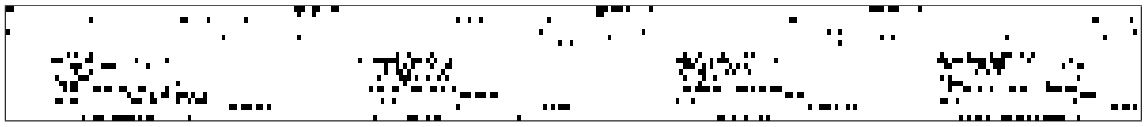


Figure 6.7. Binary event matrix of 300 seconds from QMUL Junction data, obtained by thresholding the $p(t_s, z|d)$ distribution. Rows indicate motifs and columns indicate time. We can observe a pattern in the matrix showing a periodic phenomenon.

trix we can observe periodic epochs of events caused by the traffic signals. Within each epoch, there are self repeating events, for instance, the motifs in Figure 6.6(b,c) are often triggered repeatedly due to vehicles passing one after the other within a cycle.

Sample events from the ETH-Z dataset. Figure 6.8 shows eight sample motifs out of the 20 motifs obtained from ETH video by applying PLSM. These motifs are 5 second long. The motifs are represented using the same scheme described in section 5.4.2. The events represent a variety of activities due to vehicles, trams and pedestrians. For example, Figure 6.8(a,b) show trams passing from top right to bottom left and vice-versa. Figure 6.8(c,d) show pedestrian crossing the road, and Figure 6.8(e–h) show vehicles moving along the curved road from top left to top right. Figure 6.9 shows a binary event matrix from a set of 20 events, over 300 seconds of the ETH video. Here again, we can see periodic epochs and self repeating activities due to the traffic signals. The short stretches of activations along the rows are due to self repeating activities. We will see that this property is captured by the MER model as self loops in Figure 6.15.

Parameter Settings. As we have no a-priori information on any of the parameters, we used non-informative symmetric Dirichlet distributions for $\{\alpha, \varphi, \beta\}$. Similarly, we set equal values to the Beta hyper-parameters $\xi_0 = \xi_1 = 1$ and the same follows for the Gamma distribution hyper-parameters too. Since our motifs are 5 second long, we can expect dependent events to be triggered after an average temporal lag between 1 to 10 seconds. So the prior for the mean temporal lag was given by a Gaussian distribution with a wide spread, *i.e.,* with $\mu_0 = 5$ and $\sigma_0^2 = 4$. The maximum lag T_l, *i.e.,* maximum temporal lag between dependent events is set to twice the duration of activities. We used 90% of the duration of each video mentioned for training our MER model. Gibbs sampling was run for a sufficiently long number of iterations (1000 for each of them), where each iteration consists of sampling the selector variable s_i^t and the indicator variable v_i^t for all the observations

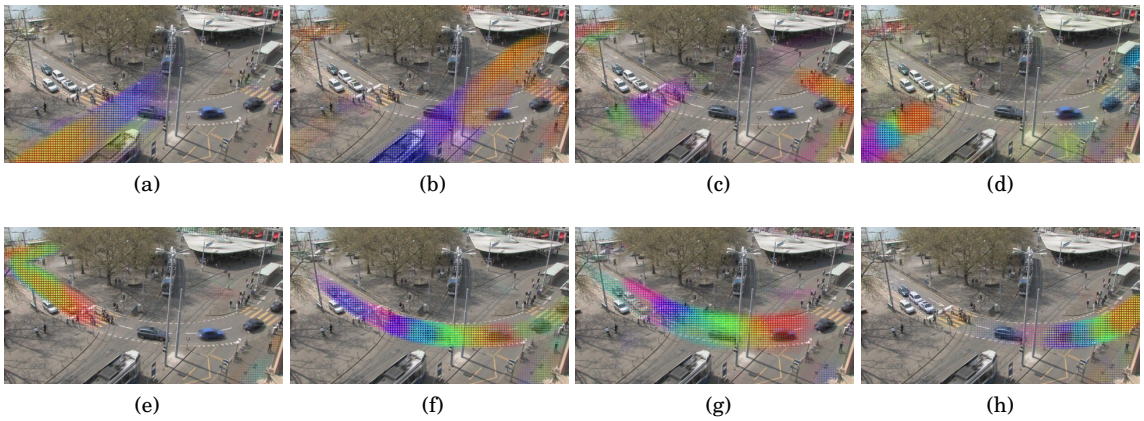


Figure 6.8. Eight sample motifs out of 20 obtained from ETH-Z video by applying PLSM. Time within the activity is color-coded from violet to red. Locations of violet show the initial positions of the event and red shows the final location of the event.

Figure 6.9. Binary event matrix of 300 seconds from ETH-Z data. Obtained by thresholding the $p(t_s, z)$ distribution. Rows indicate motifs and columns indicate time.

and the state variable c^t for all the time instants. However, in all our experiments we found that the sampler reached stationarity after 600 iterations.

6.4 Results

In this section we present some of the global scene level rules as well as local rules obtained by applying the MER model.

6.4.1 Global rules

In our model the global rules are characterized by the scene-level states, their transition probabilities, and their attributes. These correspond to the parameters $\{\pi, \theta, \epsilon, \lambda\}$. Here, we show global rules obtained from QMUL Junction and MIT datasets.

QMUL Junction dataset. With a-priori knowledge about the scene activities, we applied MER model to extract 4 global states from QMUL Junction dataset. The colored pattern (blue, cyan, yellow and red) in Figure 6.10 top, shows the inferred state sequence of the scene over 10 minutes of video. Below the state sequence we see the top ranking independent activities of the four states. These activities correspond well to vehicles moving in the vertical directions (blue), moving from bottom and turning towards the right (yellow), from left to right (red) and right to left (cyan) respectively; in summary, they form the main phases of the cycle occurring in the scene. The repetitive

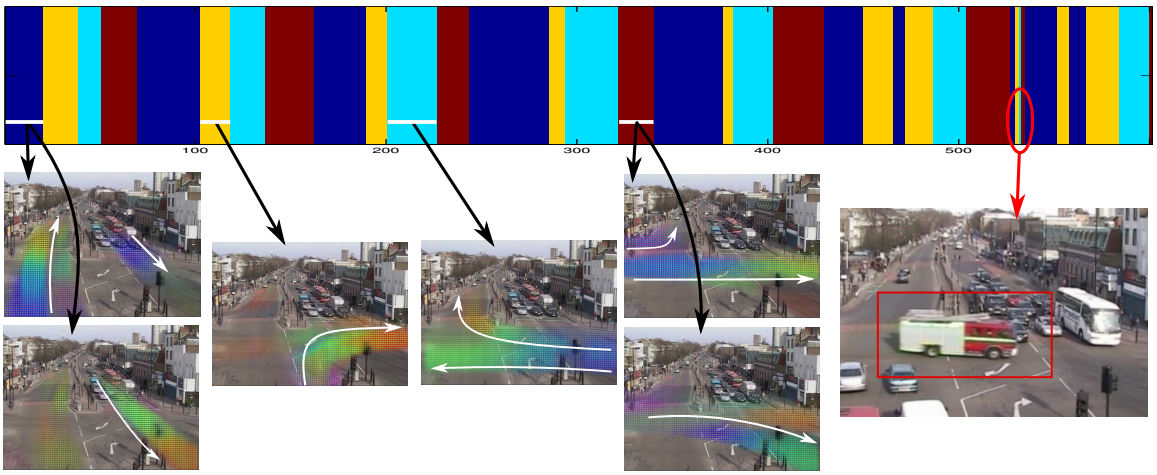


Figure 6.10. Four scene states obtained by our model from the QMUL Junction dataset. The four states represented by the colors blue, cyan, yellow and red are plotted for 600 seconds of video. Lower part: dominant activities found to occur at each of the 4 states. Red ellipse: an abnormality where an emergency vehicle interrupts the traffic momentarily.

pattern of colors (states) clearly shows that our model properly captures the periodic cycles of the scene that are due to traffic lights. A notable exception in the periodic repetition of the states is circled in the right part of Figure 6.10. This is an unusual event when a fire engine crosses from left to right, freezing all other traffic.

MIT dataset. On the MIT dataset, we used 2 states in our model to produce the results shown in Figure 6.11. We also experimented with 3 and 4 states and obtained comparable results with a finer segmentation of the scene activities. The two states inferred by our model correspond to the two main periods in the traffic cycle. The first state (in blue) corresponds to vehicles moving in the top-down directions and vehicles waiting on the horizontal lanes. Similarly, the second state (in red) captures vehicles moving in the horizontal directions, while vehicles on vertical lanes are waiting (only one of these activities is shown due to space restrictions). We see that the scene periodically follows the alternating traffic signals, except for some rare glitches.

6.4.2 Local rules

Our model captures local rules in the form of transitions from one activity to another, with a time lag and fixed-variance Normal distribution. We can represent the set of local rules in the form of a graph with (blue) edges annotated with a weight indicating the number of transitions between the edges and the mean temporal lag $\delta_{z',z}$ information as done for example in fig. 6.12 (detailed later). On such a figure, we can also show the number of times an activity was generated as an independent occurrence with some sourceless (red) edges. As independent activities can be caused by different scene-level states, the sourceless edges are annotated with the scene-level state (c_i) that cause them. For space and readability reasons we filtered out low frequency edges in the illustrations using a threshold count of 30.

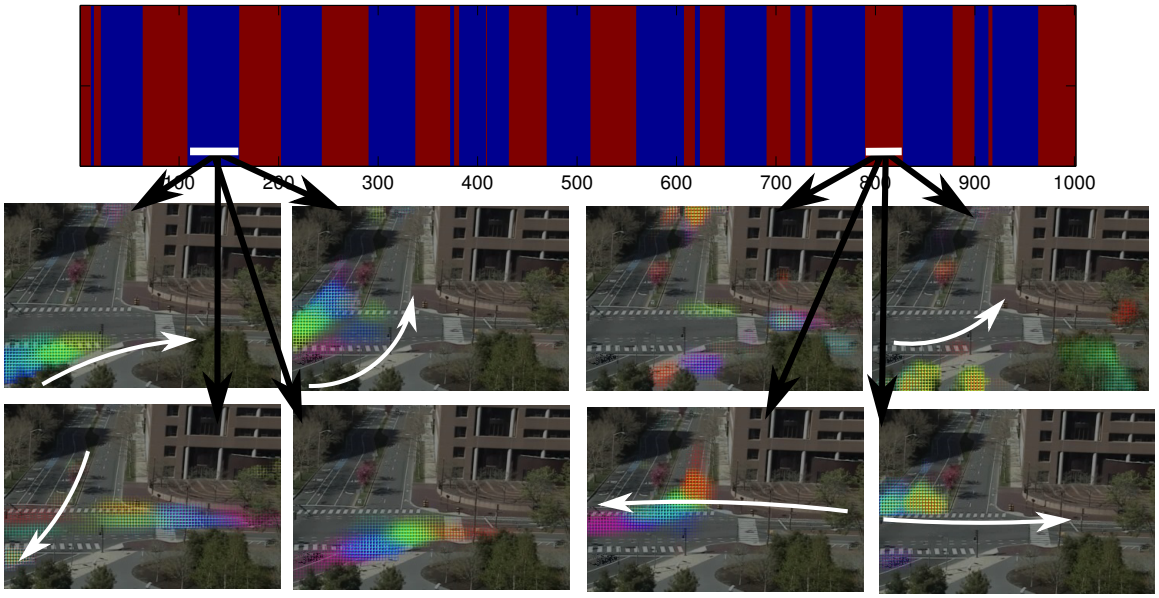


Figure 6.11. Two scene states obtained from MIT data shown for 1000 seconds. The two states correspond to distinct periods of the traffic signal cycle. The time indicated in blue correspond to motion in the top-down directions. The red region correspond to left-right movements.

Far-field dataset. Figure 6.12 shows the graph of activities recovered by our model for the Far-field dataset with PLSM motifs of 5 seconds duration. Since there are no traffic signals, we used a single global state. In this dataset, 13 of the 15 activities have been found to have notable dependencies. Interestingly, Figure 6.12 properly exhibits two independent sub-parts in the graph. The two subgraphs correspond to the two main vehicle directions. The upper part of Figure 6.12 shows a chain a-b-c-d-e for a vehicle coming from the right and disappearing on the top of the image. Only activities a and b have been found to be spontaneous starts of activities. The sum of the lags δ of each subgraph also gives the duration of the full trajectory of a vehicle in the scene which is around 10 to 15 seconds. To account for some vehicles slowing down or stopping at this location due to a single-lane tunnel on the top right, we see that d and e activities have self loops. To account for possible variations in vehicle speed, some "shortcut" transitions are also captured. For example, we see that it is common to do directly b-d in 5.4 seconds instead of doing b-c-d in 6.1 seconds. The lower part of Figure 6.12 captures the other direction for cars. Starts are spread on the three activities f, g, and h. The loop on state f (same for g and h) is explained by the scene: cars coming from the top tend to group together as there is a single-lane tunnel where cars cannot cross and have to wait for cars from the other direction to finish passing.

Figure 6.13 shows the local rules from Far-field data when the event matrix was created from 10 second long PLSM motifs. Here again, we obtain three main spontaneous starts of events: The sequence a-b-c-d) indicates activity starting from the top right and moving towards the bottom right. The sequence in Figure 6.13(e-f-h-i) indicates activity starting from bottom right and moving

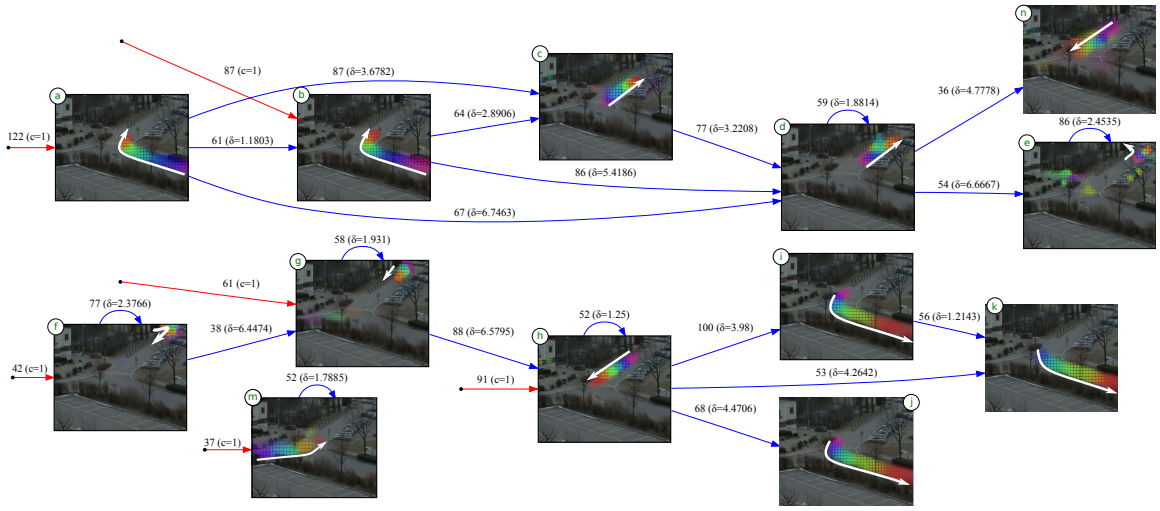


Figure 6.12. Event relationships from Far-field data with 5 second long motifs. Low frequency edges are not shown (those with counts below 30). Blue edges: transitions with weight and lag δ. Red edges: independent activity with weight and causing state (here with only 1 state). 80% of the occurrences were dependent activities.

towards top right. The sequence in Figure 6.13(g-h-i) indicates vehicles starting from the middle left and moving towards top right. Interestingly, this shares the trajectory h-i with the previous activity. One might notice that here we have longer activity segments which are approximately 10 second and therefore, fewer transitions when compared to Figure 6.12. But the results are consistent despite the change in the motif durations. We also see an isolated motif at the top. The large spatial support indicates that it is an event due to large trucks. Large trucks usually slow down to make the turn which explains the self loops.

