Non-Gaussian data modeling with hidden Markov models

Elise Epaillard

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy (Electrical and Computer Engineering) at

Concordia University

Montréal, Québec, Canada

October 2017

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: **Elise Epaillard**

Entitled: **Non-Gaussian data modeling with hidden Markov models**

and submitted in partial fulfillment of the requirements for the degree of

Doctor Of Philosophy (Electrical and Computer Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

____Chair
Dr. Luis Amador

____ External Examiner
Dr. Mohand Saïd Allili

____ External to Program
Dr. Wen Fang Xie

____ Examiner
Dr. Abdessamad Ben Hamza

____ Examiner
Dr. Yousef R. Shayan

____Thesis Supervisor
Dr. Nizar Bouguila

Approved by ____
Dr. Wei-Ping Zhu, Graduate Program Director

Monday, September 25, 2017 ____
Dr. Amir Asif, Dean
Faculty of Engineering and Computer Science

Abstract

Non-Gaussian data modeling with hidden Markov models

Elise Epaillard, Ph.D.

Concordia University, 2017

In 2015, 2.5 quintillion bytes of data were daily generated worldwide of which 90% were unstructured data that do not follow any pre-defined model. These data can be found in a great variety of formats among them are texts, images, audio tracks, or videos. With appropriate techniques, this massive amount of data is a goldmine from which one can extract a variety of meaningful embedded information. Among those techniques, machine learning algorithms allow multiple processing possibilities from compact data representation, to data clustering, classification, analysis, and synthesis, to the detection of outliers. Data modeling is the first step for performing any of these tasks and the accuracy and reliability of this initial step is thus crucial for subsequently building up a complete data processing framework. The principal motivation behind my work is the over-use of the Gaussian assumption for data modeling in the literature. Though this assumption is probably the best to make when no information about the data to be modeled is available, in most cases studying a few data properties would make other distributions a better assumption. In this thesis, I focus on proportional data that are most commonly known in the form of histograms and that naturally arise in a number of situations such as in bag-of-words methods. These data are non-Gaussian and their modeling with distributions belonging the Dirichlet family, that have common properties, is expected to be more accurate. The models I focus on are the hidden Markov models, well-known for their capabilities to easily handle dynamic ordered multivariate data. They have been shown to be very effective in numerous fields for various applications for the last 30 years and especially became a corner stone in speech processing. Despite their extensive use in almost all computer vision areas, they are still mainly suited for Gaussian data modeling. I propose here to theoretically derive different approaches for learning and applying to real-world situations hidden Markov models based on mixtures of Dirichlet, generalized Dirichlet, Beta-Liouville distributions, and mixed data. Expectation-Maximization and variational learning approaches are studied and compared over several data sets, specifically for the task of detecting and localizing unusual events. Hybrid HMMs are proposed to model mixed data with the goal of detecting changes in satellite images corrupted by different noises. Finally, several parametric distances for comparing Dirichlet and generalized Dirichlet-based HMMs are proposed and extensively tested for assessing their robustness. My experimental results show situations in which such models are worthy to be used, but also unravel their strength and limitations.

Acknowledgments

I would like to thank Dr. Nizar Bouguila for trusting me and giving me the chance to cross the Atlantic and pursue my PhD degree in Montreal. I am grateful for all the discussions about theoretical concepts we had, for all the publication tips he gave me, and also for presenting me with the opportunity to work at Sofdesk where I have been learning tons of new skills.

I also would like to thank my committee member for taking the time to read my reports, review them, and giving me valuable comments at the different stages of my degree.

I am very thankful to my labmates Mohamad, Elias, Jai, Taoufik, Gita, Asama, Ehsan, Walid, Muhammad, Nuha, Eddy for the fun times at Concordia, your help when needed, and for creating that overall feeling that we are all going to make it through!

And finally, big thanks to my family and friends for supporting me during this PhD experience: Leo for making my life a blessing every single day, my close family in France for always being supportive, and my friends Clem, Kévane, Juliette, Greg, Nico, Anne-Sophie, Thibaut, Maud, the 1405 crew, the capoeira family, and everyone I have spent some quality time with, for making life in Montreal so fun!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

“Data is the new oil.”

- Clive Humby (UK mathemetician and architect of *Tesco’s Clubcard*), 2006

“The difference between oil and data is that the product of oil does not generate more oil, whereas the product of data will generate more data.”

- Piero Scaruffi (cognitive scientist and author of *History of Silicon Valley*), 2016

In 2015, 2.5 quintillion bytes (2.5×10^{18}) were generated every single day, which to be stored on blu-ray discs would require 10^6 discs which, piled up, would be reach the height of 4 times the Eiffel tower. Every minute, more than 200 million emails were sent, 12 hours of videos uploaded on Youtube only, 277,000 tweets, 216,000 Instagram posts, more than 800,000 Facebook status and comments along with 136,000 pictures were shared. Estimates give 90% of the generated data is unstructured such as texts, images, audio tracks, or videos. These vertiginous figures keep on increasing every year, in an exponential way and predictions give estimates of 50,000 GB per second of Internet global traffic for 2018. So yes, *data is the new oil* that, if used along with appropriate techniques, can give humans at large a new understanding of their reality, change their lives by shaping their digital experience from their tastes and habits, and maybe soon solve global political, societal, and biological issues by finding patterns in these data to reduce crime, reduce costs, improve urban planning, waste management, cure diseases,... [4, 5]

Among these techniques, machine learning algorithms allow multiple processing possibilities from compact data representation, to data clustering, classification, analysis, and synthesis, to the detection of outliers. Data modeling is the first step for performing any of these tasks and the accuracy and reliability of this initial step is thus crucial for subsequently building up a complete data processing framework.

Machine learning models can be divided into two categories, discriminative and generative. Discriminative models are typically used for classification or categorization of the data, and include Support Vector Machines (SVMs), linear and logistic regressions, neural networks, conditional random fields, or random forests. Generative approaches aim at modeling how the data is generated. The resulting model can be then used to address various other tasks. Generative models include hidden Markov models (HMMs), mixture models, Naive Bayes, or Latent Dirichlet allocation.

Most probabilistic models fall into the second category by modeling via distributions how the data have been generated. The simplest generative probabilistic models are the probability density distributions (pdf) that are widely used to easily model a set of data or a parameter. For instance waiting queues are often modeled with a Poisson distribution and measurement errors as a Gaussian.

When data are multimodal, a single distribution does not have the ability to represent these data accurately and mixture models can be used for modeling them.

1.1 Mixture models and use of the Gaussian

Mixture models are a weighted sum of distributions that can be expressed as:

$$p(\vec{x}|\theta) = \sum_{m=1}^{M} w_m p_m(\vec{x}|\theta_m) \ , \tag{1}$$

where the p_m's are call the components of the mixture, and the w_m's the weights of the components. $\theta = (\theta_1, \ldots, \theta_M)$ denotes the set of parameters of the distributions of the mixture model, with θ_m the parameters related to the distribution p_m.

Their modeling ability is far more powerful that the one of a single distribution and have been proven for multiple applications, such as count data modeling and classification [6], classification schemes for myoelectric signals [7], object classification and forgery detection [8]. Mixture models have been mostly studied for the Gaussian [7,9,10], often for their mathematical convenience, with the Central-Limit theorem justifying this assumption. However, for this to hold, the training data set would need to contain a huge number of samples which is, most of the times, not the case and even not useful or desirable as it increases the computational time for the model estimation. Therefore, when some data properties are known (for instance the support, the positivity or negativity), it seems more reasonable to choose to model the data by probability distributions that share the same properties which will result in a more accurate, more compact model and which will, in some cases allow the generation of new data sharing the properties of the modeled ones. Previous works have shown this for many distributions such as the Dirichlet [11], generalized Dirichlet (GD) [12], and Beta-Liouville [13] for proportional data (or vectors of proportions, see below), the inverted Dirichlet [14], generalized inverted Dirichlet [8], Rayleigh [15,16], Weibull [17], Student's-t [18], Poisson [19], Langevin [20], and asymmetric Gaussian [21].

1.2 Proportional Data

The work realized in this thesis mainly focuses on *proportional data* (also called *compositional data*) that most commonly appear in the form of histograms and naturally arise in a number of situations such as in bag-of-words methods. A proportional data sample $\vec{x} = (x_1, ..., x_D)$ has the following two properties:

- $x_d > 0, \forall d$,
- $\sum_{d=1}^{D} x_d = 1$,

which has for direct consequence that the support of the x_d's is limited to $]0,1[$.

These data are clearly non-Gaussian (positivity, finite support, asymmetry) and their modeling with distributions belonging the exponential family such as the Dirichlet, the generalized Dirichlet, or the Beta-Liouville, that have common properties, rather than with Gaussian distributions, have been shown to be more accurate in a number of applications in the case of mixture modeling [11–13].

1.3 Distributions definitions

I define hereafter the three main distributions used in this thesis and set the notations for their parameters.

The D-dimensional Dirichlet distribution can be expressed as:

$$Dir(\vec{x}|\vec{\alpha}) = \frac{\Gamma(\sum_{d=1}^{D} \alpha_d)}{\prod_{d=1}^{D} \Gamma(\alpha_d)} \prod_{d=1}^{D} x_d^{\alpha_d - 1} , \tag{2}$$

where Γ denotes the Gamma function and $\vec{\alpha} = (\alpha_1, ..., \alpha_D)$, the distribution's parameters, all real and strictly positive. This distribution is defined for positive data that sum up to one: $\vec{x} \in \mathbb{R}_+^D$ and $\sum_{d=1}^{D} x_d = 1$.

The D-dimensional generalized Dirichlet distribution, which embeds the Dirichlet as a special case is defined as:

$$GD(\vec{x}|\vec{\alpha}, \vec{\beta}) = \prod_{d=1}^{D} \frac{\Gamma(\alpha_d + \beta_d)}{\Gamma(\alpha_d)\Gamma(\beta_d)} x_d^{\alpha_d - 1} \Big(1 - \sum_{r=1}^{d} x_r\Big)^{\nu_d} , \tag{3}$$

where $\vec{\alpha} = (\alpha_1, ..., \alpha_D)$ and $\vec{\beta} = (\beta_1, ..., \beta_D)$ are the distributions' parameters, all real and strictly positive. ν_d is a combination of these parameters and equals to $\beta_d - \alpha_{d+1} - \beta_{d+1}$, if $d \neq D$, and to $\beta_D - 1$, otherwise.