MIT dataset. On the MIT dataset, 24 out of 25 activities have been found to have dependencies. Activities with high dependencies are shown in Figure 6.14. Among them, the six activities on the right corner of the figure are only self dependent and mostly correspond to stopped cars. Dependent activities Figure 6.14(j-k) also correspond to trees waving due to wind. Three activities (lower-right corner) are found to be starting and self-looping: they all correspond to examples of cars starting. A car starting is an activity that repeats itself because multiple cars are usually waiting for the green light and start successively. In addition to repeating static activities, the model captures trajectories made of multiple activities such as h-i and d-e-c. Some interesting soft rules are also captured like in Figure 6.14(f-g), where vehicles coming from the left Figure 6.14(g) often turn only after vehicles coming in the opposite direction have passed.

ETH-Zurich dataset. The transitions from ETH dataset is presented in Figure 6.15. Experiments were run with 20 motifs of 5 seconds each, and asking for 2 global states from the MER model. We can observe that the two states capture the two dominant movements in the scene. The state c1 corresponds to tram movements and people crossing in the diagonal directions (bottom left - top right). The motifs labeled Figure 6.15(j,k) show this. The self loops show that groups of people cross the road probably following each other. The other state c2 corresponds to cars moving along the

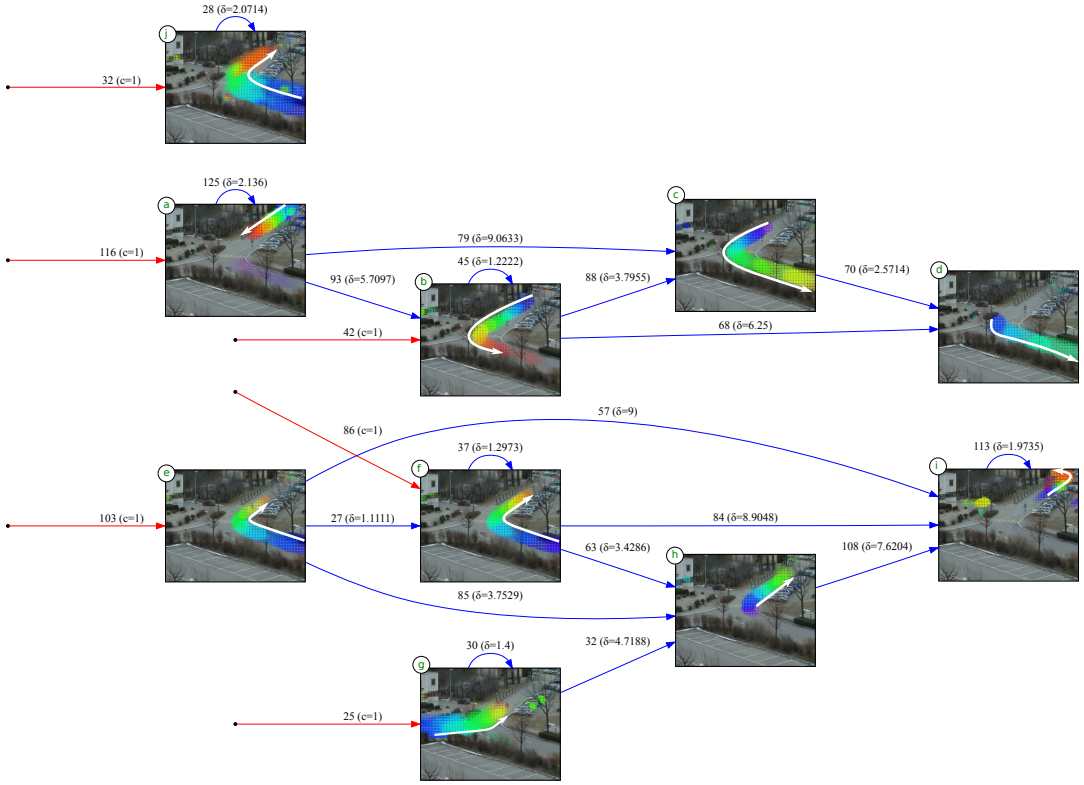


Figure 6.13. Event relationships from Far-field data with 10 second long motifs. Low frequency edges are not shown (those with counts below 30). Blue edges: transitions with weight and lag δ. Red edges: independent activity with weight and causing state (here with only 1 state). 80% of the occurrences were dependent activities.

curved road (from top-left to top-right) in the scene. One such track is Figure 6.15(i-d-g-h). We also see that after the event labeled Figure 6.15(c) generated by state c1, events usually generated by state c2 can follow, implying that in some cases the vehicles pass through the road only after people have crossed the road. This is quite evident when we look at the connection in Figure 6.15(c-g-h) where the vehicles have been waiting for people to finish crossing and start immediately after that. Here again, we observe self loops on most of the motifs indicating number of cars following each other through the curved path.

6.4.3 Numerical evaluation on a prediction task

To objectively measure the performance of MER model we used the prediction task proposed in section 5.5.4. Recall that our task is to estimate the probability $P_t^{pred}(w)$ that a word w appears at time t given all past information, that is, given the temporal document $n(w, t_a, d)$ up to time $t_a = t - 1$. As discussed in section 5.5.4, a word at time t can occur due to either a motif that has already started at a past time $t_s \in [t - T_z + 1, t - 1]$, or due to a motif that starts at the same time t.

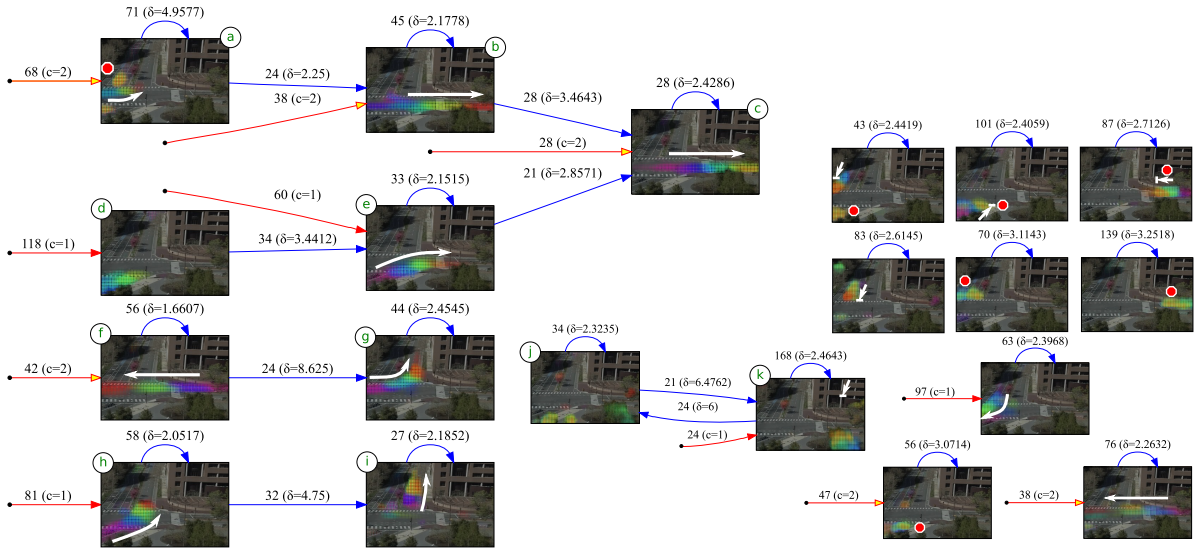


Figure 6.14. Event relationships from MIT dataset. Low frequency edges with counts below 20 are filtered. Blue edges: transitions with weight and lag δ. Red edges: independent activity with weight and causing state. 75% of the occurrences were dependent activities.

The prediction is given by:

$$P_t^{pred}(w) \propto \sum_z \left((1-\gamma) \sum_{t_s=t-T_z+1}^{t-1} \hat{P}(t_s, z|d) P(w, t-t_s|z) + \gamma \cdot P_t(z) P(w, 0|z) \right), \tag{6.3}$$

where $\hat{P}(t_s, z|d)$ denotes our estimation that the motif z starts at time t_s given the observed data, γ represents the probability that a topic starts at the current instant.

The PLSM model has a limitation in predicting which motif starts at time t based on past observations. Therefore, in the above equation (6.3), we have a blind motif prior $P_t(z)$, which is the marginal probability of a motif starting at any time t. This blind motif prior probability is estimated from training data by simply normalizing the number of times each motif appears in the temporal document. Given the global rules and inter-event relations learned from MER, we can refine this term further to give a better prediction.

Prediction using MER and PLSM. To estimate $P_t(z)$ using MER, we consider the motif start times $\hat{P}(t_s, z|d)$ within the sub-window $t_s \in [t-T_z+1, t-1]$ and create a test binary event matrix required for MER. Then, by fixing the MER model parameters learned from a training set, and using the test binary event matrix, we follow the MER generative process to create events at time t. The proportion of event counts generated by MER is then used for $P_t(z)$ in the prediction formulation. Indeed, this process can be repeated by appending $P_t(z)$ to $\hat{P}(t_s, z|d)$ to estimate $P_{t+1}(z)$, given by:

$$P_{t+1}^{pred}(w) \propto \sum_z \left((1-\gamma) \sum_{t_s=t-T_z+2}^{t-1} \hat{P}(t_s, z|d) p(w, t-t_s|z) + \frac{\gamma}{2} P_{t+1}(z) P(w, 0|z) + \frac{\gamma}{2} P_t(z) p(w, 1|z) \right) \tag{6.4}$$

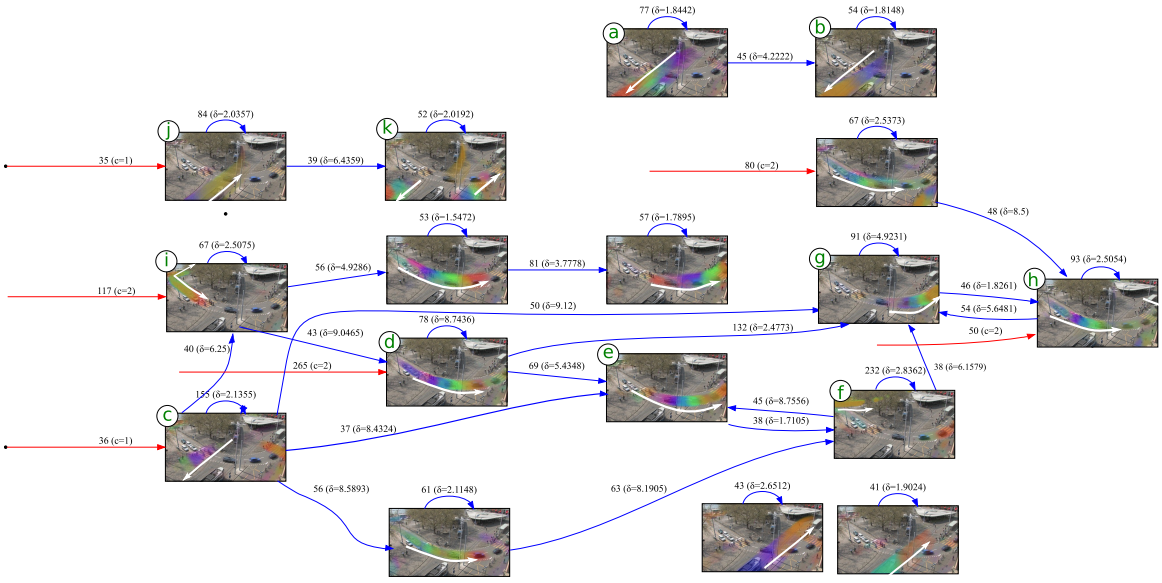


Figure 6.15. Event relationships from ETH-Z data with motifs of 5 seconds duration Low frequency edges are not shown (those with counts below 30). Blue edges: transitions with weight and lag δ. Red edges: independent activity with weight and causing state (here with only 1 state). 95% of the occurrences were dependent activities.

In equation (6.4), the prediction at time $t+1$ in addition to the motif starts in $t_s \in [t - T_z + 2, t - 1]$, is affected by motifs that start at t and $t+1$ and hence we have two prior terms now. The terms $P_t(z)$ and $P_{t+1}(z)$ refer to the proportion of event starts obtained from MER for time t and $t+1$. We call this prediction method MER+PLSM. We validate this approach by testing it on the Far-field data, by varying the number of PLSM motifs used to build the binary event matrix. In all the cases the maximum motif duration was 10 seconds, with a single state for the MER model as demonstrated in Figure 6.13. The baselines used for the comparison were: i) Temporal constancy method, ii) PLSM with a blind prior, and iii) Topic-HMM; all extended to predict observations at $t+1$ also. Parameters such as the sliding window size, the values for γ were same as in the experiments presented in section 5.5.4.

Figure 6.16 shows the Average Normalized predictive log-likelihood (ANL) computed from these models using a 10 fold cross validation procedure. The X-axis indicates the number of PLSM motifs with 10 second duration or the number of behavioral states in Topic-HMM. First, we know from Figure 5.23 and Figure 6.16 that PLSM performs well in predicting the first time-step. But the prediction accuracy drops down a bit for the second time step, in particular as the number of motifs is increased. This can be explained by the absence of any scene level knowledge on which activity triggers which others. The Topic-HMM does not do as well as PLSM for both first and second time-steps. But they consistently improve as the states are increased with a small reduction for the second time-step. The MER+PLSM model outperforms both PLSM with the blind prior and Topic-HMM at both the first and second time-steps. Recall that using the MER model with only one state and no dependent events boils down to the blind prior. Since the MER model uses the

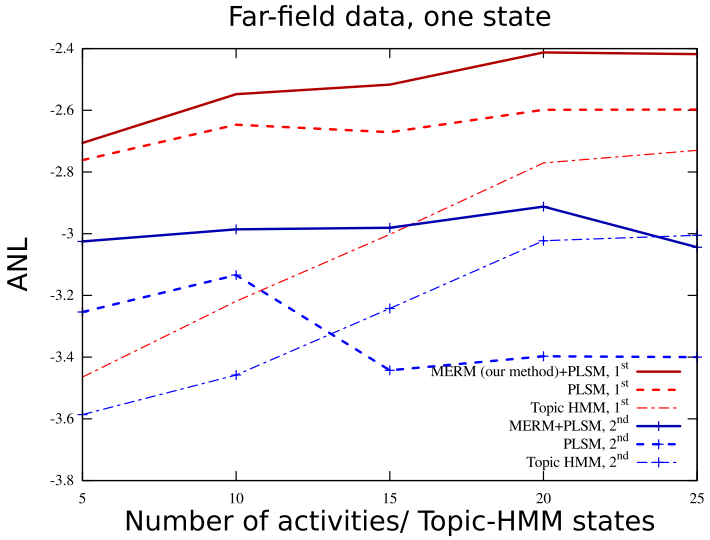


Figure 6.16. Prediction accuracy – Average Normalized predictive log-likelihood (ANL) for the first or the second future time steps.

inter-activity relations in addition to the regular PLSM method, it is reasonable to conclude that the activity dependencies learned from MER has improved the performance of PLSM prediction. Finally, for the temporal constancy method we obtained ANL values -3.5 for the first time-step and -4.7 for the second time step. The low measure for the second time step shows that the temporal constancy assumption can not generalize well for longer term predictions.