Finally, the D-dimensional Beta-Liouville distribution, which also embeds the Dirichlet as a special case is defined as:

$$BL(\vec{x}|\vec{\alpha}, \alpha, \beta) = \frac{\Gamma(\sum_{d=1}^{D} \alpha_d)\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \prod_{d=1}^{D} \frac{x_d^{\alpha_d - 1}}{\Gamma(\alpha_d)} \Big(\sum_{d=1}^{D} x_d\Big)^{\alpha - \sum \alpha_d} \Big(1 - \sum_{d=1}^{D} x_d\Big)^{\beta - 1} , \tag{4}$$

where $\vec{\alpha} = (\alpha_1, ..., \alpha_D)$, α and β are the distributions' parameters, all real and strictly positive.

These latter two distributions are defined for positive data that sum up to less than one: $\vec{x} \in \mathbb{R}_+^D$ and $\sum_{d=1}^{D} x_d < 1$, which corresponds to proportional data of dimension $(D+1)$.

1.4 Hidden Markov models

Mixtures models, though proving good performance for many applications, do not grasp any temporal or ordering information in the data. When solving problems involving temporal data (video processing, speech processing, sensor temporal measurements for instance), other models are needed.

The models I focus on are the hidden Markov models, well-known for their capabilities to easily handle dynamic ordered multivariate data. They have been shown to be very effective in numerous fields for various applications for the last 30 years and especially became a corner stone in speech processing. Despite their extensive use in almost all computer vision areas, they are still mainly used for Gaussian data modeling [22–24].

The mathematical foundations of the HMMs have been first proposed by Baum [25] but their wider practical use is mostly due to the early works of Rabiner and Juang in the field of speech processing [26]. Based on [26], a first-order HMM is a probabilistic model assuming an ordered observation sequence $O = \{O_1, ..., O_T\}$ to be generated by some hidden states, each of them being associated with a probability distribution that governs the emission of the observed data. The hidden states $H = \{h_1, ..., h_T\}$, $h_i \in [1, K]$, with K being the number of states, are assumed to form a Markov chain.

At each time t, a new state is entered based on a transition matrix $B = \{b_{jj'} = P(h_t = j'|h_{t-1} = j)\}$ that specifies the transition probabilities between the states. Once in the new state, an observation is generated following its associated probability distribution. For discrete observation symbols taken from a vocabulary $\vartheta = \{v_1, ..., v_S\}$, the emission matrix is defined as $V = \{V_i(k) = P(O_t = v_k|h_t = i)\}$, $[t, k, i] \in [1, T] \times [1, S] \times [1, K]$. For continuous observation vectors, the emission probability distributions are usually taken as Gaussian, defined by their means μ and covariance matrices Σ, denoted $\varphi = (\mu, \Sigma)$ for concision, or mixtures of Gaussian [22–24, 26]. In the latter case, denoting the set of mixture components as $L = m_1, \ldots, m_M$, a matrix $C = \{c_{i,j} = P(m_t = j|h_t = i)\}$, $j \in [1, M]$, is defined with M being the number of mixture components associated with state j (which can be assumed to be the same for all states without loss of generality). An initial probability distribution π controls the initial state. I denote an HMM as $\lambda = \{B, V, \pi\}$ or $\{B, C, \varphi, \pi\}$.

HMMs are well suited for classification tasks and rely on the probability of an observation sequence given a model λ, that is computed using a forward-backward procedure. Model training consists in the estimation of the parameters that maximize the probability of a given set of observations and is addressed with the Baum-Welch algorithm, an Expectation-Maximization process. Finally, finding the most probable sequence of states and mixture components that generated a series of observations can be solved with the Viterbi algorithm [26].

The number of hidden states and the parameters' initial values are set a priori. Both are strongly linked to the model's performance. Indeed, the former is a trade-off between performance and complexity [27], while the latter leads the Baum-Welch procedure to converge towards the closest local maximum of the likelihood function, not guaranteed to be the global one given its high modality [24]. Finally, the choice of the emission distributions also has to be set in advance and can thus only be induced by the data or the nature of their features along with their properties.

1.5 Contributions

The first attempt for designing an HMM suited to proportional data has been published in [28], where the learning equation for a Dirichlet-based HMM (based on a Expectation-Maximization approach) are derived and enhanced performance compared to a Gaussian-based model shown over synthetic data. No further study of the performance of such models on real-world data has been performed before the work presented in this dissertation. Also, no other HMM has been proposed since then for proportional data modeling though other distributions and learning approaches could have been used too. This is this gap that the present work is striving to bridge.

In Chapter 2, a preliminary use example of Dirichlet-based HMMs is presented in the context of texture classification in images. The classification capability of this type of HMM over time-series of proportional features is demonstrated thanks to the use of a very reduced vocabulary of 10 words in a bag-of-visual-word approach. This work can also be found in the conference paper referenced as [29].

In Chapter 3, the learning equations of the Baum-Welch approach for generalized Dirichlet and Beta-Liouville based HMMs are derived. Experiments on synthetic data along with an application to action recognition in video sequences are presented for comparison between the Dirichlet and the generalized Dirichlet assumptions. These experiments can also be found in the conference paper references as [30]. Finallly, larger scale experiments with the Beta-Liouville are presented over several data sets with the goal of detecting and localizing unusual events in video surveillance footage, reaching state-of-the-art detection rates. The theory and these experiments have been published in the journal paper referenced as [31].

With the good modeling capabilities shown in the aforementioned works, one can expect that a more accurate estimation of the HMM parameters is achieved, better performance could be obtained. Therefore, in Chapter 4, I propose to derive the equation for the variational learning of the Dirichlet and the generalized Dirichlet based HMMs. Leading experiences in the context of unusual event detection again, I was able to show how changing the learning technique of the model can significantly improve the detection rate of anomalies in videos. This work can also be found in the journal paper referenced as [32].

Chapter 5 presents a simple way of combining several existing HMMs models into a hybrid HMM for modeling mixed discrete/continuous and continuous/continuous data. Experiments on synthetic data show encouraging results and provide insights for future developments for modeling this utmost complex type of data. An original application for change detection in a pair of satellite images illustrates how this new model can be used. This work has been presented at the MMSP'15 conference where it received a Top 10% paper award [33].

In Chapter 6, I propose new parametric distances between the proposed Dirichlet and generalized Dirichlet HMMs. This includes the research of meaningful quantities for characterizing and assessing the performance of the proposed similarity measures, as well as experiments over synthetic data. An illustration for the use of such distances over real-world data provides hints about what information they can unravel in the scope of the main application studied in Chapters 3 and 4. The work presented in this chapter is submitted

Finally, a general conclusion closes this dissertation and proposes open theoretical, practical, and applicative questions for future work on the topic (Chapter 7).

Chapter 2

Preliminary work on the Dirichlet-based HMM

2.1 Introduction

As mentioned in Chapter 1, the equations of the Dirichlet-based HMM (i.e., HMM with Dirichlet mixtures as emission probability functions) have been first proposed in [28]. For recall, a D-dimensional Dirichlet distribution is expressed as

$$Dir(\vec{x}|\vec{\alpha}) = \frac{\Gamma(\sum_{d=1}^{D} \alpha_d)}{\prod_{d=1}^{D} \Gamma(\alpha_d)} \prod_{d=1}^{D} x_d^{\alpha_d - 1} \ , \tag{5}$$

where Γ denotes the Gamma function and $\vec{\alpha} = [\alpha_1, ..., \alpha_D]$, the distributions' parameters, all real and strictly positive. This distribution is defined for positive data that sum up to one: $\vec{x} \in \mathbb{R}_+^D$ and $\sum_{d=1}^{D} x_d = 1$. As a preliminary work, this model has been applied to real-world data in the context of texture classification. The choice of the application has been mostly driven by the availability of public data sets, the usual representation of textures in the form of histograms (proportional data), as well as by the need of using data embedding apparent or latent dynamics. Furthermore, as for a first application of this rather complex model, images were easier to handle than videos. Later on, this algorithm has been successfully applied to video sequences (see Chapters 3 and 4).

Most of our natural environment can be interpreted as textures, in the sense it is mainly composed of more or less repetitive pattern involving some spatial dynamics. Their thorough study is of great importance and texture categorization, segmentation, and synthesis have been the topics of unnumbered studies with applications as various as medical imaging [34], special effects [35], remote sensing [36], etc. HMMs have been seldom used for texture classification purpose though their capabilities to unravel latent structures of textures that direct observations could not provide alone has been brought to light long ago [37]. Their capabilities have then been investigated in the wavelet domain [38–40] giving promising results, but no further study seems to have been led with HMMs on this topic. As dynamical processes, they seem however very appropriate to model textures that naturally embed spatial dynamics.

One popular texture representation approach is the bag-of-visual-words (BoVW) method. First introduced as the image counterpart of document classification works, it has been broadly employed and shown to be efficient for texture classification tasks [41]. Among the numerous studies employing it, [42] proposes two approaches known for their good performance in text classification, namely Probabilistic Latent Semantic Indexing and Non-negative Matrix Factorization. A sparse image representation is used by first detecting local regions of interest with Harris- and Hessian-affine detectors and then extracting SIFT [43] in these regions. A global texton-dictionary is built with an optimal number of textons found to be 500 as a trade-off between computational load and performance. Classification is finally led in an unsupervised manner. In [44], local regions are also detected and then normalized to be mapped into subspaces. Texton-dictionaries of size 100 are built from different feature types (intensity and gradient) with several linear and non-linear embedding methods and a Support Vector Machine (SVM) classifier is used. In [41], a method based on the projection of a set of points from a high-dimensional space to a randomly chosen low-dimensional subspace, referred to as random projections (RP), is developed. Images are seen as an ensemble of patches from which RP features are extracted. Rotation-invariance is obtained by sorting the pixels intensity or the pixels differences projections. Each texture class is represented by a BoVW model of 40 to 80 textons, leading to a full dictionary of

1000 to 2000 words. Classification is performed using nearest-neighbor (NN) and SVM classifiers. [45] makes use of fractal analysis to describe textures at different resolutions. The authors provide an interesting comparison of multifractal spectrum (MFS) and histograms. Their analysis shows that MFS overcomes the issue of the loss of spatial distribution information inherent to histograms by providing a multilayer aspect to key points count. In comparison, the use of series of histograms over image patches is proposed to help maintaining partial information about the spatial distribution of the key points (textons). This makes the comparison of my method with this one of high interest.

In all these works, the dictionary size used goes from one hundred up to few thousands words. The main contributions presented in this chapter are:

- to show that a well-tailored classifier can achieve state-of-the-art results with a dictionary as small as of 10 visual-words, leading to the most compact representation ever reported
- to confirm that the results on Dirichlet-based HMMs (HMMD) found with synthetic data in [28] hold with real-world data.

Furthermore, my original representation uses series of histograms and gives rise, for comparison purpose with the NN classifier, to the need of a similarity measure for multiple ordered histograms, leading to a generalization of the Bhattacharyya distance.

This short study has been presented at the 3rd International Conference on Adaptive and Intelligent Systems (ICAIS'14) in Bournemouth, UK. The publication is referred as [29] in the Bibliography section.

In the following the method's steps are detailed in Section 2.2 before reporting experimental results in Section 2.3, and concluding in Section 2.4.

2.2 Method steps

A common way to obtain proportional data from images is to work with a BoVW strategy. In this preliminary work, the quality of the extracted features is not crucial to assess the performance of a classifier relatively to other classifiers, and I therefore simply use SIFT [43] and a two-stage k-means clustering to build a texton-dictionary.

HMMD has in practice a big restriction on input data dimension that I empirically found to be of the order of 10 (higher dimensions lead intermediate matrices to be singular, causing invertibility issues). While this number is very small as a dictionary size, I chose to give it a try and to perform my experiments with a global dictionary of only 10 words. To the best of my knowledge, this constitutes the smallest dictionary ever reported for experiments over large texture data sets, several orders lower than usual ones (500 words in [42] and 1000 in [41] for a 25-class representation). However, it is worthwhile noticing that 10 words theoretically have the capability to represent much more than 25 classes. Supposing that only 2 value levels are allowed for each word (e.g., *present* or *absent*), 2^{10} configurations are possible. Though the intra-class variability probably reduces the effective number of classes that could be discriminated with such a naive representation, I state that there is no need of hundreds of words to represent distinctively a few dozens classes.

2.2.1 Textons dictionary building

The SIFT detector proposed in [43] is used on each image with dense sampling (no local region detector). Depending on the texture class, the number of descriptors extracted goes from around 100 up to 7000. As the global dictionary aimed to be used is very compact, it is problematic to have such inter-class variation in the number of extracted descriptors. Indeed, if all kept when picking up the 10 words forming the dictionary, the words will most likely be all taken from the classes with the richest representations and thus, not be representative of the whole data set. Another issue might come from the too numerous descriptors, leading to a computationally impractical clustering task. In order to avoid these potential issues, a k-means clustering is performed on every image lowering its number of descriptors down to 60. As the focus of the work presented in this chapter is on the classification method, no further study has been conducted for optimizing this value which is a trade-off between the image representation precision and the computational load of this step. A set of N images is randomly selected from each of the c classes to form a training set. By the aforementioned process, 60 SIFT features per image are extracted. A second k-means clustering is then applied to the gathering of all the training set features, i.e., $60 \times c \times N$ SIFT vectors, from which 10 centers are obtained, forming the global dictionary. These centers are later on referred to as SIFT-words.

2.2.2 Series of histograms computation

As mentioned earlier, textures are here considered as being quite repetitive patterns involving some spatial dynamics. The embedding of these spatial dynamics into every image representation is performed by scanning the image following a predefined path and building a corresponding series of histograms. Each image is divided into P patches of equal size and the scan path is arbitrarily defined as going from the upper row to the bottom one, describing them from the left to the right. For each patch, all the originally extracted features (i.e., the ones obtained before any clustering) are assigned to their nearest SIFT-word in the dictionary. This operation results in a series of P 10-bin histograms (which can also be interpreted as a 2D-histogram) representing the image. This process is used in both training and testing phases.

2.2.3 Model computation and Classification

One HMM is trained for each texture class using the N available training series of histograms. Two types of emission probability distributions are compared; Dirichlet mixture models in the setting developed in [28] and Gaussian mixtures models which are the most commonly used emission functions in HMMs applications. Same numbers of states K and mixture components M, have been used in both cases, empirically determined by making them vary from 1 to 4, values which keep the model computation tractable. It has been noticed that when the product KM is too large (above 12 here), some class models fail to be estimated (matrices singularities appear at some point, stopping the whole estimation process). The best results have been obtained for KM products equal to 8 and 9.

Changing the probability distributions from mixtures of Gaussian to mixtures of Dirichlet involves modifying the initialization and the parameters estimation step in the EM-algorithm (i.e., the M-step), keeping the rest of the HMM estimation algorithm unchanged. The details of the distributions substitution are discussed in [28]. The model's parameters initialization has been shown intractable if accurately computed [28]. Following [28], KM single Dirichlet distributions are initialized and then assigned to the HMM states in an ordered manner, while other parameters are randomly initialized. More details about the initialization are given in Chapter 3.

As a new image arrives, all its SIFT features are computed and allocated to the different histograms bins depending on their location and value. Once the series of histograms is built, its likelihood with respect to each class model is computed using a forward algorithm and the image is classified into the category of highest likelihood.

2.2.4 Baseline method

To quantify the performance of the HMM classifiers, a baseline method using an NN classifier is implemented. The Bhattacharyya distance between two histograms G and H

$$d(G,H) = \sqrt{1 - \sum_i \sqrt{G(i)}\sqrt{H(i)}}\ , \tag{6}$$

where i denotes the bin number [46], can be straightforwardly generalized to series of T histograms by

$$d(G_T,H_T) = \sqrt{1 - \frac{1}{T}\sum_{t=1}^{T}\sum_i \sqrt{G_t(i)}\sqrt{H_t(i)}}\ . \tag{7}$$

However, this distance, denoted *BD1* later, is clearly not robust to translation. Working at the patch level, if P patches are used then, P translated patterns exist. Hence, I propose the following patch-translation robust distance, denoted *BD2*:

$$d_{tr}(G_T,H_T) = \min_{p\in[1,P]} d(G_T,H_T^p)\ , \tag{8}$$

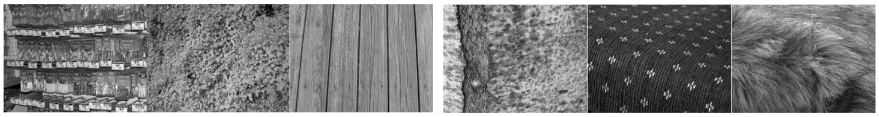


Figure 2.1: Sample images from the *UMD* (left) and *UIUC* (right) data sets.

where the superscript p stands for the translation of the first patch of G_T source image onto the p^{th} patch of H_T source image, spatially warping around the other patches.[1]

2.3 Experiments

2.3.1 Results

This section assesses the performance of the HMMD classifier on real-world proportional data compared to the HMMG and NN classifiers. From the results obtained with simple mixture models in [11, 47, 48], the use of the Dirichlet distribution is expected to improve the results obtained with a Gaussian-based model. To the best of my knowledge, this work represents the first use of HMMD on real-world data. The work of [28], which first introduced it, only presents experiments on synthetic data, generated from a known HMMD. Therefore, the capabilities of HMMD have to be investigated and leveraged on more realistic data. The experiments are performed on the two recent challenging natural texture images data sets from UIUC [49] and UMD [45], and compared with other BoVW-based methods.

The UIUC and UMD data sets each contain 1000 images of size 480x640 and 1280x960 pixels, respectively, divided up into 25 different classes (40 instances in each). They are challenging by the variety of 2D and 3D transformations and illumination variations present in it. For fair results comparison, the UMD data set is downsampled to the same resolution as the UIUC one. Sample images are presented in Fig. 2.1.

The experimental results presented here have been obtained by fixing $M = K = 3$ and $P = 12$ with random training sets of $T = 5, 10, 20$ images of each class, running the algorithm 50 times. Results are reported in Figs 2.2 and 2.3. The F-score [50] of the

[1] One can note that the length T of the series of histograms is typically equal to the number of patches P. However, the two distinct notations help to the clarity of the equations.

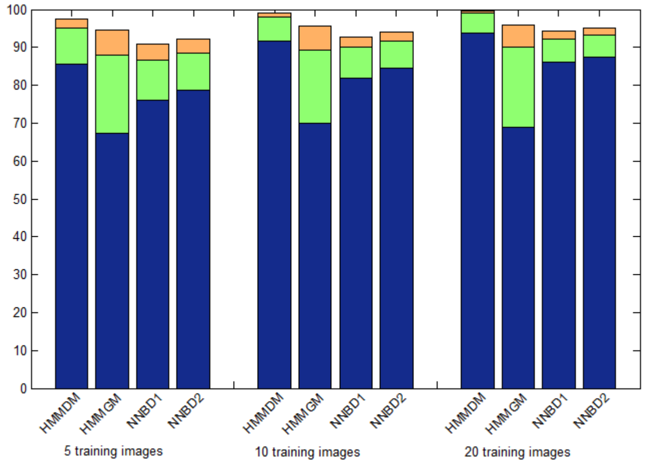


Figure 2.2: Accuracy (in blue) and rank statistics of order 3 (in green) and 5 (in orange) for the different classifiers on the *UMD* data set using 5, 10, and 20 training images per class.

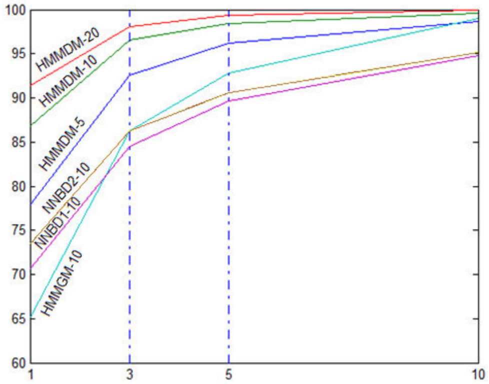


Figure 2.3: Rank statistics of order 1 (recall), 3, 5, and 10 for the different classifiers on the *UIUC* data set. The number after the '-' indicates the number of training images per class.

proposed approach with $T = 20$ is 94.3% on the UMD data set and 91.7% on the UIUC one (only the accuracy is reported on the graphs).

2.3.2 Comparison and interpretation

From these experiments, it is clear that the HMMD classifier can be used for real texture classification purpose and outperforms NN classifiers independently of the data set. It is worth noticing that the use of a dynamical model is not sufficient to get good classification accuracy. Appropriate emission distributions that match features properties is essential and the use of HMMs with non-suited probability emission functions dramatically degrades the results even compared to a simple NN classifier. Careful study of features properties is therefore crucial for choosing these distributions. Experiments with HMMG even led to the misclassification of entire classes which is critical for recognition applications. HMMD performs better than the other tested classifiers even with a training set reduced to 5 images. As expected, larger training sets improve the accuracy.