6.5 Conclusion

In this chapter, we presented a novel method called (MER) model that accepts a binary event matrix and discovers temporal relationships between event pairs as well as global rules dictating the scene. The main novelty comes from our observation that events need not be strictly dependent on the current state of the system but also on any event in the past. We proposed an efficient Gibbs sampling algorithm to infer the latent variables and the parameters. The hard task of attributing an activity to the past or to the global scene state is solved thanks to the joint sampling of the latent variables. We evaluated our method on a variety of scenes containing complex activity patterns. The results demonstrate our model's ability to capture both local rules (event-event relationships) and global rules (event-state relationships) automatically. This is in contrast with methods such as (Tran and Davis, 2008), where a combination of logical rules, representing domain knowledge, and probabilistic models were used to perform activity analysis.

However, there are some limitations of the MER model that needs discussion. Firstly, the MER model assumes that events are detected reliably and the data is given as a binary event matrix. It does not rely on strength or confidence of the event. Secondly, in our case we fix the prior mean and variance of the temporal lag parameter using *a priori* knowledge on the average motif durations.

While the model adapts the mean temporal lag $\delta_{z',z}(.)$ using the data, it cannot accommodate large variations e.g., 10 seconds lag for some event-pairs and 1 minute lag for some other event-pairs. One way to address this is by applying MER repeatedly with different values for the prior mean and variance.

Chapter 7

Conclusions and Future work

7.1 Conclusions

The subject of this thesis was to extract recurrent activity patterns from sensor logs by observing data over a long period of time. A wide context of applications motivated this choice: video content analysis for surveillance, stream selection and automatic summarization, video search and retrieval, human activity analysis using multi-modal sensors such as accelerometers, PIR, GPS and cell phone logs. To achieve this, we followed three paradigms established in Computer Vision and Machine Learning: 1) use of low-level features like back-ground subtraction and optical flow, so that they can be readily extracted without involving higher level scene semantics like object detection and tracking. 2) use of unsupervised methods, so that labeling large amount of data can be avoided. 3) use of a Probabilistic Topic Model formalism, due to its ability to succinctly capture latent themes in the data through co-occurrence analysis. The contributions made and the experiments done were chronologically presented in the chapters each adding a new dimension to the previous one. More precisely, after having presented the literature review in chapter 2 and summarized our datasets and features in chapter 3, chapter 4–6 described our original contributions that are summarized as follows:

In **Chapter 4**, we first investigated the use of PTMs for activity analysis. Using PLSA, with location, motion and size features, dominant patterns were successfully extracted. The resulting patterns were useful in understanding the common activities in the scene. Furthermore, it also provided scope to analyze the activities based on the participation of the visual features. Then, using the topic posterior distribution at each location as features, a novel activity scene segmentation algorithm was presented. The clustering algorithm showed meaningful results for any cluster size. Abnormality detection using PLSA was another task attempted. We proposed a novel abnormality detection measure based on document reconstruction error that compared well with existing measures. This exercise helped us to identify key areas to address, and provided avenues for further investigation.

In **Chapter 5** we moved from discovering simple topics with unordered words to more infor-

mative sequential patterns called motifs and introduced a novel probabilistic activity modeling approach called Probabilistic Latent Sequential Motifs (PLSM). The novelties were many fold. First, unlike methods similar to PLSA, where topics only model the co-occurrence of words within a time window, PLSM uses a temporal document given as word-time occurrences and discovers motifs, *i.e.,* topics with a temporal structure. Second, the model considers the important case where activities occur concurrently but not necessarily in synchrony in the temporal document. Third, the method explicitly models with latent variables, the starting time of the activities within the temporal document, enabling to implicitly align the occurrences of the same pattern during the joint inference of the temporal topics and their starting times. Fourth, the inference framework is enhanced with a sparsity constraint and a MAP estimation of the temporal variables of the motif. Elaborate experiments with structured as well as less structured data, spanning single and multiple views were used to demonstrate that the different outputs of the model have semantic significance; the motifs provide information about how and where an activity starts, how it proceeds and ends, the start time distributions along with the motifs can be used for event detection and future word prediction. For the challenging problem on selecting the appropriate number of motifs, we also proposed (in collaboration with Dr. Rémi Emonet) a new method called HDLSM that is based on the principles of Bayesian non-parametrics.

In **Chapter 6**, we considered motif detections or activiations called events to identify the underlying processes that govern their occurrences over time in complex surveillance scenes. To this end, we proposed a novel model called MER model, that accounts for two main factors: (i) the existence of global scene states that regulate which of the events can spontaneously occur; (ii) local rules that link past events to current ones with temporal lags. These complementary factors were mixed in the probabilistic generative process, thanks to the use of a binary random variable that selects for each event which one of the above two factors is applicable. The MER model uses a binary event matrix derived from the start time distributions learned from PLSM. All model parameters are efficiently inferred using a collapsed Gibbs sampling inference scheme. Experiments on various datasets from the literature showed that the model is able to capture temporal processes at multiple scales: scene-level first order Markovian process, and causal relationships amongst events, thus providing a rich interpretation of the scene's dynamical content.

7.2 Limitations and Future work

While we have shown many insights into activity modeling, the work does have some limitations. In this section, we highlight them and propose some ways of improving the model. At the end, we also present some potential new directions for research investigation.

Duplicate Motifs. In the PLSM model, while we used the BIC criteria to decide on the number of motifs, we still get duplicates apparently representing the same activity. Our understanding of this problems is that it could be due to over specified SLA vocabulary or motifs trying to capture minor variations in object speed.

We encountered a problem due to duplicate motifs while building the event detector in section 5.5.3. The event detector was built by associating a single motif to an activity. But occasionally, the competing duplicate motifs (that were not considered for the event detector) explain away an occurrence resulting in a missed detection. A naive solution to this problem is to apply a post processing step that merges motifs that are similar beyond a threshold. An alternative is to include a procedure in the inference process such that, after every few iterations, duplicate motifs are merged if it returns a better likelihood. Also using non-parametric methods like HDLSM can address this issue by selecting the appropriate motif size automatically.

Modeling speed variations. There is significant speed variations in the scene activities due to vehicle acceleration and perspective issues. While the PLSM model handles variations in local activity execution timing well, it can fail to handle large variations in the overall execution speed. There are several ways to handle this. First, we can conduct an a-posteriori analysis, by identifying motif replicas differing by speed execution variations. Second, we can apply PLSM on multiple temporal documents of the same video, each created with a different time resolution, for instance, 0.5 second to 5 seconds. Or we can introduce an explicit latent variable to model the execution speed. Although this can be added in a straightforward manner in the model, this would result in increased computational complexity.

Parametric forms for motifs. Currently, the motifs $P(w, t_r|z)$ are represented as probability tables. The parameters used to represent motifs therefore, increase as the the motif duration is increased. An interesting alternative is to model the temporal variable distribution $P(t_r|w, z)$ using parametric forms. For instance, the Beta distribution $\mathrm{Beta}(\alpha, \beta)$ covering the (normalized) duration of the motif is an attractive choice due to its flexibility in capturing a variety of shapes. But this limits $P(t_r|w, z)$ to have a single mode. Another alternative is a multi-modal distribution like the Gaussian mixture model. Changing to parametric forms will require some adaptations in the model. The mathematical tractability of the inference method needs to be considered while making the choice of the distribution.

Integrating MER and PLSM. Currently our activity analysis pipeline has several components: PLSA (SLA patterns), PLSM and finally the MER model, each of them performed independently. It would be better to create an end to end model that uses visual features and discovers words, motifs, scene states and event relationships. This can be achieved by building a generative process with multiple layers. The benefits apart from ease of use would be that each layer can propagate its learning to the other resulting in better overall learning. For instance, the scene states and event relations learned from the MER model at an iteration can constrain the $P(t_s, z|d)$ distribution at the next iteration resulting in better motifs, this in turn resulting in better estimates of MER parameters. But such a method would involve a fair amount of sophistication in terms of the generative modeling and its inference.

Issues with MER model. As discussed in section 6.5, the MER model currently relies on a binary event matrix. But it would be better if the detection strengths from $P(t_s, z|d)$ are taken into account.

This can be addressed by reformulating the MER generative process such that the observations are not just binary values but a value from the $[0, 1]$ interval. Correspondingly, the inference procedure needs to be updated so that instead of just co-occurrence counts, their strengths are also accommodated. Furthermore, performance of the MER model in the presence of spurious detections needs to be tested.

Online methods. As explained in chapter 1, one of the thesis motivations is to use activity analysis algorithms for automatic stream selection. While the strength of PLSA and PLSM on detecting unusual events were demonstrated in chapter 4 and (Emonet *et al.*, 2011b) respectively, deploying them on a metro station for automatic stream selection can pose additional challenges. Currently, PLSA and PLSM are learned in a batch mode and the model is not updated, whereas in a real-life monitoring situation there is continuous flow of data that may contain novel actions. So there is a need to design online versions of the proposed models that are capable of updating the model parameters as new data arrive. This requires answers to few important questions: a) what criteria should be used to judge novelty of the incoming data and b) how can the model be updated, which in our case translates to relearning the vocabulary, re-selecting the number of topics or motifs and adjusting their weights.

Action recognition. Unsupervised methods like PLSA have also been used to automatically classify or localize different actions in video sequences (Niebles *et al.*, 2008). These methods mostly relied on a bag-of-words approach, learned a model and associated a topic or cluster with each class. The classification and localization of actions is done by simply considering the action category with the maximum topic posterior distribution. In similar directions, we believe that some unique features of the PLSM method such as: a) the ability to capture temporal structures through motifs, b) ability to multiplex several activities in parallel and c) localize the motifs precisely in time, could prove advantageous in building a good action detector. However, the choice of visual features (*e.g.,* spatio-temporal features instead of optical flow), the ability to handle view changes and the discriminative ability of the motifs, are some aspects that need investigation.

Other modalities. Many sensor logs other than videos including PIRs in buildings, GPS and call logs in cell phones, accelerometers have been used along with unsupervised data-mining tools. MERL dataset (Wren *et al.*, 2007) and the Reality mining dataset (Eagle and Pentland, 2009) are some example candidates for this. But there are several challenges that need to be addressed before. Each of these sensors may be sampled with varying frequencies, each has varying timescales and different characteristics, and each has its own sources of noise For example, GPS data from cell phones suffers from noise due to the sensor as well as human behavior. GPS often does not work indoors, there might be problems in signal reception during travels, the cell phone might be switched off and the person might forget the phone or lend it to friends. Therefore, to study the performance of the proposed models on a different modality data, possibly with some model adaptations could be an interesting line of research.

Appendices

Appendix A

Parameter estimation for PLSM

In this section we detail the derivation of equations involved in PLSM inference. Our data $\mathcal{D} = n(w, t_a, d)$ is a matrix of counts, where each triplet (w, t_a, d) denotes the number of times a word w, appears at time t_a in document d. The probability of observing this data is given by:

$$P(\mathcal{D}) = \prod_{d=1}^{D} \prod_{t_a=1}^{T_d} \prod_{w=1}^{N_w} P(w, t_a, d)^{n(w,t_a,d)} \tag{A.1}$$

Taking the log on both sides we obtain our objective log-likelihood function as

$$\mathcal{L}(\mathcal{D}) = \log P(\mathcal{D}) = \sum_{d=1}^{D} \sum_{t_a=1}^{T_d} \sum_{w=1}^{N_w} n(w, t_a, d) \log P(w, t_a, d) \tag{A.2}$$

Since $P(w, t_a, d) = \sum_{t_s=1}^{T_{ds}} \sum_{z=1}^{N_z} P(w, t_a, d, z, t_s)$, the log-likelihood equation conditioned on the model parameters Θ *i.e.,* the probability distributions $P(z|d)$, $P(t_s|z, d)$, and $P(w, t_r|z)$ is written as

$$\mathcal{L}(\mathcal{D}|\Theta) = \sum_{d=1}^{D} \sum_{w=1}^{N_w} \sum_{t_a=1}^{T_d} n(w, t_a, d) \log \sum_{z=1}^{N_z} \sum_{t_s=1}^{T_{ds}} P(w, t_a, d, z, t_s) \tag{A.3}$$

We want to infer the model parameters with a constraint that the distribution on motif starting times $P(t_s|z, d)$ is sparse and peaky. To achieve this goal as said in section 5.2, we want to maximize $D_{KL}(U||P(t_s|z, d))$, the KL divergence between Uniform distribution and $P(t_s|z, d)$. So the constrained log-likelihood equation is given by:

Symbol	Description
$\mathcal{D}$	Dataset, count matrices of the form $n(w, t_s, d)$
z	Motif
w	Word (SLA patterns for real data)
d	Document index
t_a	Time index in document
t_s	Start time of a motif
t_r	Relative time from the start of the motif
Θ	Model parameters $\{P(z\|d), P(t_s\|z, d), P(w, t_r\|z)\}$
N_z	Number of motifs
N_w	Vocabulary size (number of different words)
D	Number of documents
T_z	Duration of a motif
T_d	Duration of the document
T_{ds}	Number of motif start time indices in a document
$\lambda_{z,d}$	Sparsity constraint weight
λ_{bic}	Penalty term weight in BIC equation

Table A.1. Notations used in the paper

$$\mathcal{L}_c(\mathcal{D}|\Theta) = \mathcal{L}(\mathcal{D}|\Theta) + \sum_{z,d} \lambda_{z,d} D_{KL}(U||P(t_s|z,d)) \tag{A.4}$$

$$= \mathcal{L}(\mathcal{D}|\Theta) + \sum_{z,d} \lambda_{z,d} \sum_{t_s=1}^{T_{ds}} \frac{1}{T_{ds}} \log \frac{1/T_{ds}}{P(t_s|z,d)} \tag{A.5}$$

$$= \mathcal{L}(\mathcal{D}|\Theta) + \sum_{z,d} \lambda_{z,d} \left(\sum_{t_s}^{T_{ds}} \frac{1}{T_{ds}} \log \frac{1}{T_{ds}} - \sum_{t_s=1}^{T_{ds}} \frac{1}{T_{ds}} \log P(t_s|z,d) \right) \tag{A.6}$$

By removing the constant factor we obtain equation (5.5) given by

$$\mathcal{L}_c(\mathcal{D}|\Theta) = \mathcal{L}(\mathcal{D}|\Theta) - \sum_{t_s,z,d} \frac{\lambda_{z,d}}{T_{ds}} \log P(t_s|z,d) \tag{A.7}$$

But the above equation cannot be solved directly due to summation terms inside the log making it intractable. Instead, the EM approach works by optimizing the expected log-likelihood of the complete data w.r.t to the hidden variables keeping the constraint unchanged, which gives

$$E[\mathcal{L}_c] = \sum_{d=1}^{D} \sum_{w=1}^{N_w} \sum_{t_a=1}^{T_d} \sum_{z=1}^{N_z} \sum_{t_s=1}^{T_{ds}} n(w, t_a, d) P(z, t_s|w, t_a, d) \log P(w, t_a, d, z, t_s) - \sum_{t_s,z,d} \frac{\lambda_{z,d}}{T_{ds}} \cdot \log(P(t_s|z,d)) \tag{A.8}$$

Notice that now, the log operates over the joint probability over all the variables and not just the observed variables. An optimized way to compute this is by using equation (5.2) and indices t_r and t_s instead of t_a. The joint distribution can be split into its constituent distributions using

equation (5.2) So, the expected log-likelihood equation is re-written as:

$$
\begin{aligned}
E[\mathcal{L}_c] &= \sum_{d=1}^{D}\sum_{w=1}^{N_w}\sum_{t_r=1}^{T_z}\sum_{z=1}^{N_z}\sum_{t_s=1}^{T_{ds}} n(w,t_a,d)P(z,t_s|w,t_s+t_r,d)\log[P(z|d)P(t_s|z,d) \\
&\quad P(t_r|z)P(w|t_r,z)] - \sum_{t_s,z,d}\frac{\lambda_{z,d}}{T_{ds}}\cdot\log P(t_s|z,d) \\
&= \sum_{d=1}^{D}\sum_{w=1}^{N_w}\sum_{t_r=1}^{T_z}\sum_{z=1}^{N_z}\sum_{t_s=1}^{T_{ds}} n(w,t_a,d)P(z,t_s|w,t_s+t_r,d)[\log P(z|d) \\
&\quad + \log P(t_s|z,d) + \log P(t_r|z) + \log P(w|t_r,z)] - \sum_{t_s,z,d}\frac{\lambda_{z,d}}{T_{ds}}\cdot\log P(t_s|z,d)
\end{aligned}
$$

The goal is thus to optimize this expression with constraints so that the distributions sum to one. Therefore such constraints are enforced using lagrangian multipliers. Finally, the constrained objective function that is optimized is given by:

$$
\begin{aligned}
\mathcal{H}(\Theta) &= E[\mathcal{L}_c] + \sum_z \gamma_z\left(1-\sum_{w,t_r}P(w,t_r|z)\right) + \sum_{z,d}\delta_{z,d}\left(1-\sum_{t_s}P(t_s|z,d)\right) \\
&\quad + \sum_d \tau_d\left(1-\sum_z P(z|d)\right) \qquad \text{(A.9)}
\end{aligned}
$$

Where γ_z, $\delta_{z,d}$ and τ_d are the lagrangian multipliers. The EM algorithm works by iterating through the following E-step and M-step:

E-step: In the Expectation step, the posterior distribution of hidden variables (z,d) is computed where the parameters come from the previous iteration's M-step,

$$
\begin{aligned}
P(z,t_s|w,t_a,d) &= \frac{P(d,z,t_s,t_a,w)}{P(w,t_a,d)} \\
&= \frac{P(z|d)P(t_s|z,d)P(t_r|z)P(w|t_r,z)}{\sum_{z',t'_s}P(z'|d)P(t'_s|z',d)P(t_a-t'_s|z')P(w|t_a-t'_s,z')} \qquad \text{(A.10)}
\end{aligned}
$$

M-step: In the Maximization step, maximizing $\mathcal{H}(\Theta)$ w.r.t to the parameters which are the probability mass functions results in the following set of equations.

$$
\begin{aligned}
&\sum_{t_s=1}^{T_{ds}}\sum_{t_r=0}^{T_z-1}\sum_{w=1}^{N_w} n(w,t_s+t_r,d)P(z,t_s|w,t_s+t_r,d) - \tau_d P(z|d) = 0, 1\le d\le D, \\
&\sum_{t_a=1}^{T_d}\sum_{w=1}^{N_w} n(w,t_s+t_r,d)P(z,t_s|w,t_s+t_r,d) - \frac{\lambda_{z,d}}{T_{ds}} - \delta_{z,d}P(t_s|z,d) = 0, 1\le d\le D, 1\le z\le N_z \\
&\sum_{d=1}^{D}\sum_{t_s=1}^{T_{ds}} n(w,t_s+t_r,d)P(z,t_s|w,t_s+t_r,d) - \gamma_z P(w,t_r|z) = 0, 1\le z\le N_z
\end{aligned}
$$

by eliminating the lagrangian multipliers [1], we obtain the following expressions that were presented in (equations (5.8–5.10))

$$P(z|d) \propto \sum_{t_s=1}^{T_{ds}} \sum_{t_r=0}^{T_z-1} \sum_{w=1}^{N_w} n(w, t_s + t_r, d) P(z, t_s|w, t_s + t_r, d) \tag{A.11}$$

$$P(t_s|z, d) \propto \max\left(\varepsilon, \left(\sum_{w=1}^{N_w} \sum_{t_r=0}^{T_z-1} n(w, t_s + t_r, d) P(z, t_s|w, t_s + t_r, d)\right) - \frac{\lambda_{z,d}}{T_{ds}}\right) \tag{A.12}$$

$$P(w, t_r|z) \propto \sum_{d=1}^{D} \sum_{t_s=1}^{T_{ds}} n(w, t_s + t_r, d) P(z, t_s|w, t_s + t_r, d) \tag{A.13}$$

1. also taking into account that probabilities need to be positive. As a technical detail, a minimum value ε is required for $p(t_s|z, d)$ to avoid undefined log-likelihoods in the objective function.

Appendix B

Hierarchical Dirichlet Latent Sequential Motifs

In this chapter, we present the Hierarchical Dirichlet Latent Sequential Motifs (HDLSM) model, which is a non-parametric improvisation of the PLSM model. This work was done in collaboration with Rémi Emonet and published in (Emonet *et al.*, 2011a). We also note that the notations used in this chapter differ from the notations previously used in chapter 5. Here, we confirm to the notation style that is widely followed in the Bayesian non-parametrics community.

The rest of this chapter is organized as follows: In section B.1, we will first present an overview of our approach and its goals. In section B.2, we give a background of the Dirichlet Processes (DP). followed by the explanation of the HDLSM model in section B.2.2. For details about the inference procedure and implementation, we refer to (Emonet *et al.*, 2011a).

B.1 Approach Overview

The input to our HDLSM model is a set of temporal documents (possibly a single long one) as presented in section 5.1 and show here in Figure B.1. This observed document is defined as a table of counts, where the entries reflect the amount of presence of a word from a fixed vocabulary at every instant of the temporal document. Our approach is depicted in Figure B.1 where each document is represented as a set of motif occurrences (*e.g.,* 7 of them in Figure B.1). Each occurrence is defined by a starting time instant and a motif. Motifs are shared by different occurrences within and across documents.

In our model, an important aspect is the use of Dirichlet Processes (DP). A DP is a non-parametric Bayesian process that represents an infinite mixture model. The term non-parametric refers to the fact that the model grows automatically to account for the observations. Dirichlet processes are often used to determine automatically the number of relevant elements in a mixture model (*e.g.,* number of topics or number of gaussians). A DP is an infinite mixture but observations

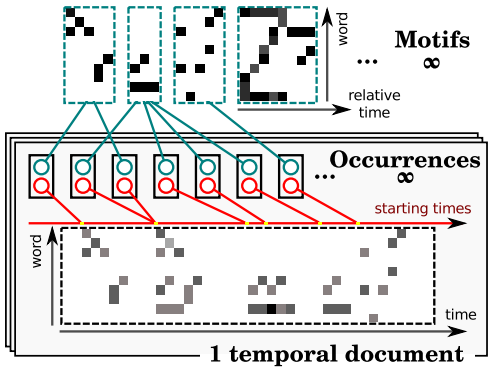


Figure B.1. Schematic generative model. A temporal document is made of word counts at each time instant. Each document is composed of a set of occurrences, each being defined as a motif type and a starting time. The motifs are shared by the occurrences within and across documents.

from a DP most probably tend to cluster on some limited elements of the mixture.

We use two levels of DP in our approach. At a lower level, within each document, we model the set of occurrences using a DP: the observations then cluster around an automatically determined number of occurrences. At a higher level, we model the set of motifs using a DP: the occurrences (and their associated observations) within and across documents then cluster around an automatically determined number of motifs. With this hierarchical approach, each observation is associated through its occurrence to a motif.

B.2 Proposed Model

Our model relies on Dirichlet Processes (DP) to discover motifs, their number, and find their occurrences. We will thus start by introducing DP before describing the core of our model in details.

B.2.1 Background on Dirichlet Processes (DP)

Here we introduce Dirichlet Processes, a method to naturally handle infinite mixture models and a building block of our proposed model. The mixture components we are using are categorical distributions [1] but all elements in the current subsection can be interpreted identically with any mixture model such as a Gaussian Mixture Model. We use $Comp$ to denote the component distribution in this introductory section.

Figure B.2a is a graphical representation of a finite mixture model with K components. The β vector is giving the weight of each mixture component and α is a prior (possibly uninformative) on these weights. Each Φ_k represents the parameters of a mixture component and for each observation x_i, z_i represents the index of the mixture component this observation is coming from.

Figure B.2b first shows that we can explicitly represent the mixture component selected by

1. sometimes "multinomial" is used in place of "categorical"

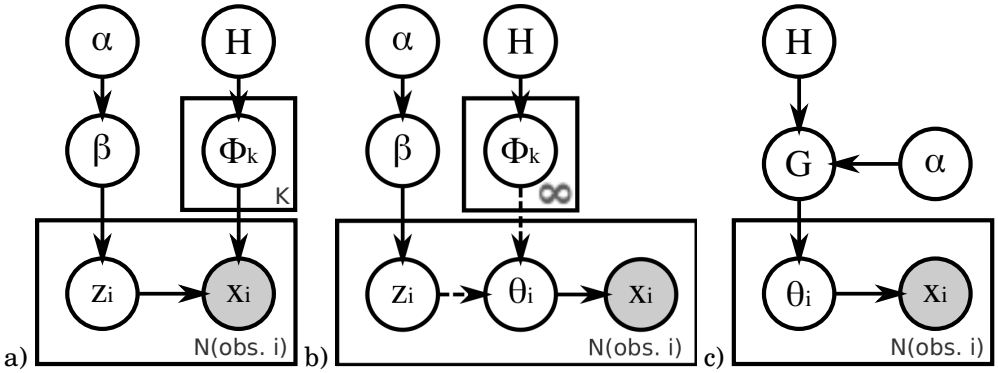


Figure B.2. Finite mixture and Dirichlet Process (infinite mixture): a) finite mixture with K elements; b) mixture representation for DP; c) compact representation for DP.

each observation noted θ_i. We use dashed arrows to indicate deterministic relations, here $\theta_i = \Phi_{z_i}$ (or, expressed as a draw from a Dirac distribution: $\theta_i \sim \delta_{\Phi_{z_i}}$). More importantly, Figure B.2b also illustrates the uniqueness of a Dirichlet Process, i.e., there are an infinite number of mixture components instead of a finite number K. To adapt to this infinite mixture elements, the weight vector β is of infinite length and the prior α takes a specific form. The α prior is now a single positive real value used as the parameter of a "GEM" (Griffiths, Engen, McCloskey) also known as a "stick breaking" process. This process produces an infinite list of weights that sum to 1: the first weight $\beta_1 = \beta'_1$ is drawn from a beta distribution $Beta(1, \alpha)$, the second weight is drawn in the same way but only from the remaining part, i.e. $\beta_2 = (1 - \beta_1) * \beta'_2$ with β'_2 drawn from $Beta(1, \alpha)$, and so on for the other weights, hence the "stick breaking" name. In addition to these weights, each mixture component parameter set Φ_k is drawn independently from a prior H, and each observation is drawn from its mixture component. We thus have the following:

$$\beta \sim GEM(\alpha) \tag{B.1}$$

$$\forall k \qquad \phi_k \sim H \tag{B.2}$$

$$\forall i \qquad z_i \sim \text{Categorical}(\beta) \tag{B.3}$$

$$x_i \sim Comp(\phi_{z_i}) \tag{B.4}$$

A more compact equivalent notation can be used to represent a Dirichlet Process. While the mixture representation is well adapted for deriving the Gibbs sampling scheme, the compact representation is widely used and might help us to get a quick overview of the model. In the compact representation from Figure B.2c, individual mixture components are not shown and instead their weighted countable infinite mixture $G = \sum_{k=1}^{\infty} \beta_k \delta_{\phi_k}$ is used. The corresponding representation,

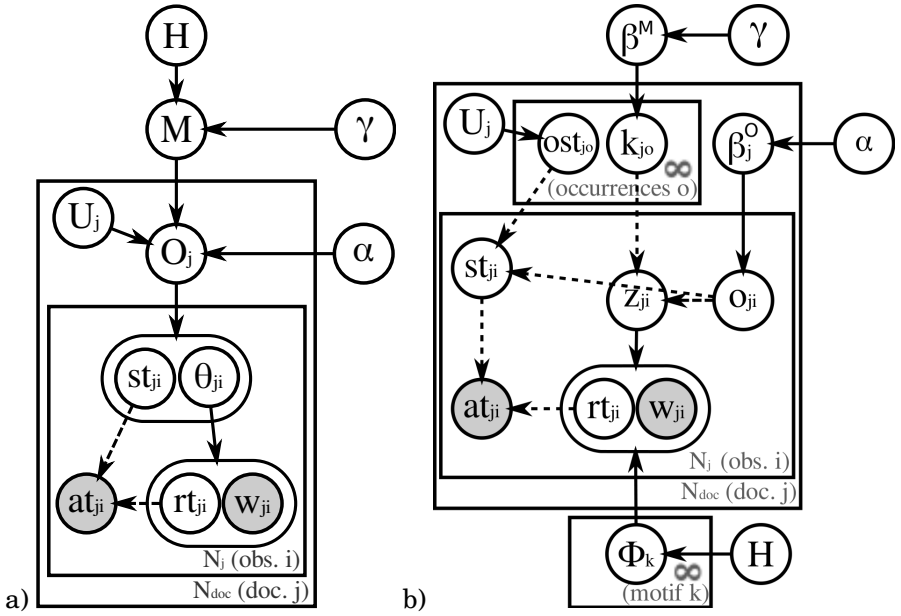


Figure B.3. The HDLSM model. a) with DP compact notation; b) with developed Dirichlet processes (using stick-breaking convention at both levels). Dashed arrows represents deterministic relations (conditional distributions are a Dirac).

using a DP notation, is given as:

$$G \sim DP(\alpha, H) \tag{B.5}$$

$$\forall i \qquad \theta_i \sim G \tag{B.6}$$

$$x_i \sim \theta_i \tag{B.7}$$

B.2.2 Base of the Proposed Model

Our goal is to automatically infer a set of motifs (temporal activity patterns) from a set of documents containing time-indexed words. More precisely, let us define a document j as a set of observations $\{(w_{ji}, at_{ji})\}_{i=1...N_j}$, where w_{ji} is a word belonging to a vocabulary V and at_{ji} is the absolute time instant at which the observation occurs within the document.

We also consider time information when defining our "motifs" as temporal probabilistic maps. More precisely, if ϕ_k denotes a motif table (i.e., the parameters of a categorical distribution), then $\phi_k(w, rt)$ denotes the probability that the word w occurs at a relative time instant rt after the start of the motif.

Our goal is to infer the set of motifs from one or more temporal documents. As discussed previously, this must be done altogether with inferring the occurrences (instants of occurrence) of all motifs in the documents. As it is difficult to fix the number of motifs before hand, we use a DP to allow the learning of a variable number of motifs from the data. Similarly, within each temporal document, we use another DP to model all motif occurrences as we don't know their number in advance.

Our generative model is thus defined using the graphical models presented in Figure B.3. Figure B.3a depicts our model using the compact Dirichlet process notation done for DP in Figure B.2c, whereas Figure B.3b depicts the developed notation (cf Figure B.2b). Notice that in these drawings, two variables in an elongated circle form a couple, indicating that they are generated together: the pair itself is drawn from a distribution over the pairs.