Rank statistics of order 3 and 5 show that most of time, even when not well-classified, the likelihood of the query with respect to its ground truth class is high. Introduction of a prior might help to improve the performance of the HMMD classifier (as shown later on in Chapter 4). For instance, no advantage has been taken from the information about SIFT density extracted at the first clustering level while it varies a lot depending on the texture class and could thus provide a valuable clue.

The proposed patch-translation robust Bhattacharyya distance (*BD2*) systematically improves the results of NN classification with respect to a simpler generalization (*BD1*) and gives acceptable accuracy on the UMD data set despite its simplicity. It however seems less versatile than HMM classifiers as the results on the UIUC data set are significantly more degraded with respect to the ones obtained with HMMD.

The proposed approach is compared with [41,42,44,45]. As said earlier, all these methods use BoVW strategies or similar texture representation and therefore constitute good references to assess the performance of the presented method. On the UIUC data set, [42] achieves 77.2% of accuracy using 500 textons (but unsupervised classification), [41], 95.8% with 1000 textons, and [44], 97.9% with 100 textons. MFS approach [45] leads to 92.7% of accuracy and the proposed one to 91.4% using 10 textons. On the UMD data set, the proposed approach achieves an accuracy of 93.9%, equal to the one reported in [45], while [41] and [44] reach 98.7% and 98.2%, respectively. In all cases, 20 training images per class have been used. The correct classification rate of the proposed method falls only few percents below current top state-of-the-art methods [41, 44], while using a dictionary 10 to 100 times smaller. This shows the potential power of the HMMD for proportional data modeling. Moreover, these results might be further improved with more appropriate features and optimized parameters.

One weakness of the histograms is the loss of spatial information in the image representation. I overcome this issue by considering series of histograms over patches while [45] proposes a multi-resolution representation with MFS features. Both methods achieve same or close results over the two tested data sets, showing that series of histograms can also help to solve this point while being more straightforward.

The good results obtained while using an approximate clustering method for features extraction (double stage k-means clustering, allowing easy convergence towards local extrema), tend to confirm the observation made in [51] regarding the estimation precision needed for the clusters centers to efficiently model data for classification tasks. Indeed, in their study, the authors have shown that good accuracy results could be obtained even if the dictionary textons were randomly selected. These conclusions open a window towards lower complexity algorithms in this field, especially for applications involving restricted memory size for image representation storage.

2.4 Conclusion

This preliminary work proposes a method based on a SIFT features double stage clustering in order to form a very compact 10-word dictionary. Series of histograms are used for partially keeping spatial information and HMMD for performing the classification, outperforming other tested classifiers. The initial guess that 10 words had the capability to discriminate well among few dozens of classes proves to be true. Despite the huge changes in scale, rotation, illumination and even more challenging 3D-transformations present in the used data sets, this roughly determined 10-word dictionary performs classification with a very acceptable accuracy. This raises the question of the necessity of the hundreds-word dictionaries most often used in the literature.

Chapter 3

EM-based learning of HMMs for proportional data and their application for anomaly detection and localization

3.1 Introduction

The development of informatics and cameras in the last decades led to the enforcement of numerous public security policies and private security expectations that naturally led to an outburst of research work on the topic of unusual event detection through video surveillance [3,23,52–56]. Indeed, the increasing use of CCTV cameras [57,58], traditionally monitored by human operators that simultaneously watch multiple screens for hours, invokes the need for a capability to assist them in the real-time detection of threats and anomalous events. Within the last few years, numerous acts of aggression and accidents that occurred in public spaces have been recorded by video surveillance cameras and publicly released afterward in the TV news, the internet, or the social media. This tends to demonstrate that such systems can significantly aid the society in large to avoid incidents if monitored automatically and in real-time, in order to communicate potential threats to the competent authorities as promptly as possible.

The release of real-world data sets [55,56] permits the development, testing, and quantification of the efficiency of various methods with the goal of detecting such malicious threats. From a probabilistic point of view, a threat (or anomaly) is a rare event, which means that its occurrence has a low probability. Alternatively, it can be defined as a "divergence from a dominant pattern" [59]. A threat can therefore take a countless number of different forms that mostly depend on the context; someone walking can be considered as a normal behavior in most cases, however, if the person walks in the opposite direction in an attempt to avoid the crowd, then it could disclose a possible threat. Thus, it is highly challenging to design an algorithm that is capable of recognizing, at early stages, a forthcoming threat. The strategy in this case is to initially model normal activities for which it is easy to obtain and process numerous non-malicious sample sequences, to subsequently define a threat as an outlier [23, 52, 53].

As mentioned in Chapter 1, HMMs are used in various fields such as speech processing [26], object and gesture classification [24, 27], and unusual event detection [23, 53]. It is particularly suited when working with dynamic data such as videos. This model is first trained on data (videos, audio signals, or most often features derived from them) whose characteristics are known in order to model well a specific class of data. Further, when a new data sequence arrives, the probability that it could have been generated by the model is computed. Depending on the result, it is either accepted or rejected as belonging to the tested class. In the context of threat (or event) detection, the model is usually trained on non-malicious video sequences and thus represents the normality. All detected outliers, corresponding to video frames (or parts of video frames) with a likelihood lower than a predefined threshold, are then considered as anomalous and as potential threats.

Nowadays, the training process of an HMM itself is quite standardized, and improvements mainly focus on its initialization in terms of topology and parameters estimation [24, 27,53]. One of the claimed reasons is that the initial tuning has a direct impact on the accuracy of the results. However, this model also involves emission probability functions for the observation generation, which are the probability distributions assigned to each state of the HMM. The choice of the employed type of emission probability distributions is very seldom discussed and Gaussian Mixture Models are typically applied as a standard [22–24, 27], for their mathematical convenience. However, the symmetric property of the Gaussian distribution as well as its unbounded support may not be accurate to model the emission of all data types, and enhanced results could be obtained using asymmetric emission probability distributions with compact support. The latter fact was exploited in [60] by revealing that Dirichlet mixture models yield better results than Gaussian ones in the context of texture classification, and as previously said in [28] and in the published work [29] presented in Chapter 2, in an HMM framework, for synthetic and real-world proportional data classification.

Although the Dirichlet mixture model offers a good alternative for proportional data modeling, it suffers from a restriction on the data it can accurately model. Indeed, the moment's equations of the Dirichlet impose on the covariance to be negative, which is not representative of the general case. From this observation, I propose in this chapter to investigate and assess the capabilities of HMMs using generalized Dirichlet (GD) and Beta-Liouville (BL) mixtures as emission probability distributions, as applied for threat and event detection related to public security, and to compare them to Dirichlet-based HMMs' performance. Both distributions belong to the exponential family, embed the Dirichlet distribution as a special case, and are not constrained by the sign of their covariance [61]. An initial work presented in Section 3.4.1 of this chapter and published in [30] has been done with GD-based HMMs and has shown superior performance on synthetic and real-world data classification in the context of an action recognition application. To the best of my knowledge, this is the first study attempting to integrate the generalized Dirichlet and Beta-Liouville distributions into the HMM framework. This latter distribution has the advantage of being represented by less parameters than the GD distribution and thus allows the model estimation to be computationally faster.

The contributions presented in this chapter are the following:

- The complete derivation of the equations for the integration of the GD and BL distributions mixtures into the HMM framework. (Section 3.3)

- The preliminary testing of these models over synthetic data and for an action recognition classification task. (Section 3.4.1)
- The application of these new models to three different scenarios related to the surveillance of public areas under real conditions. (Section 3.4.2)
- The analysis of the behavior of the models and the formulation of easily applicable rules for the tuning of the detection threshold, depending on the available data. (Section 3.4.2)

The theoretical parts of this chapter along with the experiments aiming at detecting unusual events in videos have been published in a paper in the journal *Pattern Recognition*, which is referenced in the Bibliography section as [31]. The experiments on synthetic data and on the action recognition data set have been presented at the 6th IAPR TC3 International Workshop on Artificial Neural Networks in Pattern Recognition (ANNPR'14), in Montreal, QC, Canada. The publication is referenced as [30] in the Bibliography section.

The rest of this chapter is organized as follows: Section 3.2 presents the related work focusing on both the theoretical contribution made in this chapter and on the issues of unusual event detection and some of the most common approaches. Section 3.3 details the equations of the GD- and BL-based HMMs, including the estimation of the distribution parameters. Section 3.4 presents the experimental work done with these models. Conclusive remarks are provided in Section 3.5.

3.2 Related work

At a time where video surveillance is widely used to insure security in public areas (e.g., airports, subway stations, university campus,...), real-time algorithms capable of detecting abnormal events or behaviors would be of great help for operators in charge of simultaneously monitoring sometimes numerous video screens.

Crowded scenes and dynamic environments are the most challenging scenarios for anomaly and event detection. In the literature, numerous tracking methods have been proposed [62–

64] but all suffer from degraded performance as soon as the crowd becomes too dense. Indeed, in crowded environments, the number of independent objects moving at the same time as well as the occlusions it involves prohibit the use of tracking for high performance. In [65], trajectories are used as main features but the authors propose to focus on trajectories of sufficient length that link predefined zones of interest such as windows of stores or paved areas of the road. The different groups of trajectories are modeled making use of first order Markov chains over trajectories elements called *tracklets*. In [66], tracking is only used as a support to detect human heads. Raw video input of these regions of interest is then directly used to feed a 3D convolutional neural network and detect specific actions such as being on the phone or pointing at something.

Dynamic background is an important restriction for tracking and more generally for all movement-based methods. Typically, background subtraction methods are used to detect foreground moving objects [3,62,67] but this requires to preprocess every frame, hence extra computations.

In response to the drawbacks of the tracking and background subtraction, methods that are working at a higher information level, from a global view of the situation, have been developed [22,23,52,54,59]. They rely on different features such as optical flow, pixel intensity gradients or dynamic textures. Bertini et al. [52] for instance present a purely data-driven method with no assumption made on the used type of data or the type of unusual event that can occur. The system is trained from video sequences in which no anomaly is present, forming a statistical description of it from spatio-temporal features. The anomaly detection is performed by computing the likelihood of a query sequence with respect to the normalcy model and comparing it to an adaptive threshold. In this framework, multi-scale observations as well as contextual information can also be taken into account. A recent survey [68] dedicates a full section to the detection of anomaly in crowded environments.

In this context, the use of HMMs would be of particular interest. Indeed, the data to be processed are dynamic and the nature of anomalies is unknown. HMMs can both model *normal* scenes and then determine whether an unseen video sequence deviates from this normality or not, which perfectly suits the anomaly detection purpose. In [52], the features used are histograms which, once normalized, can be seen as proportional data. The likelihood criterion for anomaly detection works well, though the classifier uses a simple adaptive threshold. Improved results are obtained to the cost of the addition of a second scale and the use of contextual information which help reducing the false positive rate but involve extra computations. To this end, the approach is an accumulation of processes and the detection results are the intersection of the results of the different processes. This superposition of processes is a clear limitation to the improvement of the global approach and the use of a unique, more powerful model and classifier can lead to a more compact representation of the data and thus a more accurate anomaly detection.

The use of HMMs has been popularized by Rabiner and Juang in [26] and a brief recall of their definition and main properties was provided is Chapter 1. Since this fundamental work, numerous extensions and adaptations of this model to specific applications have been developed. Among these extensions, the study of time-series generated from multiple processes and/or involving dynamics at different scales led to the development of factorial HMMs [69]. In this framework, each state is broken down into a collection of sub-states, often assumed to be independent at each time step for algorithmic complexity reduction.

State duration (i.e., state self-transitioning) has also been a study focus as classic HMMs naturally embed a geometric distribution as for state duration, with parameter depending on the state transition matrix [70]. Variable Duration HMMs have been a first attempt to modify the state duration probability distribution [71]. At each state transition, the duration of the new state is drawn from a probability mass function and the corresponding number of observations is generated before drawing a new state accordingly to the state transition matrix. An alternative, known as the Nonstationary HMM, that explicitly introduces the time variable into the state transition matrix is proposed in [70]. This model has been shown to be equivalent to the Variable Duration HMM though allowing an easier and computationally more efficient parameter estimation.

The most widely used estimation algorithm for HMMs is the so-called Baum-Welch algorithm, though its iterative nature can be prohibitive for some applications. [72] proposed a non-iterative method for parameters estimation. Based on subspace estimation, the idea has been theoretically derived in [73] and provides, on a few conditions, a computationally fast method to estimate HMMs with a finite discrete output.

HMMs have been initially developed for discrete and Gaussian data [26]. The multiplication of applications in domains such as weather forecast or medical studies raised the need for modifying the original HMM algorithm so it can efficiently work with new data types [28–30, 74, 75]. Longitudinal or panel data are time-series collected from multiple entities. An example of these data in the context of a medical study could be the evolution of some disease characteristics evaluated every day for a given period of time on a number of patients (see [76] for a concrete example). At the entity level, data heterogeneity is involved by the presence of multiple data sources. HMMs have been shown to be able to model this heterogeneity by introducing a random variable in the model, known as the *random effect*, that follows a predefined probability distribution. By doing so, the conditional independence of the observed data given the latent states assumption is relaxed. [75] provides a review of the use of these HMMs that are known in the literature as Mixed HMMs. With a similar aim of adaptivity, [74] discusses circular data processing, i.e., data taking cyclic values such as directions or angles. Von Mises, Wrapped Normal, and Wrapped Cauchy are proposed as state emission probability distributions to handle such data. A Maximum-Likelihood estimation algorithm is derived and applied to circular time-series. An HMM with a reduced sensitivity to outliers compared to the Gaussian has been proposed [77], using Student's t-mixtures as emission functions. In the same trend, the use of nonelliptically contoured distributions has been proposed in order to model heavy-tailed or skewed data, with an HMM based on multivariate normal inverse Gaussian distributions [78]. These two last approaches have shown superior performance compared to the Gaussian-based HMM for applications such as hand gesture, phonetic, or speaker recognition.

As for proportional data their modeling through HMMs has been first studied in [28] where Dirichlet mixtures are used as emission probability functions, involving a deep modification of the M-step of the Expectation-Maximization algorithm (EM) for the Dirichlet parameters estimation. A real-world application was presented in Chapter 2. Some limitations of the Dirichlet distribution have been brought to light by [60]. Adapting HMM to generalized Dirichlet and Beta-Liouville mixtures emission probability functions is expected to improve the modeling accuracy of a broader range of data. As mentioned in the introduction, this distribution generalizes the Dirichlet while allowing a more flexible covariance structure.

A few works have used the generalized Dirichlet and/or the Beta-Liouville distributions for machine learning applications. Among these, the GD distribution has been used for a document topic representation application in a Latent Dirichlet Allocation framework [79], and for the design of generative kernels for Support Vector Machine [80]. The latter has shown enhanced results compared to the use of the Dirichlet distribution for object recognition and content-based image classification. Recently, BL distributions have been exploited in the context of mixture modeling for facial expression and action recognition applications [13, 81].

As specified earlier, the main HMM framework for continuous data has been designed assuming the data to be Gaussian. However, there is a wide range of applications in which data or the used features are proportional. The Gaussian representation takes as an implicit assumption an unbounded support of the variables and is therefore not adapted for a precise modeling of this special data type. The following section develops the equations for the GD and BL distributions, that also belong to the exponential family. I explicitly build the parallel between the derivations for the two distributions, opening a window for an easy adaptation of HMMs to other exponential distributions. The need of distributions that are more complex than the Dirichlet is driven by the fact that it imposes an always negative data covariance. Therefore, distributions that relax this restriction might logically be more adapted to model all types of proportional data. GD and BL distributions both overcome this limitation and embed the Dirichlet distribution as a special case, with the advantage for the BL to be represented by a fewer number of parameters than the GD.

3.3 EM learning of the generalized Dirichlet and Beta-Liouville HMMs

3.3.1 Expected Complete-Data Log-Likelihood

I recall that D-dimensional GD and BL distributions are defined as

$$GD(\vec{x}|\vec{\alpha},\vec{\beta}) = \prod_{d=1}^{D} \frac{\Gamma(\alpha_d+\beta_d)}{\Gamma(\alpha_d)\Gamma(\beta_d)} x_d^{\alpha_d-1}\Big(1-\sum_{r=1}^{d} x_r\Big)^{\nu_d} , \tag{9}$$

$$BL(\vec{x}|\vec{\alpha},\alpha,\beta) = \frac{\Gamma(\sum_{d=1}^{D}\alpha_d)\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \prod_{d=1}^{D} \frac{x_d^{\alpha_d-1}}{\Gamma(\alpha_d)} \Big(\sum_{d=1}^{D} x_d\Big)^{\alpha-\sum\alpha_d} \Big(1-\sum_{d=1}^{D} x_d\Big)^{\beta-1} , \tag{10}$$

where Γ denotes the Gamma function and $\vec{\alpha} = (\alpha_1,\dots,\alpha_D)$, $\vec{\beta} = (\beta_1,\dots,\beta_D)$, α and β the distributions' parameters, all real and strictly positive. ν_d is a combination of these parameters and equals to $\beta_d - \alpha_{d+1} - \beta_{d+1}$, if $d \neq D$, and to $\beta_D - 1$, otherwise. Same notation $\vec{\alpha}$ is used in both cases as no confusion is possible between these two distributions in what follows. However, the numerical values in both cases have no reason being the same. These distributions are defined for positive data that sum up to less than one: $\vec{x} \in \mathbb{R}_+^D$ and $\sum_{d=1}^{D} x_d < 1$. This corresponds to proportional data of dimension $(D+1)$.

Changing the emission probability distribution type involves modifications in the EM estimation process. I set notations for the quantities

$$\gamma_{h_t,m_t}^t \triangleq p(h_t, m_t|x_0,\dots,x_T) , \tag{11}$$

and

$$\xi_{h_t,h_{t+1}}^t \triangleq p(h_t, h_{t+1}|x_0,\dots,x_T) , \tag{12}$$

that represent the estimates of the states and mixture components, and of the local states sequence given the whole observation set, respectively. The E-step leads to γ_{h_t,m_t}^t and

$\xi^t_{h_t,h_{t+1}}$ estimates for all $t \in [1, T]$. These two quantities are obtained using the initial parameters at step 1 and the result of the last M-step subsequently. They are computed using a similar forward-backward procedure (not detailed here) as for an HMM with mixtures of Gaussians.

The M-step aims at maximizing the data log-likelihood by maximizing its lower bound. If Z represents the hidden variables and X the data[1], the data likelihood $\mathcal{L}(\theta|X) = p(X|\theta)$ can be expressed as

$$\begin{aligned} E(X,\theta) - R(Z) &= \sum_Z p(Z|X)\ln(p(X,Z)) - \sum_Z p(Z|X)\ln(p(Z|X)) \\ &= \sum_Z p(Z|X)\ln(p(X)) \quad \text{(Bayes' rule)} \\ &= \ln(p(X))\sum_Z p(Z|X) \\ &= \ln(p(X)) = \mathcal{L}(\theta|X)\ , \end{aligned} \tag{13}$$

with θ, representing all the HMM parameters, omitted on the given variables side of all the quantities involved. $E(X,\theta)$ is the value of the complete-data log-likelihood with the true/maximized parameters θ. $R(Z)$ is the log-likelihood of the hidden data given the observations and has the form of an entropy representing the amount of information brought by the hidden data itself (see Equation (30) for the detailed form of $R(Z)$). As I estimate the complete-data log-likelihood using non-optimized parameters, I have $E(X,\theta,\theta^{old}) \leq E(X,\theta)$, and hence $E(X,\theta,\theta^{old}) - R(Z)$ is a lower bound of the data likelihood.