The equations associated with Figure B.3a are as follows:

$$M \sim DP(\gamma, H) \text{ where } H = Dir(\eta) \tag{B.8}$$

$$\forall j \qquad O_j \sim DP(\alpha, (U_j, M)) \tag{B.9}$$

$$\forall j\, \forall i \qquad (st_{ji}, \theta_{ji}) \sim O_j \tag{B.10}$$

$$(rt_{ji}, w_{ji}) \sim \text{Categorical}(\theta_{ji}) \tag{B.11}$$

$$at_{ji} = st_{ji} + rt_{ji} \tag{B.12}$$

where deterministic relations are denoted with "=". The first DP level generates our list of motifs in the form of an infinite mixture M. Each motif is drawn from H, defined as a Dirichlet distribution of parameter η.

Contrary to simpler mixture models such as LDA or HDP, our set of mixture components is not only shared across documents, but also across motif occurrences using the DP at the second level. More precisely, the document specific distribution O_j is not defined as a mixture over motifs, but as an infinite mixture over occurrences from "start-time $\times$ motif" (cf Figure B.1), since the base distribution is defined by (U_j, M). Each of the atoms is thus a couple (ost_k, ϕ_k), where $ost_k \sim U_k$ is the occurrence starting time drawn from U_j, a uniform distribution over the set of possible motif starting times in the document j, and $\phi_k \sim M$ is one of the motifs drawn from the mixture of motifs.

Observations (w_{ji}, at_{ji}) are then generated by repeatedly sampling a motif occurrence (equation (B.10)), using the obtained motif θ_{ji} to sample the word w_{ji} and its relative time in the motif rt_{ji} (equation (B.11)). From the relative time rt_{ji}, using the sampled starting time st_{ji}, the word absolute time occurrence at_{ji} can be deduced (equation (B.12)).

The fully developed model given in Figure B.3b helps to understand the generation process and the inference better. The corresponding equivalent equations can be written as:

$$\beta^M \sim GEM(\gamma) \tag{B.13}$$

$$\forall k \qquad \phi_k \sim H \tag{B.14}$$

$$\forall j \qquad \beta^o_j \sim GEM(\alpha) \tag{B.15}$$

$$\forall j \forall o \qquad ost_{jo} \sim U_j \quad \text{and} \quad k_{jo} \sim \beta^M \tag{B.16}$$

$$\forall j \forall i \qquad o_{ji} \sim \beta^o_j \tag{B.17}$$

$$z_{ji} = k_{jo_{ji}} \quad \text{and} \quad st_{ji} = ost_{jo_{ji}} \tag{B.18}$$

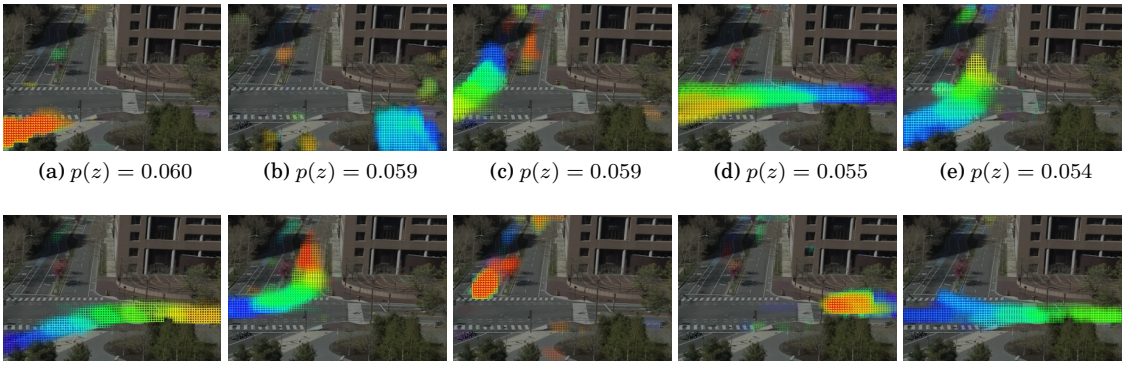
$$(rt_{ji}, w_{ji}) \sim \text{Categorical}(\phi_{z_{ji}}) \tag{B.19}$$

$$at_{ji} = st_{ji} + rt_{ji} \tag{B.20}$$

The main difference with the compact model is that the way motif occurrences are generated is explicitly represented. Occurrences are the analog of the "tables" in the Chinese Restaurant Process analogy of the HDP model: both the global GEM distribution over motifs β^M and U_j are used to associate motif indices k_{jo} and starting times ost_{jo} to each occurrence (Eq. B.16), while the document specific GEM β^o_j is used to sample the occurrence associated with each word (equation (B.17)), from which generating the observations can be done as presented above (equation (B.18–B.20)).

B.3 PLSM vs HDLSM

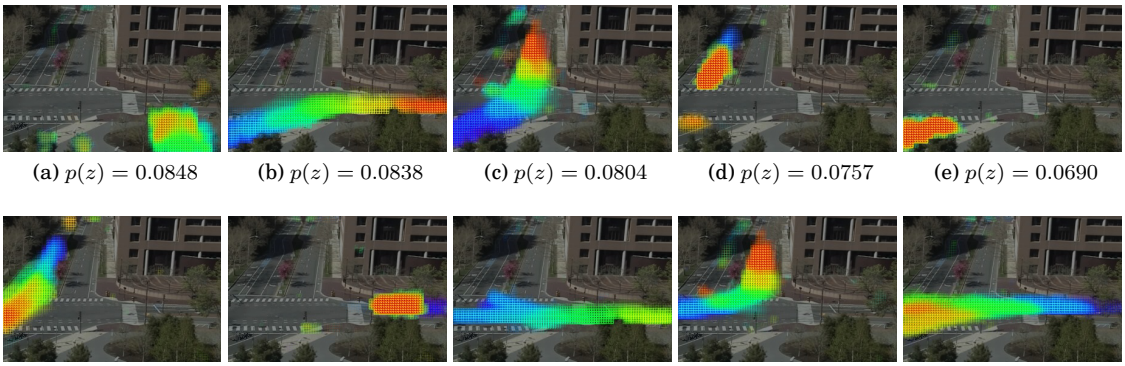
In this section we briefly present a qualitative comparison of the PLSM and HDLSM method interms of the motifs obtained and their related aspects.



(a) $p(z) = 0.060$ (b) $p(z) = 0.059$ (c) $p(z) = 0.059$ (d) $p(z) = 0.055$ (e) $p(z) = 0.054$

(f) $p(z) = 0.052$ (g) $p(z) = 0.049$ (h) $p(z) = 0.048$ (i) $p(z) = 0.043$ (j) $p(z) = 0.041$

Figure B.4. Top ten motifs out of 26 from MIT data discovered using PLSM method. The motifs are 10 seconds long. The BIC criteria was used to select the number of motifs.



(a) $p(z) = 0.0848$ (b) $p(z) = 0.0838$ (c) $p(z) = 0.0804$ (d) $p(z) = 0.0757$ (e) $p(z) = 0.0690$

(f) $p(z) = 0.0648$ (g) $p(z) = 0.0627$ (h) $p(z) = 0.0620$ (i) $p(z) = 0.0536$ (j) $p(z) = 0.0524$

Figure B.5. Top ten motifs out of 24 motifs from MIT data discovered using HDLSM method. The motifs are 10 seconds long.

In Figure B.4, we show 10 top ranked motifs out of 26, extracted from MIT data using the PLSM method. The BIC criteria was used to select the number of motifs. In Figure B.5, we show 10 top

ranked motifs out of 24, extracted from MIT data using HDLSM method. The number of motifs is automatically decided by HDLSM. From this experiment, we first note that we arrive at the same ball park figure for the number of motifs using both BIC criteria and HDLSM. When comparing the top ten motifs from both the methods, we see that they capture very similar activities, with some differences in the weights assigned to the motifs.

Appendix C

Parameter estimation for MER model

As is the case of many hierarchical Bayesian models like Blei *et al.* (2003a); Heinrich (2004), exact inference for our model is intractable. But since the model consists of conjugate pairs like Gamma-Poisson, Dirichlet-Multinomial and Beta-Bernoulli and Normal-Normal, it is possible to derive a collapsed Gibbs sampling algorithm by integrating out the parameters $\{\pi, \lambda, \epsilon, \theta, \tau, \delta\}$. The algorithm proceeds by iteratively sampling the decision variable s_i^t, indicator variable v_i^t for each observation O_i^t conditioned on all other variables, parameters and hyper-parameters. The state indicators c^t are sampled for each time instant conditioning on rest of the variables.

Since each occurrence also gives its occurrence time implicitly, we will drop the t associated with s_i^t, v_i^t, O_i^t and simplify them by using s_i, v_i and o_i instead except in places where time needs to be mentioned explicitly. We will also use capital letters O, S, V to refer to the set of occurrences, their corresponding selector variables, and indicator variables. $O_{\neg i}, S_{\neg i}, V_{\neg i}$, will indicate all the occurrences, selector variables and the indicator variables except the i^{th} one. $C_{\neg i}$ is used to indicate the state variables at all time instants except at the current time *i.e.,* c^t. The set of hyper-parameters set $\{\varphi, \gamma, \xi, \alpha, \beta, \mu_0, \sigma_0^2\}$ is simply referred as hp.

Selector and Indicator variables.

We would like to sample the selector variables s_i and v_i together for each observation o_i. We need to sample this according to the probabilities of four different conditions.

– Case 1: $p(s_i = 1, v_i = -1)$: When $s_i = 1$, v_i takes a dummy value of -1. This probability depends on two factors: i) Probability of seeing a $s_i = 1$ at the current state and ii) Probability of the event corresponding to the current observation to be generated by the current state

directly.

$$p(s_i = 1, v_i = -1|S_{\neg i}, V, O, C, hp..) \propto p(s_i = 1, v_i = -1, S_{\neg i}, V_{\neg i}, O, C|hp)$$

$$\propto p(v_i = -1|s_i = 1)p(o_i = z|s_i = 1, c_i = k, S_{\neg i}, V_{\neg i}, O_{\neg i}, hp)\cdot$$

$$p(s_i = 1|S_{\neg i}, c^t = k, C_{\neg i}, hp) \tag{C.1}$$

$$\propto p(o_i = z|c_i = k, s_i = 1, S_{\neg i}, V_{\neg i}, O_{\neg i}, hp)\cdot$$

$$p(s_i = 1|c^t = k, S_{\neg i}, C_{\neg i}, hp) \tag{C.2}$$

$$\propto \int p(o_i = z|s_i = 1, c_i = k, \theta_k)p(\theta_k|S_{\neg i}, O_{\neg i}, C_{\neg i})\mathbf{d}\theta_k\cdot$$

$$\int p(s_i = 1|c_i = k, \epsilon_k)p(\epsilon_k|S_{\neg i}, C_{\neg i})\mathbf{d}\epsilon_k \tag{C.3}$$

Expression C.1 is obtained by splitting the LHS of the equation into $p(o_i = z|s_i = 1, c_i = k, S_{\neg i}, V_{\neg i}, O_{\neg i}, hp)$ which is the likelihood part and $p(s_i = 1|S_{\neg i}, c^t = k, C_{\neg i}, hp)$ which is the prior part and by noting that
$p(v_i = -1|s_i = 1) = 1$.
Finally the two parts of expression C.3 are Multinomial-Dirichlet, and Bernoulli-Beta pairs respectively. They are conjugate pairs and therefore the parameters can be integrated out in closed form. Evaluating the integral leads to calculating the expectation of the Dirichlet and Beta distributions which is given as:

$$p(s_i = 1, v_i = -1|S_{\neg i}, V, O, C, hp..) \propto \frac{q^{(z)}_{-i,k} + \alpha}{q^{(.)}_{-i,k} + N_z\alpha} \cdot \frac{l^{(1)}_{-i,k} + \xi_1}{l^{(.)}_{-i,k} + \xi_0 + \xi_1} \tag{C.4}$$

In the above expression $q^z_{-i,k}$ is a count of number of times motif z is observed with state k when $s^t_i = 1$ removing the current observation. Similarly, and $l^1_{-i,k}$ is the count of $s_i = 1$ appearing with state k barring the current observation.

- Case 2: $p(s_i = 1, v_i = j)$: By our definition, this case is impossible and therefore $p(s_i = 1, v_i = j) = 0$
- Case 3: $p(s_i = 0, v_i = -1)$: Again by the definition of v^t_i, it has no meaning when $s_i = 0$. Therefore, this probability also equals zero.
- Case 4: $\forall j \in P^t, p(s_i = 0, v_i = j)$:

$$p(s_i = 0, v_i = j | c_i = k, O, S_{\neg i}, V_{\neg i}, C_{\neg i}, hp) \propto p(s_i = 0, v_i = j, c_i = k, S_{\neg i}, V_{\neg i}, O, C_{\neg i} | hp) \tag{C.5}$$

$$\propto p(v_i = j | s_i = 0, c_i = k, S_{\neg i}, V_{\neg i}, P^t, C_{\neg i}, hp)$$
$$p(s_i = 0 | o_i = z, c_i = k, S_{\neg i}, V_{\neg i}, O_{\neg i}, C_{\neg i}, hp) \tag{C.6}$$
$$\propto p(v_i = j | s_i = 0, P^t) p(s_i = 0 | c_i = k, S_{\neg i}, C_{\neg i}, \xi)$$
$$p(o_i = z | s_i = 0, P^t(j) = z', S_{\neg i}, O_{\neg i}, V_{\neg i}, \beta, \mu_0, \sigma_0^2) \tag{C.7}$$
$$\propto \frac{1}{N_p^t} \cdot p(s_i = 0 | c_i = k, S_{\neg i}, C_{\neg i}, \xi)$$
$$p(o_i = z | s_i = 0, P^t(j) = z', S_{\neg i}, O_{\neg i}, V_{\neg i}, \beta, \mu_0, \sigma_0^2) \tag{C.8}$$
$$\propto \frac{1}{N_p^t} \int p(s_i = 0 | \epsilon_k, c_i = k) p(\epsilon_k | S_{\neg i}, C_{\neg i}) \mathbf{d}\epsilon_k \cdot$$
$$\int p(o_i = z | s_i = 0, P^t(j) = z', \tau_{z'}) p(\tau_{z'} | S_{\neg i}, V_{\neg i}) \mathbf{d}\tau_{z'} \cdot$$
$$\int p(o_i = z | s_i = 0, P^t(j) = z', \delta_{z'}(z)) \cdot p(\delta_{z'}(z) | S_{\neg i}, V_{\neg i}) \mathbf{d}\delta_{z'}(z) \tag{C.9}$$

We move from expression C.5 to expression C.7 by splitting it into likelihood prior terms and simplifying using conditional independence of the variables, *i.e.,* v_i is conditionally independent of all other variables given s_i and P^t. Also the $\frac{1}{N_p^t}$ factor in expression C.8 comes from our definition of v_i in the generative process.