The key quantity for data likelihood maximization is the expected complete-data log-likelihood which is written as

$$E(X,\theta,\theta^{old}) = \sum_Z p(Z|X,\theta^{old})\ln(p(X,Z|\theta))\ . \tag{14}$$

In the case of a unique observation $X = \vec{x}$ (the case of multiple observation sequences

[1] Previously, I used $\vec{x}$ to represent a sample vector. X is here used to represent all the available data.

is addressed later), the complete-data likelihood can then be expanded as

$$p(X, Z|\theta) = p(h_0) \prod_{t=0}^{T-1} p(h_{t+1}|h_t) \prod_{t=0}^{T} p(m_t|h_t)p(x_t|h_t, m_t) \ , \tag{15}$$

and the different terms identified as

$$p(X, Z|\theta) = \pi_{h_0} \prod_{t=0}^{T-1} B_{h_t,h_{t+1}} \prod_{t=0}^{T} C_{h_t,m_t} BL(x_t|h_t, m_t) \ , \tag{16}$$

when BL emission probability distributions are used. Equation (16) can be written the same way for the GD distribution. I later refer to this equation as Equation (16'). Equation (78) is then substituted into Equation (16) and the logarithm is applied to the expression. Using the logarithm sum-product property, the complete-data log-likelihood is split up into eight terms:

$$\begin{aligned}\ln(p(X, Z|\theta)) = {} & \ln(\pi_{h_0}) + \sum_{t=0}^{T} \ln(C_{h_t,m_t}) + \sum_{t=0}^{T-1} \ln(B_{h_t,h_{t+1}}) \\ & + \sum_{t=0}^{T} \Bigg\{ \ln(\Gamma(\sum_{d=1}^{D} \alpha_d)) + \ln(\Gamma(\alpha + \beta)) - \ln(\Gamma(\alpha)) \\ & - \ln(\Gamma(\beta)) + (\alpha - \sum_{d=1}^{D} \alpha_d) \ln(\sum_{d=1}^{D} x_d) + (\beta - 1) \ln(1 - \sum_{d=1}^{D} x_d) \\ & + \sum_{d=1}^{D} \Bigg\{ (\alpha_d - 1) \ln(x_d) - \ln(\Gamma(\alpha_d)) \Bigg\} \Bigg\} \ . \end{aligned} \tag{17}$$

A similar equation can be derived from Equation (16'). Using Equation (17) or its GD counterpart into Equation (14), the expected complete-data log-likelihood can then be written:

$$\begin{aligned} E(X, \theta, \theta^{old}) = {} & \sum_{k=1}^{K} \sum_{m=1}^{M} \gamma_{k,m}^{0} \ln(\pi_k) + \sum_{t=0}^{T} \sum_{k=1}^{K} \sum_{m=1}^{M} \gamma_{k,m}^{t} \ln(C_{k,m}) \\ & + \sum_{t=0}^{T-1} \sum_{i=1}^{K} \sum_{j=1}^{K} \xi_{i,j}^{t} \ln(B_{i,j}) + L_{GD,BL} \ , \end{aligned} \tag{18}$$

with,

$$L_{GD}(\vec{\alpha},\vec{\beta}) = \sum_{t=0}^{T}\sum_{d=1}^{D}\sum_{k=1}^{K}\sum_{m=1}^{M}\gamma_{k,m}^{t}\times\Big\{\ln(\Gamma(\alpha_{k,m,d}+\beta_{k,m,d})) + (\alpha_{k,m,d}-1)\ln(x_d^t)$$

$$+ (\nu_{k,m,d}\ln(1-\sum_{r=1}^{d}x_r^t)) - \ln(\Gamma(\alpha_{k,m,d})) - \ln(\Gamma(\beta_{k,m,d}))\Big\}\,. \tag{19}$$

$$L_{BL}(\vec{\alpha},\alpha,\beta) = \sum_{t=0}^{T}\sum_{k=1}^{K}\sum_{m=1}^{M}\gamma_{k,m}^{t}\Big\{\ln(\Gamma(\sum_{d=1}^{D}\alpha_{k,m,d})) + \ln(\Gamma(\alpha_{k,m}+\beta_{k,m})) - \ln(\Gamma(\alpha_{k,m}))$$

$$-\ln(\Gamma(\beta_{k,m})) + (\alpha_{k,m}-\sum_{d=1}^{D}\alpha_{k,m,d})\ln(\sum_{d=1}^{D}x_d) + (\beta_{k,m}-1)\ln(1-\sum_{d=1}^{D}x_d)$$

$$+\sum_{d=1}^{D}\Big\{(\alpha_{k,m,d}-1)\ln(x_d) - \ln(\Gamma(\alpha_{k,m,d}))\Big\}\Big\}\,. \tag{20}$$

Equation (18) is set making use of the two following properties, in which I omit to mention θ^{old} in the given variables side of the probabilities involved. Using the independence of h_t and m_t from h_{t+1}, I get $p(Z|X) = p(h_t = k, m_t = m|X)p(h_{t+1} = k')$ with $\sum_{k'=1}^{K} p(h_{t+1} = k') = 1$. Similar steps bring $p(Z|X) = p(h_t = k, h_{t+1} = k'|X, m_t = m)p(m_t = m)$, with $\sum_{m=1}^{M} p(m_t = m) = 1$.

Furthermore, if $N \geq 1$ observation sequences are available, all can be used in order to avoid overfitting. In Equation (18), a sum over $n \in [1, N]$ has to be added in front of the entire formula. The sum over time goes then from 0 to T_n, the length of the n-th observation sequence, and the x_d's become $x_{d,n}$'s.

3.3.2 Update Equations of HMM and GD Parameters Estimation

The maximization of the expectation of the complete-data log-likelihood with respect to π, B, and C is solved by introducing Lagrange multipliers in order to take into account the constraints due to the stochastic nature of these parameters. The resulting update equations are:

$$\pi_k^{new} \propto \sum_{n=1}^{N}\sum_{m=1}^{M}\gamma_{k,m}^{0,n}, \quad B_{k,k'}^{new} \propto \sum_{n=1}^{N}\sum_{t=0}^{T_n-1}\xi_{k,k'}^{t,n}, \quad C_{k,m}^{new} \propto \sum_{n=1}^{N}\sum_{t=0}^{T_n}\gamma_{k,m}^{t,n}\,, \tag{21}$$

where k and k' are in the range $[1, K]$, and m, in the range $[1, M]$.

GD distributions parameters update is less straightforward. Indeed, a direct method would lead to maximize $L_{GD}(\vec{\alpha}, \vec{\beta})$ (Equation (19)). Instead of going through heavy computations, I propose to use a practical property of the GD distribution that reduces the estimation of a D-dimensional GD to the estimation of D independent one-dimensional Dirichlet distributions (i.e., Beta distributions). The latter is a known problem that can be solved using a simple Newton method [28, 82]. The use of this property calls the need to express the problem in a transformed space that I refer to as the W-space. Each observation $\vec{x}$ is transformed from its original space into its W-space by [60, 61]:

$$W_d = \begin{cases} x_d & \text{for } d = 1 \\ x_d/(1 - \sum_{i=1}^{d-1} x_i) & \text{for } d \in [2, D] \ . \end{cases} \tag{22}$$

In the transformed space, each W_d follows an independent Beta distribution with parameters (α_d, β_d), which is also a one-dimensional Dirichlet distribution. The estimation of the D Beta distributions governing the D W_d clearly leads to the complete characterization of the GD distribution governing the observation vector $\vec{x}$. In the M-step of the HMMGD algorithm, the update of the GD distribution parameters can thus be done using D times a process similar to the one used in [28], considering the transformed data instead of the original one. The other parameters (B, C, π, γ, ξ) are estimated from the original data.

The accurate initialization of the Dirichlet-based HMM parameters has been shown in [28] to be intractable as soon as the product KM grows up. The same holds for other exponential distributions and following their recommendations, KM single GD distributions are initialized with a method of moments that uses the transformed data (detailed in [83]) and are then assigned to the HMM states. The parameters π, C, and B, are randomly initialized for fair comparison with the Dirichlet-based HMM proposed in [28]. However, an initialization resulting from the clustering used in the method of moments is also possible. I observed throughout my experiments that, even with a random initialization, the convergence towards quite precise estimates is quick (within a few iterations).

3.3.3 BL Parameters Estimation

Contrary to the GD distribution, there is no known transformation for simplifying the estimation of the BL distribution parameters. A Newton-Raphson estimation method is then used for maximizing $L_{BL}(\vec{\alpha}, \alpha, \beta)$ (Equation 20). The equations for the estimation of a mixture of BL are developed in [84]. In my application, each BL distribution is separately estimated which simplifies the equations. The global estimation equation is given by:

$$\theta^{new} = \theta^{old} - H(\theta^{old})^{-1} \frac{\partial \mathcal{L}(X|\theta^{old})}{\partial \theta^{old}} \ . \tag{23}$$

The Hessian matrix is computed from the second order derivatives of the likelihood. The computation of these derivatives, which is straightforward and not detailed here, shows the independence between the vector of variables $\vec{\alpha}$ and the parameters (α, β). Therefore, the Hessian matrix is composed of two diagonal blocks, one of size 2×2 and the other one of size $D \times D$. The inverse of the matrix can also be computed blockwise: $H(\theta)^{-1} = \mathrm{diag}(H(\alpha_{1 \dots D})^{-1}, H(\alpha, \beta)^{-1})$.

The upper term can be written simplifying the expression given in [84],

$$H(\alpha_{1 \dots D})^{-1} = S^\star + \delta^\star a^\star a^{\star T} \ , \tag{24}$$

with

$$S^\star = \mathrm{diag}\Big\{ -\frac{1}{\gamma_{cum}\Psi_1(\alpha_1)}, \dots, -\frac{1}{\gamma_{cum}\Psi_1(\alpha_D)} \Big\} \ , \tag{25}$$

$$a^{\star T} = \Big(-\frac{1}{\gamma_{cum}\Psi_1(\alpha_1)}, \dots, -\frac{1}{\gamma_{cum}\Psi_1(\alpha_D)} \Big) \ , \tag{26}$$

$$\delta^\star = -\gamma_{cum}\Psi_1(\sum_{d=1}^{D} \alpha_d)\Big(1 + \Psi_1(\sum_{d=1}^{D} \alpha_d) \sum_{d=1}^{D} \frac{-1}{\Psi_1(\alpha_d)}\Big)^{-1} \ , \tag{27}$$

where $\gamma_{cum} = \sum_{t=1}^{T} \gamma^t$. [2]

The lower block of the inverse of the Hessian matrix can be computed by hand from the matrix itself as it is only a 2×2 matrix:

$$H(\alpha,\beta)^{-1} = \frac{\gamma_{cum}}{|H(\alpha,\beta)|} \times \begin{pmatrix} \Psi_1(\alpha+\beta) - \Psi_1(\beta) & -\Psi_1(\alpha+\beta) \\ -\Psi_1(\alpha+\beta) & \Psi_1(\alpha+\beta) - \Psi_1(\alpha) \end{pmatrix} , \tag{28}$$

where $|H(\alpha,\beta)|$ is the Hessian matrix determinant $|H(\alpha,\beta)| = (\gamma_{cum}^2 \times \{\Psi_1(\alpha)\Psi_1(\beta) - \Psi_1(\alpha+\beta)(\Psi_1(\alpha)+\Psi_1(\beta))\})^{-1}$.