There are three integrals to the right hand side of the above equation. The first integral is similar to the second part of expression C.4, but instead deals with counts of $s_i = 0$ in state k. The second integral is a Dirichlet-Multinomial distribution arising from the transition matrix. The third integral is a Gaussian-Gaussian conjugate coming from the temporal lag. Thanks to the conjugacy property, we can evaluate all the three integrals in closed form and get the following proportionality term:

$$p(s_i = 0, v_i = j | O, S_{\neg i}, V_{\neg i}, c_i = k, C_{\neg i}, hp) \propto \frac{1}{N_p^t} \cdot \frac{l_{-i,k}^{(0)} + \xi_0}{l_{-i,k}^{(.)} + \xi_0 + \xi_1} \cdot \frac{r_{-i,z'}^{(z)} + \beta}{r_{-i,z'}^{(.)} + N_z \cdot \beta} \cdot \mathcal{N}\left(t - t' | \delta_{-i,z'}(z), \sigma_{z',z}^2 + \sigma^2\right) \tag{C.10}$$

In the above equation, $l_{-i,k}^{(0)}$ indicates the number of times $s_i = 0$ occurs with state k. $r_{-i,z'}^{(z)}$ is the number of times motif z appears after z'. $r_{-i,z'}^{(.)}$ is the total count of any event appearing after z'. $t - t'$ is the temporal lag between the occurrences $P^t(v_i) = z'$ and $o_i = z$. $\delta_{-i,z'}(z)$ and σ_n^2 are the posterior mean and variance of the temporal lag calculated from all associations of type z' and z. All the above calculations exclude the current observation. The posterior mean and variance are given by

$$\delta_{z'}(z) = \left(\frac{\mu_0}{\sigma_0^2} + \frac{D(z',z)}{\sigma^2}\right) \cdot \sigma_{z',z}^2 \tag{C.11}$$

$$\sigma_{z',z}^2 = \left(\frac{1}{\sigma_0^2} + \frac{r_{-i,z'}^{(z)}}{\sigma^2}\right)^{-1} \tag{C.12}$$

μ_0 and σ_0^2 are the prior mean and standard deviation of the lag parameter. The variance of lags between motif pairs is fixed at σ^2. $D(z',z)$ is the sum of all lag due to associations of type z' and z. For a more detailed derivation of this expression we refer to Murphy (2007).

State variables.

Here we derive the expression for getting the current state variable c^t. This is done by looking at all the connections coming in and out of the node c^t in the graph.

$$p(c^t = k|C_{\neg i}, S, V, O, N_o, hp) \propto p(c^t = k, C_{\neg i}, S_{\neg i}, V, O, N_o, hp) \tag{C.13}$$

$$\begin{aligned}
&\propto \int_{\lambda_k} p(N_o^t|\lambda_k, c^t = k)p(\lambda_k|N_o^{-t}, \gamma_1, \gamma_2)\mathbf{d}\lambda_k \cdot \\
&\prod_{i=1}^{N_o^t} \int_{\epsilon_k} p(s_i^t|\epsilon_k, c^t = k)p(\epsilon_k|C^{-t}, S^{-t}, \xi)\mathbf{d}\epsilon_k \cdot \\
&\prod_{i \in \{j:s_j^t=1\}} \int_{\theta_k} p(o_i|s_i^t = 1, \epsilon_k, c^t = k)p(\theta_k|C^{-t}, S^{-t}, O^{-t}, \alpha)\mathbf{d}\theta_k \\
&\int_{\pi} p(c^t = k, c^{t-1}, c^{t+1}|\pi)p(\pi|C^{-\{t-1,t,t+1\}}, S^{-t}, N_o^{-t})\mathbf{d}\pi
\end{aligned} \tag{C.14}$$

Here $C^{-t}, S^{-t}, O^{-t}N_o^{-t}$ refer to all the states, selector variables, and occurrences and number of occurrences except at time t. Based on the edges coming in and out of c^t we get four parts in the above equation relating to: 1) number of events, 2) number of independent and dependent events generated, 3) the motifs coming from the current state and, 4) the probability of arriving at c^t from c^{t-1} and reaching c^{t+1} from c^t. We will deal with each of them individually.

Occurrence Count.

$$\int_{\lambda_k} p(N_o^t|\lambda_k, c^t = k)p(\lambda_k|N_o^{-t}, \gamma_1, \gamma_2)\mathbf{d}\lambda_k = \int_{\lambda_k} \frac{\lambda_k^{N_o^t} \cdot e^{-\lambda_k}}{N_o^t!} \cdot \frac{\gamma_1^{\gamma_2}\lambda_k^{\sum_{i \in \{c^j=k, j \neq t\}} N_o^i + \gamma_1 - 1} \cdot e^{-\lambda_k(\gamma_2+u)}}{\Gamma(\gamma_1)}\mathbf{d}\lambda_k \tag{C.15}$$

where γ_1 and γ_2 are the parameters of Gamma distribution and $u = \sum_i I(c^i = k, i \neq t)$. Making use of the Gamma-Poisson conjugacy we can write the above equation as:

$$\int_{\lambda} p(N_o^t|\lambda_k, c^t = k)p(\lambda_k|N_o^{-t}, \gamma_1, \gamma_2)\mathbf{d}\lambda_k = \int_{\lambda_k} \frac{\lambda_k^{\sum_{i \in \{c^j=k, j \neq t\}} N_o^i + N_o^t + \gamma_1 - 1} e^{-\lambda_k(\gamma_2+u+1)}\gamma_1^{\gamma_2}}{N_o^t! \cdot \Gamma(\gamma_1)}\mathbf{d}\lambda_k \tag{C.16}$$

by noting that for the Gamma distribution

$$\int_{\lambda} \text{Gamma}(\lambda|\gamma_1,\gamma_2)\mathbf{d}\lambda = \int_{\lambda} \frac{e^{-\gamma_2\lambda}\lambda^{\gamma_1-1}\gamma_2^{\gamma_1}}{\Gamma(\gamma_1)}\mathbf{d}\lambda = 1 \tag{C.17}$$

and by multiplying and dividing the equation by the appropriate terms and removing constant terms, we get,

$$\int_{\lambda_k} p(N_o^t|\lambda_k, c^t = k)p(\lambda_k|N_o^{-t},\gamma_1,\gamma_2)\mathbf{d}\lambda_k \propto \frac{\Gamma(\omega_1)}{N_o^t!\cdot\omega_2^{\omega_1}} \tag{C.18}$$

where, $\omega_1 = \sum_{i\in\{c^j=k, j\neq t\}} N_o^i + N_o^t + \gamma_1, \omega_2 = \gamma_2 + u + 1$

Motifs and Selector variables.

$$\begin{aligned}\prod_{i\in\{j:s_i^t=1\}} \int_{\epsilon} p(s_i^t = 1|\epsilon_k, c^t = k)p(\epsilon_k|C^{-t}, S^{-t},\xi)\mathbf{d}\epsilon_k &= \int_{\epsilon_k} \epsilon_k^{l_t^1}(1-\epsilon_k)^{l_t^0}\text{Dir}(l_{-t,k}+\xi)\mathbf{d}\epsilon_k \\ &= \frac{\Delta(l_t + l_{-t,k} + \xi)}{\Delta(l_{-t,k}+\xi)} \qquad \text{(C.19)} \\ &= \frac{\Gamma(l_t^1 + l_{-t,k}^1 + \xi_1)\Gamma(l_t^0 + l_{-t,k}^0 + \xi_0)}{\Gamma(l_t^1 + l_{-t,k}^1 + \xi_1 + l_t^0 + l_{-t,k}^0 + \xi_0)} \cdot \frac{\Gamma(l_{-t,k}^0 + l_{-t,k}^1 + \xi_0 + \xi_1)}{\Gamma(l_{-t,k}^0 + \xi_0)\Gamma(l_{-t,k}^1 + \xi_1)} \qquad \text{(C.20)}\end{aligned}$$

The above distribution is a Beta-Bernoulli distribution, with the Beta hyper-parameters $\xi = \{\xi_0, \xi_1\}$. Due to their conjugacy, we obtain the expression directly in terms of Gamma functions. where,

$$\Delta(\vec{\alpha}) = \frac{\prod_{i=1}^K \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^K \alpha_i)} \tag{C.21}$$

In the above expression $l_t = \{l_t^0, l_t^1\}$ stands for the number of times, $s_i^t = 0/1$ at the current instant t. $l_{-t,k} = \{l_{-t,k}^0, l_{-t,k}^1\}$ is the number of times $s_i^t = 0/1$ from all other time instants when the state is at k except the current one.

$$\begin{aligned}\prod_{i\in\{j:s_j^t=1\}} \int_{\theta} p(o_i|s_i^t = 1, \epsilon_k, c^t = k)p(\theta_k|C^{-t}, S^{-t}, O^{-t},\alpha)\mathbf{d}\theta_k &= \int_{\theta_k} \prod_{z=1}^{N_z} \theta_{k,z}^{q_t^z}\text{Dir}(q_{-t,k}^z + \alpha)\mathbf{d}\theta_k \\ &= \frac{\Delta(q_{t,k} + q_{-t,k} + \alpha)}{\Delta(q_{-t,k}+\alpha)} \qquad \text{(C.22)}\end{aligned}$$

Here again we have a Dirichlet-Multinomial combination. The multinomial parameter is integrated out and the resulting expresion is in terms of Gamma functions. The count variables $q_t = \{q_t^z\}_{z=1}^{N_z}$ is the vector containing the number of times a motif z is observed at the current instant (it is either 1 or 0 in our case), and $q_{-t,k} = \{q_{-t,k}^z\}_{z=1}^{N_z}$ gives the the number of times each motif z is observed with k at all other instants except the current time t.

State transitions.

$$\int_{\pi} p(c^t = k, c^{t-1}, c^{t+1}|\pi)p(\pi|C^{-\{t-1,t,t+1\}}, \varphi)\mathbf{d}\pi \propto \frac{n^{c^t}_{-i,c^{t-1}} + \varphi}{n^{(\cdot)}_{-i,c^{t-1}} + N_c\varphi} \cdot \frac{n^{c^{t+1}}_{-i,c^t} + I(c^{t-1} = c^t = c^{t+1}) + \varphi}{n^{(\cdot)}_{-i,c^t} + I(c^{t-1} = c^t) + N_c\varphi} \tag{C.23}$$

The state transition is defined by the counts $n^{c^t}_{-i,c^{t-1}}$, $n^{c^{t+1}}_{-i,c^t}$, which are number of times a transition occurs from state c^{t-1} to state c^t, and from c^t to c^{t+1} after removing the current link. I is an Identity function that takes value 1, when the argument is true and 0 otherwise. $n^{(\cdot)}_{-i,c^t}$ is the the count of transitions from c^t to all other states barring the current transition. Please refer Griffiths *et al.* (2004) for this derivation.

Putting all the four parts together we get:

$$p(c^t = k|C_{\neg i}, S, V, O, N_o, hp) \propto \frac{\Gamma(\omega_1)}{N_o^t! \cdot \omega_2^{\omega_1}} \cdot \frac{\Gamma(l^1_t + l^1_{-t,k} + \xi_1)\Gamma(l^0_t + l^0_{-t,k} + \xi_0)}{\Gamma(l^1_t + l^1_{-t,k} + \xi_1 + l^0_t + l^0_{-t,k} + \xi_0)} \cdot \frac{\Gamma(l^0_{-t,k} + l^1_{-t,k} + \xi_0 + \xi_1)}{\Gamma(l^0_{-t,k} + \xi_0)\Gamma(l^1_{-t,k} + \xi_1)} \cdot \frac{\Delta(q_{t,k} + q_{-t,k} + \alpha)}{\Delta(q_{-t,k} + \alpha)} \cdot \frac{n^{c^t}_{-i,c^{t-1}} + \varphi}{n^{(\cdot)}_{-i,c^{t-1}} + N_c\varphi} \cdot \frac{n^{c^{t+1}}_{-i,c^t} + I(c^{t-1} = c^t = c^{t+1}) + \varphi}{n^{(\cdot)}_{-i,c^t} + I(c^{t-1} = c^t) + N_c\varphi} \tag{C.24}$$

It is important to note that, in practice calculating Delta functions $\Delta(\cdot)$ or Gamma functions $\Gamma(\cdot)$ result in overflows. This was the case while sampling the state probabilities. So we made use of log probabilities instead to calculate the state transitions. Calculations in Log probabilities are both efficient and safe.

Appendix D

Bayesian Statistics

Here are some basic concepts involving conjugacy of distributions that are typically used in this thesis and more generally in Bayesian parameters estimation. The core of this analysis is the use of Bayes rule which tells us how to combine the prior probability $p(\theta)$ and the likelihood $p(\mathcal{D}|\theta)$ to get the posterior $p(\theta|\mathcal{D})$:

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})}, \tag{D.1}$$

where the denominator is

$$p(\mathcal{D}) = \int p(\mathcal{D}|\theta)p(\theta)d\theta \tag{D.2}$$

This equation (D.2) is often computationally difficult to compute. However when the prior and likelihood terms have certain nice mathematical forms, we can work out the integration easily. This computation is also used to calculate the predictive probablity of some new data given the parameters computed from existing data. More specifically, we say a prior distribution is conjugate to a likelihood if, when multiplied together, the posterior has the same mathematical form as the prior distribution. Here are some prior-likelihood conjugate pairs.

Dirichlet-Multinomial model.

Consider a K dimensional multinomial random variable $\vec{\theta}$, where $0 \leq \theta_i \leq 1, \forall i$ and $\sum_i \theta_i = 1$. Rolling a 6 faced dice is a typical example of draws from a multinomial distribution where $K = 6$ *i.e.,* there are one of six possibilities at each draw. Let $\mathcal{D}$ be the counts of the K events in a set of N trials. Then the likelihood of this data is simply:

$$L(\vec{\theta}|\mathcal{D}) = p(\vec{\theta}|C) = \prod_{i=1}^{K} \theta_{i=1}^{n_i} \tag{D.3}$$

where, n_i is the number of times the i^{th} event is observed. Let us consider that we have some prior information about this experiment and roughly know the prior counts of each of the K events

i.e., a vector of counts $\vec{\alpha}$. Then the multinomial parameter $\vec{\theta}$ can be thought as a draw from this prior counts given by the Dirichlet distribution.

$$p(\vec{\theta}|\vec{\alpha}) = \frac{1}{\Delta(\vec{\alpha})} \prod_{i=1}^{k} \theta_i^{\alpha_i - 1}, \tag{D.4}$$

where $\Delta(\vec{\alpha}) = \frac{\prod_{i=1}^{k} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{k} \alpha_i)}$ and Γ denotes the Gamma distribution. Note that $\frac{\vec{\alpha}}{\|\vec{\alpha}\|_1}$ represents the expected values of the parameter θ (where $\|\vec{\alpha}\|_1$ is the L1 norm of $\vec{\alpha}$), and, when the Dirichlet is used as a prior over the parameters θ of a multinomial distribution, $\|\vec{\alpha}\|$ denotes the strength of the prior, and can be viewed as a count of virtual observations distributed according to $\frac{\vec{\alpha}}{\|\vec{\alpha}\|_1}$.
Property 1:

$$\int_{\Theta} p(\theta|\vec{\alpha}) d\theta = \int_{\Theta} \frac{1}{\Delta(\vec{\alpha})} \prod_{i=1}^{k} \theta_i^{\alpha_i - 1} d\theta = 1 \tag{D.5}$$

therefore we get

$$\Delta(\vec{\alpha}) = \int_{\Theta} \prod_{i=1}^{k} \theta_i^{\alpha_i - 1} d\theta \tag{D.6}$$

Property 2: Now given the data $\mathcal{D}$ and the prior counts $\vec{\alpha}$, we can obtain a posterior estimate of the multinomial parameters, using the above properties.

$$p(\vec{\theta}|C, \vec{\alpha}) = \frac{\prod_{n=1}^{N} p(c_n|\vec{\theta}) p(\vec{\theta}|\vec{\alpha})}{\int_{\Theta} \prod_{n=1}^{N} p(c_n|\vec{\theta}) p(\vec{\theta}|\vec{\alpha}) d\vec{\theta}} \tag{D.7}$$

$$= \frac{\prod_{i=1}^{K} \theta_i^{\alpha_i + n_i - 1}}{\Delta(\vec{\alpha} + \vec{n})} \tag{D.8}$$

$$= Dir(\vec{\theta}|\vec{\alpha} + \vec{n}), \text{where, } \vec{n} = \{n_i\}_{i=1}^{K} \tag{D.9}$$

Finally what we get is in the form of a Dirichlet distribution $Dir(\vec{\alpha} + \vec{n})$. This is simply the effect of Dirichlet-Multinomial Conjugacy. But called in several names as Polya urn scheme or sampling with over-replacement.
Property 3: Modeling a new dataset $\mathcal{D}_{new}$, we can calculate the probability of this data given our multinomial and Dirichlet parameters. This is done by integrating out the multinomial parameters as follows:

$$p(\mathcal{D}_{new}|\vec{\alpha}) = \int_{\Theta} p(C_{new}|\vec{\theta}) p(\vec{\theta}|\vec{\alpha}) d\vec{\theta} \tag{D.10}$$

$$= \int \prod_{i=1}^{K} \theta_i^{n_i} \frac{1}{\Delta(\vec{\alpha})} \prod_{i=1}^{K} \theta_i^{\alpha_i - 1} d\vec{\theta} \tag{D.11}$$

$$= \frac{\Delta(\vec{\alpha} + \vec{n})}{\Delta(\vec{\alpha})}, \text{where, } \vec{n} = \{n_i\}_{i=1}^{K} \tag{D.12}$$

We see that the above equation (D.12) is independent of $\vec{\theta}$ as it is integrated out. This distribu-

tion is called the Polya distribution or Dirichlet-Multinomial distribution.

Property 4: A Gamma function over variables taking discrete values behaves as, $\Gamma(x) = x!$, therefore, $\Gamma(x+1) = x \cdot \Gamma(x)$

Beta-Bernoulli model. The Beta-Bernoulli model is a special case of the Dirichlet-Multinomial model where the number of possibilities at each draw is limited to two *e.g.,* a head or toss. Therefore all the properties mentioned in the Dirichlet-Multinomial model are applicable to the Beta-Bernoulli model by setting $K = 2$.

Gamma-Poisson model. Let $\mathcal{D} = \{x_1, x_2, \ldots, x_N\}$ be a set of observations that are independent and identically distributed according to a Poisson distribution $p(x|\lambda)$ with parameter λ. Then the likelihood of the observations is given as:

$$p(\mathcal{D}|\lambda) = \Pi_{i=1}^{N} p(x_i|\lambda) \tag{D.13}$$

$$\propto \lambda^{\sum_i x_i} e^{-N\lambda} \tag{D.14}$$

The conjugate prior for the Poisson distribution is the Gamma distribution which is given by:

$$p(\lambda|\alpha, \beta) = \lambda^{\alpha-1} e^{-\beta\lambda} \frac{\beta^\alpha}{\Gamma(\alpha)} \tag{D.15}$$

where $\Gamma(\alpha)$ is the Gamma function, as described above giving $(\alpha - 1)!$ for integers. α and β are the hyper-parameters, where α is the mean prior observations seen in β intervals.

Now writing the Bayes rule with Poisson likelihood and Gamma prior we get,

$$p(\lambda|\mathcal{D}, \alpha, \beta) \propto \lambda^{\sum_i x_i} e^{-N\lambda} \lambda^{\alpha-1} e^{-\beta\lambda} \frac{\beta^\alpha}{\Gamma(\alpha)} \tag{D.16}$$

$$\propto \lambda^{\alpha-1+\sum_i x_i} e^{-\lambda(N+\beta)} \tag{D.17}$$

which in its mathematical form resembles the Gamma distribution. By setting $\alpha^* = \sum_i x_i + \alpha$ and $\beta^* = N + \beta$, we get:

$$p(\lambda|\mathcal{D}, \alpha, \beta) = \frac{\lambda^{\alpha-1+\sum_i x_i} e^{-\lambda(N+\beta)} (N+\beta)^{\sum_i x_i + \alpha}}{\Gamma(\sum_i x_i + \alpha)} \tag{D.18}$$

$$= \text{Gamma}(\alpha^*, \beta^*) \tag{D.19}$$

The predictive distribution for new data under this Gamma-Poisson model follows a Negative-Binomial distribution.

Normal-normal model. Let $\mathcal{D} = \{x_1, x_2, \ldots, x_N\}$ be a set of observations that are independent and identically distributed according to a normal distribution with mean μ and variance σ^2. Then the likelihood of the data is given by:

$$p(\mathcal{D}|\mu,\sigma^2) = \Pi_{i=1}^{N} p(x_i|\mu,\sigma^2) \tag{D.20}$$

$$\propto \exp(-\frac{N}{2\sigma^2}(\bar{x}-\mu)^2) \tag{D.21}$$

where $\bar{x}$ is the empirical mean obtained as $\bar{x} = \sum_i x_i/N$. Now let us assume for simplicity that the variance σ^2 is known and kept fixed. We are interested in adding a prior distribution over the parameter μ. The prior is a natural conjugate prior *i.e.,* a normal distribution with mean μ_0 and variance σ_0^2. The the prior distribution is given by:

$$p(\mu|\mu_0,\sigma_0^2) \propto \exp(-\frac{1}{2\sigma_0^2}(\mu-\mu_0)^2) \tag{D.22}$$

$$\propto \mathcal{N}(\mu|\mu_0,\sigma_0^2) \tag{D.23}$$

The calculation of the posterior distribution entails a more involved derivation for which we refer to (Murphy, 2007). The final for of the updated mean μ_n is then given as

$$\mu_n = \sigma_n^2\left(\frac{\mu_0}{\sigma_0^2} + \frac{N\bar{x}}{\sigma^2}\right), \text{ where } \sigma_n^2 = \frac{\sigma^2\sigma_0^2}{N\sigma_0^2+\sigma^2} \tag{D.24}$$

The posterior predictive is then given by

$$p(x|\mathcal{D}) = \int p(x|\mu,\sigma^2)p(\mu|\mathcal{D})d\mu \tag{D.25}$$

$$= \int \mathcal{N}(x|\mu,\sigma^2)\mathcal{N}(\mu|\mu_n,\sigma_n^2)d\mu \tag{D.26}$$

$$= \mathcal{N}(x|\mu_n,\sigma_n^2+\sigma^2) \tag{D.27}$$

Bibliography

Andrade, E. L., Blunsden, S., and Fisher, R. B. (2006). Hidden markov models for optical flow analysis in crowds. In *IEEE International Conference on Pattern Recognition*, pages 460–463, Washington, DC, USA.

Andriluka, M., Roth, S., and Schiele, B. (2008). People-tracking-by-detection and people-detection-by-tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Atev, S., Masoud, O., and Papanikolopoulos, N. (2006). Learning traffic patterns at intersections by spectral clustering of motion trajectories. In *IEEE International Conference on Intelligent Robots and Systems*, pages 4851–4856.

Basak, J., Sudarshan, A., Trivedi, D., and Santhanam, M. S. (2004). Weather data mining using independent component analysis. *Journal of Machine Learning Research*, **5**, 239–253.

Besnerais, G., Bercher, J., and Demoment, G. (1999). A new look at entropy for solving linear inverse problems. *IEEE Trans. on Information Theory*, **45**(5), 1565–1578.

Blei, D. and Lafferty, J. (2006a). A correlated topic model of science. *Annals of Applied Statistics*, **1**(1), 17–35.

Blei, D. M. and Lafferty, J. D. (2006b). Dynamic topic models. In *International Conference on Machine Learning*, pages 113–120.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003a). Latent dirichlet allocation. *Journal of Machine Learning Research*, **3**(4-5), 993–1022.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003b). Latent dirichlet allocation, c implementation. `http://www.cs.princeton.edu/~blei/lda-c/`.

Bobick, A. F. and Davis, J. W. (2001). The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **23**(3), 257–267.

Boiman, O. and Irani, M. (2007). Detecting irregularities in images and in video. *International Journal of Computer Vision*, **74**(1), 17–31.

Bradley, D. and Bagnell, J. A. D. (2008). Differentiable sparse coding. In *Proceedings of Neural Information Processing Systems 22*.

Brand, M. and Kettnaker, V. (2000). Discovery and segmentation of activities in video. *IEEE Transactions Pattern Analysis and Machine Intelligence*, **22**.

Brand, M., Oliver, N., and Pentland, A. (1997). Coupled hidden markov models for complex action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Breitenstein, M. D., Grabner, H., and Gool, L. V. (2009a). Hunting nessie – real-time abnormality detection from webcams. In *IEEE International Workshop on Visual Surveillance*.

Breitenstein, M. D., Reichlin, F., Leibe, B., Koller-Meier, E., and Gool, L. V. (2009b). Robust tracking-by-detection using a detector confidence particle filter. In *IEEE International Conference on Computer Vision*.

Buntine, w. (2002). Variational extensions to em and multinomial pca. In *European Conference in Machine Learning*, pages 23–34. Springer-Verlag.

Chan, M. T., Hoogs, A., Bhotika, R., Perera, A., Schmiederer, J., and Doretto, G. (2006). Joint recognition of complex events and track matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1615–1622.

Chen, S. S. and Gopalakrishnan, P. S. (1998). Speaker, environment and channel change detection and clustering via the bayesian information criterion. In *DARPA Broadcast News Transcription and Understanding Workshop*, pages 127–132.

Chien, J.-T. and Wu, M.-S. (2008). Adaptive bayesian latent semantic analysis. *IEEE Transactions on Audio, Speech, and Language Processing*, **16**(1), 198–207.

Corduneanu, A. and Bishop, C. M. (2001). Variational bayesian model selection for mixture distributions. *Artificial Intelligence and Statistics*, **8**.

Eagle, N. and Pentland, A.and Lazer, D. (2009). Inferring social network structure using mobile phone data. *Proceedings of the National Academy of Sciences (PNAS)*.

Efros, A. A., Berg, A. C., Mori, G., and Malik, J. (2003). Recognizing action at a distance. In *IEEE International Conference on Computer Vision*, Nice, France, October 2003.

Emonet, R., Varadarajan, J., and Odobez, J.-M. (2011a). Extracting and locating temporal motifs in video scenes using a hierarchical non parametric bayesian model. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Emonet, R., Varadarajan, J., and Odobez, J.-M. (2011b). Multi-camera open space human activity discovery for anomaly detection. In *IEEE Conference on Audio and Video Signal based Surveillance*.

Farrahi, K. and Perez, D. G. (2008). What did you do today?: discovering daily routines from large-scale mobile data. In *ACM International Conference on Multimedia*, pages 849–852, New York, NY, USA.

Faruquie, T. A., Kalra, P. K., and Banerjee, S. (2009). Time based activity inference using latent Dirichlet allocation. In *British Machine Vision Conference*, London, UK.

Fei-Fei, L. and Perona, P. (2005). A bayesian hierarchical model for learning natural scene categories. *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 524–531.

Figueiredo, M. A. F. and Jain, A. K. (2002). Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**(3), 381–396.

Fu, Z., Hu, W., and Tan, T. (2005). Similarity based vehicle trajectory clustering and anomaly detection. In *IEEE Conference on Image Processing*, volume 2.

Gaussier, E. and Goutte, C. (2005). Relation between plsa and nmf and implication. In *ACM Conference on Research and Development in Information Retrieval*, pages 601–602.

Girolami, M. and Kabán, A. (2003). On an equivalence between PLSI and LDA. In *ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pages 433–434.

Gohr, A., Hinneburg, A., Schult, R., and Spiliopoulou, M. (2009). Topic evolution in a stream of documents. In *SIAM International Conference on Data Mining*, pages 859–870.

Gong, S. and Xiang, T. (2003). Recognition of group activities using dynamic probabilistic networks. In *IEEE International Conference on Computer Vision*, volume 2, pages 742–749.

Griffiths, T., Steyvers, M., Blei, D., and Tenenbaum, J. (2004). Integrating topics and syntax. In *Advances in Neural Information Processing Systems*.

Griffiths, T., Steyvers, M., and Tenenbaum, J. (2007). Topics in semantic representation. *Psychological Review*, **114**, 211–244.

Griffiths, T. L. and Steyvers, M. (2002a). Prediction and semantic association. In *Neural Information Processing Systems*, pages 11–18.

Griffiths, T. L. and Steyvers, M. (2002b). A probabilistic approach to semantic representation. In *Annual Conference of the Congintive Science Society*.

Griffiths, T. L. and Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, **101**, 5228–5235.

Gruber, A., Rosen-Zvi, M., and Weiss, Y. (2007). Hidden topic Markov model. In *International Conference on Artificial Intelligence and Statistics*, San Juan, Puerto Rico.

Harris, C. and Stephens, M. (1988). A combined corner and edge detector. pages 147–151.

Heili, A., Chen, C., and Odobez, J.-M. (2011). Detection-based multi-human tracking using a crf model. In *The Eleventh IEEE International Workshop on Visual Surveillance*.

Heinrich, G. (2004). Parameter estimation for text analysis. Technical report.

Hervieu, A., Bouthemy, P., and Cadre, J.-P. L. (2008). A statistical video content recognition method using invariant features on object trajectories. *IEEE Transactions on Circuits and Systems for Video Technology*, **18**(11), 1533–1543.

Hofmann, T. (2001). Unsupervised learning by probability latent semantic analysis. *Journal of Machine Learning Research*, **42**, 177–196.

Horn, B. K. P. and Schunk, B. G. (1981). Determining optical flow. In *Artificial Intelligence*, pages 185–203.

Hospedales, T., Gong, S., and Xiang, T. (2009). A Markov clustering topic model for mining behavior in video. In *IEEE International Conference on Computer Vision*, Kyoto, Japan.

Hoyer, P. O. (2005). Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, **5**(2), 1457–1470.

Hu, W., Tan, T., Wang, L., and Maybank, S. (2004). A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man and Cybernatics*.

Hu, W. H. W., Xiao, X. X. X., Fu, Z. F. Z., Xie, D., Tan, T. T. T., and Maybank, S. (2006). A system for learning statistical motion patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**(9), 1450–64.

Huynh, T., Fritz, M., and Schiele, B. (2008). Discovery of activity patterns using topic models. In *Ubiquitous Computing*, pages 10–19.

Jiao, L., Wu, Y., Wu, G., Chang, E. Y., and Wang, Y.-f. (2004). Anatomy of a multicamera video surveillance system. *Multimedia Systems*, **10**(2), 144–163.

Jouneau, E. and Carincotte, C. (2011a). Mono versus multi-view tracking-based model for automatic scene activity modeling and anomaly detection. In *IEEE International Conference on Advanced Video and Signal-Based Surveillance*.

Jouneau, E. and Carincotte, C. (2011b). Particle-based tracking model for automatic anomaly detection. In *International Conference on Image Processing*.

Ke, Y., Sukthankar, R., and Hebert, M. (2007). Event detection in crowded videos. In *IEEE International Conference on Computer Vision*.

Kim, J. and Grauman, K. (2009). Observe locally, infer globally: a space-time mrf for detecting abnormal activities with incremental updates. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Kratz, L. and Nishino, K. (2009). Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In *IEEE International Conference on Computer Vision and Pattern Recognition*.

Kuettel, D., Breitenstein, M. D., Gool, L. V., and Ferrari, V. (2010). What's going on? discovering spatio-temporal dependencies in dynamic scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1951–1958.

Laptev, I. and Lindeberg, T. (2005). Space-time interest points. *International Journal Computer Vision*, **64**(2-3), 107–123.

Laptev, I. and Pérez, P. (2007). Retrieving action in movies. In *IEEE International Conference on Computer Vision*.