By nature, the EM-algorithm is iterative and thus needs a stop parameter. As the data log-likelihood is maximized by the means of its lower bound, convergence of this bound can be used as such. This lower bound is given by $E(X,\theta,\theta^{old}) - R(Z)$ (see Equations (13) and (18)) and $R(Z)$ is derived using Bayes' rule:

$$\begin{aligned} p(Z|X) &= p(h_0)p(m_0|h_0)\prod_{t=1}^{T} p(h_t|h_{t-1})p(m_t|h_t) \\ &= p(h_0)\frac{p(m_0,h_0)}{p(h_0)}\prod_{t=1}^{T}\frac{p(h_t,h_{t-1})p(m_t,h_t)}{p(h_{t-1})p(h_t)} . \end{aligned} \tag{29}$$

Denoting $\eta_t \triangleq p(h_t|\vec{x})$ and using the independence properties set earlier, the following expression is derived (see detail in [28]):

$$\begin{aligned} R(Z) = &\sum_{k=1}^{K}\left[\eta_k^0 \ln(\eta_k^0) + \eta_k^T \ln(\eta_k^T) - 2\sum_{t=0}^{T}\eta_k^t \ln(\eta_k^t)\right] \\ &+ \sum_{t=0}^{T}\sum_{m=1}^{M}\sum_{k=1}^{K}\gamma_{k,m}^t \ln(\gamma_{k,m}^t) + \sum_{t=0}^{T-1}\sum_{k=1}^{K}\sum_{k'=0}^{K}\xi_{k,k'}^t \ln(\xi_{k,k'}^t) . \end{aligned} \tag{30}$$

This expression is valid for any type of emission function and stands for a unique observation sample. If more are to be used, a summation over them has to be added in front of the whole expression, the index T has to be adapted to the length of each sequence, and the η's have to be computed for every sample. At each iteration, the difference between the

[2] Here the γ subscripts k and m are not mentioned as these two parameters are fixed (as said earlier, each BL distribution is separately estimated). If $N > 1$ observations sequences are available, $\gamma_{cum} = \sum_{n=1}^{N}\sum_{t=1}^{T}\gamma^{n,t}$.

former and current data likelihoods is computed. Once it goes below a predefined threshold, the algorithm stops and the current parameters values are kept for the HMM. This threshold is a trade-off between estimates precision and computational time.

In the case of the HMMBL, the initial parameters are determined using the method of moments with the assumption that the distribution is a Dirichlet. The reason for this is the difficulty to have exact equations that can straightforwardly be solved for applying the method of moments. Attempts with simplified equations and random initialization for the BL parameters α and β yielded worst results than an initialization through a Dirichlet distribution. However, as expected, this initialization can give initial parameters quite far from their real value, especially for the parameter α, which is assumed to be the sum of the N first parameters. In order to avoid a divergence towards very high values, I use a quite large stop parameter (empirically set to 10^{-3} in my experiments) that still gives time to the algorithm to well estimate the other HMM parameters (transitions, mixing coefficients,...), while keeping the distributions parameters close to their initial estimates. The bias created by this initial estimates has less impact than random values would have as it is present in all models, and have thus a reduced impact in the likelihood computation in the classification step, especially in the case of multiple classes modeling. Furthermore, choosing a large stop parameter here reduces the computational time, which is a crucial specification for some of the applications presented hereafter.

3.4 Experiments

3.4.1 Succinct comparative study of the behaviors of the Dirichlet/GD-based HMMs

Synthetic data

As a first experiment, synthetic data are used in order to assess the benefits of using a more general emission probability distribution than Dirichlet. This preliminary work to the application of the new models to real-world data has only been led for the HMMGD model.

1000 observations sequences of length randomly taken in the range $[10, 20]$ are generated from known HMMs with randomly chosen parameters. The generation of GD samples is described in [61]. The generative state and mixture component are recorded for each sample. As proposed in [28], the performance is evaluated as the proportion of combinations of states and mixture components correctly retrieved by an HMM trained on the generated data. Multiple experiments are run by varying the number of states K, the number of mixture components M, and the data dimension D. The study of the influence of D is of particular importance as with proportional data, the greater D, the smaller the observation values. Too small values, through numerical processing, can lead to matrices invertibility issues which is not desirable for accurate estimation.

As stated earlier, the GD distribution relaxes the constraint on the sign of the data correlation coefficients. The GD-based model is then expected to give a more accurate data representation in the case the data are mostly positively correlated (i.e., more than half of the data covariance matrix terms are positive). On the other hand, with mostly negatively correlated data, HMMD should provide as good results with a reduced complexity. To verify this, data are generated from known HMMs and an HMMGD and a HMMD are used in order to retrieve the state and mixture component that generated every sample. However, data generated from HMMGDs with parameters randomly and uniformly drawn in the range $[1, 60]$ are quasi-automatically mostly positively correlated. To overcome this point I constrained some of the HMM parameters to follow a Dirichlet distribution expressed in the form of a GD distribution. The three following scenarios have been used:

1. Data generated from HMMDs only (Scenario 1),
2. Data generated from an hybrid HMM with, for each state, half of the components being Dirichlet and half GD distributions (Scenario 2),
3. Data generated from HMMGDs only (Scenario 3).

Extensive testing confirmed the expectations. Results are illustrated in Figure 3.1 using a *correlation ratio* which represents the number of positively correlated variables (minus the autocorrelations) over the number of negatively correlated ones. A ratio greater than 1 means the variables are mostly positively correlated and vice versa.

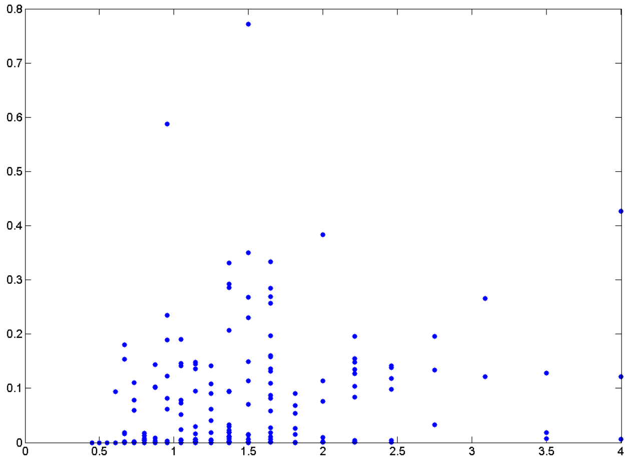


Figure 3.1: Gain of accuracy using HMMGD compared to HMMD in function of the variables correlation ratio. The gain of accuracy is computed as the difference between the two models' performance.

Experiments have been led with $K = 3$ and $M = 2$. For scenario 1, HMMGD has a 85.3 % accuracy and HMMD 84.9%, confirming that both work equally well. For scenarios 2 and 3, HMMGD has an accuracy of 81.2% and 89.7%, respectively, and HMMD of 77.6% and 80.1%, respectively. As soon as some data are positively correlated, HMMGD outperforms HMMD. It is observed in scenario 2 (correlation ratio close to 1), for unclear reasons, that it is more difficult for the HMMs to retrieve the correct state and component the sample comes from. Finally, the retrieval rate for data with a correlation ratio greater than 1 is of 86.1% for HMMGD and of 78.4% for HMMD, and of 84.8% and 83.2%, respectively, for correlation ratios smaller than 1. This shows HMMGDs overcome the weakness of HMMDs for positively correlated data.

Table 3.1 reports the results of experiments led fixing $D = 10$, generating 100 sequences only (because of time constraint), and letting K and M vary. According to the previous results, only mostly positively correlated data are considered here (scenario #3). For any combination (K, M), HMMGD achieves better results than HMMD showing the benefit of using HMMGD when proportional data are processed. As the product KM increases, the retrieval rate decreases which can be explained considering that the more distributions are present, the closer to each other they are, and the more difficult it is to clearly assign a sample to a distribution.

A bad initialization of the distribution parameters can give low retrieval rates. It can find

Parameters (K,M)	Product KM	HMMD (%)	HMMGD (%)
(2,2)	4	84.2	90.9
(2,3)	6	75.9	92.8
(3,2)	6	82.0	87.8
(2,4)	8	82.0	89.8
(4,2)	8	86.2	91.5
(3,3)	9	81.2	89.1
(3,4)	12	72.9	88.9
(4,3)	12	73.3	85.2
(4,4)	16	61.9	76.6
(5,5)	25	66.0	68.8
(10,5)	50	52.4	62.2

Table 3.1: HMMGD and HMMD retrieval rates with various (K, M) combinations

its origin in the convergence of the clustering algorithm, used as the first step of the method of moments, towards local extrema. To overcome this issue, the initialization process can be run several times and the comparison of the lower bound of the data likelihood with these initial parameters be used to choose the best ones. However, this requires extra computations and does not guarantee a good convergence of the clustering procedure, even within several attempts. Here the interest was only into the relative performance of the HMMGD compared to the HMMD then, this option has not been used. Instead, in order not to introduce any bias from this issue, a unique clustering algorithm is used for both initializations.

Figure 3.2 reports the results of experiments in which $K = 3$ and $M = 2$ and D increases until retrieval rates degrade dramatically. For scenario 1, equivalent results are obtained with both HMMs, HMMGD giving sometimes slightly better results at the cost of extra computations (not reported on Figure 3.2). In other cases, HMMGD systematically outperforms HMMD up to the point data dimension is too high to perform calculations accurately (intermediate matrices become singular). Fluctuations in the overall results are due to bad initializations that involve retrieval rates to dramatically drop on some isolated runs. The general shape of the curves and their relative distance clearly shows that, within an HMM framework, mixtures of GD distributions give the best results and allow working with data of higher dimension than Dirichlet ones. This performance improvement is obtained at the

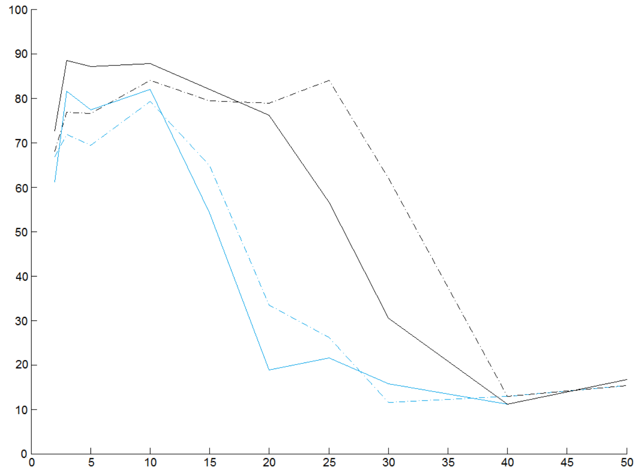


Figure 3.2: Retrieval rate (%) of HMMGD (in black) and HMMD (in blue) as a function of data dimension for scenarios 2 (dash lines) and 3 (solid lines)

cost of a more complex model involving $(2D-2)$ parameters to be estimated for every GD distribution compared to only D parameters for a Dirichlet one. These results are essential to target real applications for which HMMGD could be a potentially efficient tool.