Lee, D. D. and Seung, H. S. (2001). Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, volume 13, pages 556–562.

Li, J., Gong, S., and Xiang, T. (2008). Global behaviour inference using probabilistic latent semantic analysis. In *British Machine Vision Conference*.

Li, J., Gong, S., and Xiang, T. (2009). Discovering multi-camera behaviour correlations for on-the-fly global activity prediction and anomaly detection. In *IEEE International Workshop on Visual Surveillance*, Kyoto, Japan.

Liu, J. S. (1994). The collapsed gibbs sampler in bayesian computations with applications to a gene regulation problem. *Journal of the American Statistical Association*, **89**(427).

Lowe, D. G. (2000). Towards a computational model for object recognition in it cortex. In *IEEE International Workshop on Biologically Motivated Computer Vision*, pages 20–31.

Loy, C. C., Xiang, T., and Gong, S. (2009). Multi-camera activity correlation analysis. In *IEEE International Conference on Pattern Recognition*.

Makris, D. and Ellis, T. (2003). Automatic learning of an activity-based semantic scene model. *IEEE International Conference on Advanced Video and Signal Based Surveillance*, **2**(1), 183–188.

McLachlan, G. and Peel, D. (2005). *Finite Mixture Models*. Wiley-Interscience.

Mehran, R., Oyama, A., and Shah, M. (2009). Abnormal crowd behavior detection using social force model. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 935–942.

Minka, T. and Lafferty, J. (2002). Expectation-propagation for the generative aspect model. In *Eighteenth Conference on Uncertainty in Artificial Intelligence*. Elsevier.

Morris, B. T. and Trivedi, M. M. (2008a). Learning and classification of trajectories in dynamic scenes: A general framework for live video analysis. *IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 154–161.

Morris, B. T. and Trivedi, M. M. (2008b). A survey of vision-based trajectory learning and analysis for surveillance. *IEEE Transactions on Circuits and Systems for Video Technology*, **18**, 1114–1127.

Murphy, K. P. (2007). Bayesian statistics: a concise introduction. Technical report.

Naturel, X. and Odobez, J.-M. (2008). Detecting queues at vending machines: a statistical layered approach. In *IEEE Conference on Pattern Recognition*.

Ng, A. Y., Jordan, M. I., and Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. In *Neural Information Processing Systems*, pages 849–856.

Nguyen, M. H., Torresani, L., De la Torre, F., and Rother, C. (2009). Weakly supervised discriminative localization and classification: a joint learning process. In *Proceedings of International Conference on Computer Vision*.

Niebles, J. C., Wang, H., and Fei-Fei, L. (2008). Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, **79**(3), 299–318.

Odobez, J. and Bouthemy, P. (1995). Robust multiresolution estimation of parametric motion models. *Journal of Visual Communication and Image Represenangation*, **6**(4), 348–365.

Pasupathy, A. and Connor, C. E. (2002). Population coding of shape in area v4. *Nature Neuroscience*, **5**(12), 1332–1338.

Patron, A., Marszalek, M., Zisserman, A., and Reid, I. (2010). High five: Recognising human interactions in tv shows. In *British Machine Vision Conference*, pages 50.1–50.11.

Porikli, F. (2004). Learning object trajectory patterns by spectral clustering. In *IEEE International Conference on Multimedia and Expo*, volume 2, pages 1171–1174.

Prabhakar, K., Oh, S., Wang, P., Abowd, G., and Rehg, J. (2010). Temporal causality for the analysis of visual events. In *IEEE International Conference on Computer Vision and Pattern Recognition*.

Quelhas, P., Monay, F., Odobez, J.-M., Gatica-perez, D., and Tuytelaars, T. (2005). A thousand words in a scene. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Richardson, S. and Green, P. J. (1997). On bayesian analysis of mixtures with an unknown number of components. *Journal of the Royal Statistical Society - Series B: Statistical Methodology*, **59**(4), 731–792.

Sacchi, C. and Regazzoni, C. S. (2000). A distributed surveillance system for detection of abandoned objects in unmanned railway environments. *IEEE Transactions on Vehicular Technology*, **49**(5), 2013–2026.

Saleemi, I., Hartung, L., and Shah, M. (2010). Scene understanding by statistical modeling of motion patterns. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Schuldt, C., Laptev, I., and Caputo, B. (2004). Recognizing human actions: A local svm approach. In *IEEE International Conference on Pattern Recognition*.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, **6**(2), 461–464.

Sethuraman, J. (1994). A constructive definition of dirichlet priors. *Statistica Sinica*, **4**, 639–650.

Shet, V. D., Harwood, D., and Davis, L. S. (2005). Vidmap: video monitoring of activity with prolog. In *Proceedings IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 224–229.

Singh, V. K., Wu, B., and Nevatia, R. (2008). Pedestrian tracking by associating tracklets using detection residuals. In *IEEE Motion and Video Computing*, pages 1–8.

Smith, K. C., Quelhas, P., and Gatica-Perez, D. (2006). Detecting abandoned luggage items in a public space. In *IEEE Performance Evaluation of Tracking and Surveillance Workshop (PETS)*.

Stauffer, C. and Grimson, W. E. L. (1999). Adaptive background mixture models for real-time tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 246–252.

Stauffer, C. and L.Grimson, E. (2000). Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**, 747–757.

Steyvers, M. and Griffiths, T. (2007). Probabilistic topic models. *Latent Semantic Analysis A Road to Meaning*, **22**(7), 1028–1040.

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2006). Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, **101**(476), 1566–1581.

Tommasi, C. and Kanade, T. (1991). Detection and tracking of point features. *International Journal of Computer Vision*.

Tran, S. D. and Davis, L. S. (2008). Event modeling and recognition using markov logic networks. In *European Conference on Computer Vision*, pages 610–623.

Tritschler, A. and Gopinath, R. (1999). Improved speaker segmentation and segments clustering using the bayesian information criterion. In *Sixth European Conference on Speech Communication and Technology*.

Turaga, P., Chellappa, R., Subrahmanian, V. S., and Udrea, O. (2008). Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*.

Varadarajan, J. and Odobez, J. (2009). Topic models for scene analysis and abnormality detection. In *IEEE International Workshop on Visual Surveillance*, Kyoto, Japan.

Varadarajan, J., Emonet, R., and Odobez, J.-M. (2010). Probabilistic latent sequential motifs: Discovering temporal activity patterns in video scenes. In *British Machine Vision Conference*, pages 117.1–117.11, Aberystwyth.

Varadarajan, J., Emonet, R., and Odobez, J. (2012). Bridging the Past, Present and Future; Modeling Scene Activities from Event Relationships and Global Rules. In *IEEE Conference on Computer Vision and Pattern Recognition*, Providence, USA.

Vogler, C. and Metaxas, D. (1999). Parallel hidden markov models for american sign language recognition. In *IEEE International Conference on Computer Vision*, pages 116–122.

Wallach, H. M. (2006). Topic modeling: beyond bag-of-words. In *International Conference on Machine Learning*, pages 977–984, Pittsburgh, Pennsylvania.

Wang, C. and Blei, D. (2009). Decoupling sparsity and smoothness in the discrete hierarchical Dirichlet process. In *Neural Information Processing Systems*, pages 1982–1989.

Wang, C., Blei, D. M., and Heckerman, D. (2008a). Continuous time dynamic topic models. In *Conference on Uncertainty in Artificial Intelligence*.

Wang, X. and McCallum, A. (2006). Topics over time: A non-Markov continuous-time model of topical trends. In *ACM Conference Knowledge Discovery and Data Mining*, Philadelphia, USA.

Wang, X., Tieu, K., and Grimson, E. L. (2004). Learning semantic scene models by trajectory analysis. In *European Conference on Computer Vision*, volume 14, pages 234–778.

Wang, X., Mccallum, A., and Wei, X. (2007). Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *IEEE International Conference on Data Mining*.

Wang, X., Ma, K. T., Ng, G., and Grimson, E. (2008b). Trajectory analysis and semantic region modeling using nonparametric bayesian models. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Wang, X., Ma, X., and Grimson, E. L. (2009). Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **31**(3), 539–555.

Williamson, S., Wang, C., Heller, K., and Blei, D. (2009). Focused topic models. In *NIPS workshop on Applications for Topic Models: Text and Beyond*, Whistler, Canada.

Wren, C., Ivanov, Y., Leigh, D., and Westhues, J. (2007). The merl motion detector dataset. In *Workshop on Massive Datasets (MD)*, pages 10–14.

Wu, S., Moore, B., and Shah, M. (2010). Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Xiang, T. and Gong, S. (2005). Video behavior profiling and anomaly detection without manual labeling. *IEEE International Conference on Computer Vision*.

Xiang, T. and Gong, S. (2006). Model selection for unsupervised learning of visual context. *International Journal of Computer Vision*, **69**(2), 181–201.

Xiang, T. and Gong, S. (2008). Video behavior profiling for anomaly detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **30**(5), 893–908.

Yang, Y., Liu, J., and Shah, M. (2009). Video scene understanding using multi-scale analysis. In *IEEE International Conference on Computer Vision*, Kyoto, Japan.

Yao, J. and Odobez, J.-M. (2007). Multi-layer background subtraction based on color and texture. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1–8.

Yao, J. and Odobez, J.-M. (2008a). Fast human detection from videos using covariance features. In *IEEE 8th International workshop on Visual Surveillance workshop*.

Yao, J. and Odobez, J.-M. (2008b). Multi-camera 3d person tracking with particle filter in a surveillance environment. In *16th European Signal Processing Conference*.

Yi Zhang, Jeff Schneider, A. D. (2010). Learning compressible models. In *Proceedings of SIAM Data Mining (SDM) Conference*.

Yilmaz, A., Javed, O., and Shah, M. (2006). Object tracking: A survey. *ACM Comput. Surv.*, **38**(4).

Zen, G. and Ricci, E. (2011). Earth mover's prototypes: a convex learning approach for discovering activity patterns in dynamic scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Zhang, Z., Huang, K., and Tan, T. (2006). Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes. In *IEEE International Conference on Pattern Recognition*, volume 3, pages 1135–1138, Hong Kong.

Zhou, B., Wang, X., and Tang, X. (2011). Random field topic model for semantic region analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Curriculum Vitae

Name: Jagannadan Varadarajan
Nationality: Indian
Permanent Address: Chemin du Scex, 10, Martigny 1920, Switzerland.
Email: vjagan@gmail.com
www: www.idiap.ch/~vjagann

General areas of interest:
Video Analysis, Topic Models, Graphical Models, Machine Learning, Computer Vision and Image Processing

---

Education:

1. *June 2008 - June 2012*
   PhD in Electrical Engineering, École Polytechnique Fédérale de Lausanne, Switzerland.
   Idiap Research Institute, Martigny, Switzerland.
   Supervisor: Dr. Jean-Marc Odobez,
   Date of thesis defence: 09 July 2012.
2. *June 2003 - March 2005*
   Master of Technology, Computer Science at Sri Sathya Sai University, Puttaparthi, A.P, India.
3. *June 2001 - March 2003*
   Master of Science, Mathematics at Sri Sathya Sai University, Puttaparthi, A.P, India.
4. *June 1998 - March 2001*
   Bachelor of Science, Mathematics at Sri Sathya Sai University, Puttaparthi, A.P, India.

Professional Experience:

1. *June 2008 - August 2012*
   Research Assistant at the Idiap Research Institute, Martigny, Switzerland.
2. *June 2007 - May 2008*
   Scientist, Computation and Decision Sciences Lab, GE Global Research, JFWTC, Bangalore, India.

3. *October 2006 - May 2007*
   Research Associate, Document Analysis Team, HP Labs, Bangalore, India.

Research Projects directly involved in:

1. Swiss National Science Foundation (HAI 198), `http://www.snf.ch/E/`

2. VANAHEIM (European FP7 project), `http://www.vanaheim-project.eu`

Additional expertise:

1. Programming languages: C, C++, Java, Python.

2. Scripts: Shell, Perl, SED

3. Packages: Matlab, OpenCV.

4. Applications: Netbeans, Code::Blocks, Visual Studio.

5. Operating Systems: UNIX/Linux, Windows.

References:

1. Dr. Jean-Marc Odobez, Idiap Research Institute, odobez@idiap.ch

2. Dr. Babu O Narayanan, GE Global Research, babu@ge.com

3. Dr. Rémi Emonet, Idiap Research Institute, remi.emonet@idiap.ch

---

List of Publications:

Thesis:

1. "Sequential Topic Models for Mining Recurrent Activities and their Relationships: Application to long term video recordings", PhD Thesis, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, 2012.

Book Chapters

1. Jagannadan Varadarajan, Rémi Emonet, and Jean-Marc Odobez. **Sparsity in Topic Models, Practical Applications of Sparse Modeling: Biology, Signal Processing and Beyond.**, *To be published in MIT Press publications.*, 2012.

Journals

1. **Jagannadan Varadarajan**, Rémi Emonet, and Jean-Marc Odobez. **A Sequential Topic Model for Mining Recurrent Activities from Audio and Video Data Logs**. *under minor revision in International Journal of Computer Vision.*
2. Rémi Emonet, **Jagannadan Varadarajan** and Jean-Marc Odobez. **Temporal Analysis of Motif Mixtures using Dirichlet Processes**. *under revision in IEEE Transactions on Pattern Analysis Machine Intelligence.*

Conferences

1. **Jagannadan Varadarajan**, Rémi Emonet and Jean-Marc Odobez. **Bridging the Past, Present and Future: Modeling Scene Activities From Event Relationships and Global Rules**. *In IEEE conference on Computer Vision and Pattern Recognition (CVPR).* Rhode Island, USA, 2012.
2. Rémi Emonet, **Jagannadan Varadarajan** and Jean-Marc Odobez. **Multi-camera Open Space Human Activity Discovery for Anomaly Detection**. *In IEEE conference on Advanced Video Signal and Surveillance (AVSS).* Klagenfurt, Austria, 2011.
3. Rémi Emonet, **Jagannadan Varadarajan** and Jean-Marc Odobez. **Extracting and Locating Temporal Motifs in Video Scenes Using a Hierarchical Non Parametric Bayesian Model**. *In IEEE conference on Computer Vision and Pattern Recognition (CVPR).* Coloradao Springs, USA, 2011.
4. **Jagannadan Varadarajan**, Rémi Emonet and Jean-Marc Odobez. **Sparsity Constraint for Topic Models - Application to Temporal Activity Mining**. *In NIPS 2010, Workshop on Practical Applications of Sparse Modeling: Open Issues and New Directions.* Vancouver, Canada, 2010
5. **Jagannadan Varadarajan**, Rémi Emonet and Jean-Marc Odobez. **Probablistic Latent Sequential Motifs: Discovering Sequential Patterns in Video Scenes**. *In British Machine Vision Conference (BMVC).* Aberystwyth, UK, 2010.

6. **Jagannadan Varadarajan** and Jean-Marc Odobez. **Topic Models for Scene Analysis and Abnormality Detection**. *In ICCV-12th IEEE International Workshop on Visual Surveillance.* Kyoto, Japan, 2009.
7. **Jagannadan Varadarajan** and Sriganesh Madhvanath. **Digital ink to Form Alignment for Electronic Clipboard devices**. *In IAPR International workshop on Document Analysis Systems (DAS)* Nara, Japan, 2008.
8. **Jagannadan Varadarajan**, M.C. Prakash, R.Raghunatha Sarma, GV Prabhakar Rao. **Feature Extraction and Image Registration of Color Images using Fourier Bases**. *In IEEE International Conference on Image Processing* Genova, Italy, 2005.