Action recognition

Confirmation of the superior performance of the GD-based HMM is obtained on real-world data. The experiments are led on the Weizmann Action Recognition data set [85] which is composed of video sequences featuring 10 different actions (such as walk, run, jump,...) performed by 9 subjects. The features used are Histograms of Oriented Optical Flow [86] and 10-bin histograms are built, with each bin representing a range of optical flow angles with respect to the horizontal axis. The optical flow magnitude weights the contribution of each pixel to the histogram. [86] showed that good classification results could be obtained with features of dimension higher than 30 however, I preferred to use features of dimension 10 as, within my HMM-based framework, no improvement has been found when using more bins. Finally, for time savings, the frame rate of the video sequences is divided by 2.

Experiments are led using a Leave-One-Out cross validation, the results are averaged over 10 runs, and analyzed in terms of rank statistics. The optimal values $K = M = 4$ for both HMMs are empirically determined. With these parameters, the HMMD method achieves a 44.0% accuracy while the HMMGD one achieves 54.8%. Though these results are low [86], they show the out-performance of the HMMGD over the HMMD. The rank statistics of order 2 are 71.3% and 82.0% for the HMMD and the HMMGD, respectively. Here again it is clear that the use of the GD model yields higher likelihoods for the correct classes than the Dirichlet one and is thus much more adapted for real-world proportional data modeling. Given the small size of the feature vectors (dimension 10) and the huge gap between the rank statistics of order 1 and 2, HMMGD seems to have the potential to perform accurate classification with a parameters fine tuning and the addition of a well-chosen prior.

This last point is supported by the results of the following experiment: a very simple prior is added over the actions of the data set and combined with the already obtained HMMGD results. For each video sequence, the greatest optical flow magnitude is computed. The prior is then based on the average μ_{OF} and standard deviation σ_{OF} of the optical flow magnitude maximum values of the set of video sequences available for each class (i.e., action type). Its computation is totally data-driven, calculated from the training video sequences available. Assumption is made that, for a given class, this maximal value follows a Gaussian distribution of parameters μ_{OF} and σ_{OF}. As a new video sequence has to be classified, its optical flow maximum magnitude m is computed. The prior is computed as a distance with the following expression:

$$d_{prior} = \left| \mathrm{CDF}(m, \mu_{OF}, \sigma_{OF}) - 0.5 \right| \,, \tag{31}$$

where $\mathrm{CDF}(m, \mu_{OF}, \sigma_{OF})$ denotes the cumulative distribution function of the Gaussian with parameters μ_{OF} and σ_{OF}. The smallest the value, the highest the prior. The classification is obtained combining this prior result with the HMMGD ones.

Therefore, for a new video sequence, the quantity d_{prior} is computed for each class and a first classification result is obtained and stored. Then, a second classification result is obtained from the HMMGD itself. For each class, its rank in the HMMGD and in prior results are added up. The video sequence is then assigned to the class with the lowest score (i.e., the best cumulative rank). This simple prior used alone leads to a classification

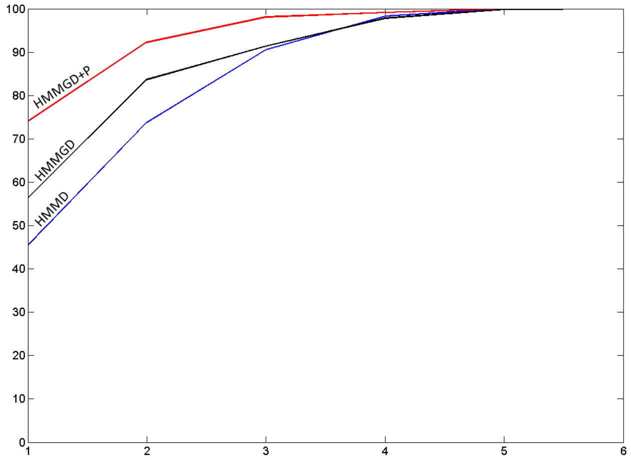


Figure 3.3: Rank statistics of HMMD, HMMGD, and of the combination of HMMGD with a prior

accuracy below 50% however, combined with HMMGD results, the algorithm ends up with a 72.6% accuracy. The rank statistics of order 2 shows an even greater potential as it reaches 91.9%. Better results could be undoubtedly obtained with a more complex prior. However, the study of the best tuning and prior choice is out of the scope of this application that strives at showing the superior performance of HMMGD over HMMD. Figure 3.3 reports the rank statistics for the three studied methods.

3.4.2 Anomaly detection framework

Method overview

I aim at assessing the performance of the proposed models for anomaly detection in public areas. I rely on the features proposed in [52] as they have been specifically designed for this type of application and have shown to be efficient for the data sets I use to lead my main experiments (Section 3.4.3). I adapt these features to my HMM-based framework and apply them to a wider range of situations (see Sections 3.4.3 to 3.4.5). I provide here a brief description of the feature extraction method (see [52] for more details) and explain the modifications I brought to them, in order for them to respect the constraints imposed by the use of the previously presented HMMs.

The preprocessing of the frames consists in a gray-level resampling to the size 160×240 pixels, using a bicubic interpolation, and a noise reduction step performed with a simple

Gaussian filter of size $[3, 3]$ with $\sigma = 1.1$. The resampling allows a faster computation and, for the data sets I used, keeps the moving objects at a size allowing the anomalies detection.

The video sequences are then divided into small volumes, called *cuboids*, each of them being subdivided into 8 subregions, 2 along each direction. A 50% overlap of the cuboids over the spatial directions and no overlap over the temporal dimension in the sampling grid has been shown to be optimal from the computational point of view [52]. Three-dimensional gradient-based features, represented in polar coordinates by a magnitude and two angles, are used. At each pixel of each frame, the quantities

$$\begin{cases} M_{3D} = \sqrt{G_x^2 + G_y^2 + G_t^2}\ , \\ \phi = tan^{-1}(G_t/\sqrt{G_x^2 + G_y^2}),\ \ \phi \in [-\frac{\pi}{2}, \frac{\pi}{2}]\ , \\ \theta = tan^{-1}(G_y/G_x),\ \ \theta \in [-\pi, \pi]\ , \end{cases} \tag{32}$$

are computed, with G_x, G_y, and G_t, being the gradients in Cartesian coordinates. The use of a dense sampling gives rise to too many features and the dimension of the video sequences representation has to be drastically reduced. For each subregion, a 12-bin histogram is built through the quantization of ϕ into 4 values/bins and of θ into 8 values/bins. The contribution of each pixel within a subregion is weighted by its magnitude. [52] concatenates the histograms of the 8 subregions that compose a cuboid to form this cuboid features vector. I propose here to model each cuboid by a series of 8 normalized histograms. This allows to fill several requirements involved by the use of my HMMs:

- The length of the histograms is kept small (12 here), so that the data dimensionality constraint mentioned earlier is respected ;
- The series of histograms illustrate a dynamic mechanism embedded in each cuboid ;
- The normalization ensures the data to be proportional ;
- Each cuboid is observed several times within the same and in different video sequences, ensuring multiple observations, which is preferable for an accurate model estimation and for avoiding model overfitting.

For each cuboid location, an HMM is trained from all available observations. Classification is performed by comparing the likelihood of each cuboid computed from the testing videos to a threshold that depends on the location, i.e., each cuboid location has its own threshold. I define this adaptive threshold using the minimum likelihood value of training samples[3] at each location and multiplying it by a factor k.

Choice of coefficient k

As it will be seen later on, the choice of the value of coefficient k is a key component for the good performance of my approaches. The choice of this threshold should be guided by the following considerations.

Case $k > 1$ This case assumes that some of the anomalous sequences will reach likelihood values greater than some of the non-anomalous training sequences. This can especially occur when the anomalies' scale is the same as or smaller than the surrounding clutter's one (e.g., UCSD data set, Section 3.4.3), or/and when there is a lot of available training sequences with different dynamics. Indeed, varied training sequences also induce the trained HMM to model the variability of the data as something *normal*. Therefore, some anomalies can be interpreted by the model as an illustration of this variability and then, not be detected.

Case $k = 1$ This choice can be made as soon as the *normal* situation is highly repetitive, if the anomalies' scale is notably greater than the normal events' one, and/or if the anomalies dynamics are dramatically different from the clutter dynamics (e.g., the detection of a boat approaching a pier, Section 3.4.4).

Case $k < 1$ This case assumes that some normal sequences will have a likelihood value smaller than what has been seen in the training set. This is especially the case when too few training samples are available to train the model. A reduced number of training samples

[3] In the ideal case, these samples should be part of a validation set, different from the training set, and clear of any anomaly. However, in the case no validation set is available, this value can still be approximated using training samples but keeping in mind that the value obtained for the minimum likelihood is more likely to be higher than what it would have been with an independent validation set. The use of the multiplying coefficient k helps controlling this bias and insures that the Equal Error Rate is reachable.

often leads to overfitting, which will raise high likelihoods only for the samples that are very similar to the training ones.

In a surveillance application, the user mostly wants to get close to the Equal Error Rate (EER) point as it optimizes the ratio between the False Alarm Rate and the Miss Rate. If one can rather easily guess if k should be greater or smaller than one, it seems quite hard to choose its exact value when not equal to 1, as the likelihood levels can vary over an extremely wide range of orders (in my experiments it approximately ranged from 10^{-20} to 10^{100}). Therefore, the case $k = 1$ can be seen as an ideal case and has to be considered as an important characteristic of the quality of the training set with respect to the used type of classifier. In the following applications, I will pay attention to the methods leading to k values close to 1. In other cases, the use of a validation set (anomalous and normal samples different from the training ones) for this threshold's tuning seems unavoidable. This extra attention can drastically improve the model's performance and is therefore absolutely worthy.

3.4.3 Anomaly detection in crowds of pedestrians

In this experiment, I aim at detecting anomalies in the video surveillance sequences of the public UCSD Ped1 and Ped2 data sets [4]. Each of these data sets is composed of video sequences of pedestrians on a walkway and split into a training set, containing normal frames only, and a testing set containing both normal and abnormal frames. The only difference between these two sets is the camera viewpoint. A frame level ground truth is provided for all test sequences. Figures 3.4 and 3.5 show frames from the training sets with different crowd densities and anomalous frames from the Ped1 test set, respectively.

With cuboids of spatial size 40 × 40 pixels, I end up with 77 cuboids, hence 77 HMMs. The likelihoods of the sequences with respect to the corresponding HMMs are computed for the threshold setting. When a new video sequence arrives, the frames are processed by batches of 8 frames (temporal length of the cuboids). The features vector of each cuboid is computed, as well as its likelihood to match with the trained HMM that corresponds to the same location in the training video sequences. In order to quantify the performance

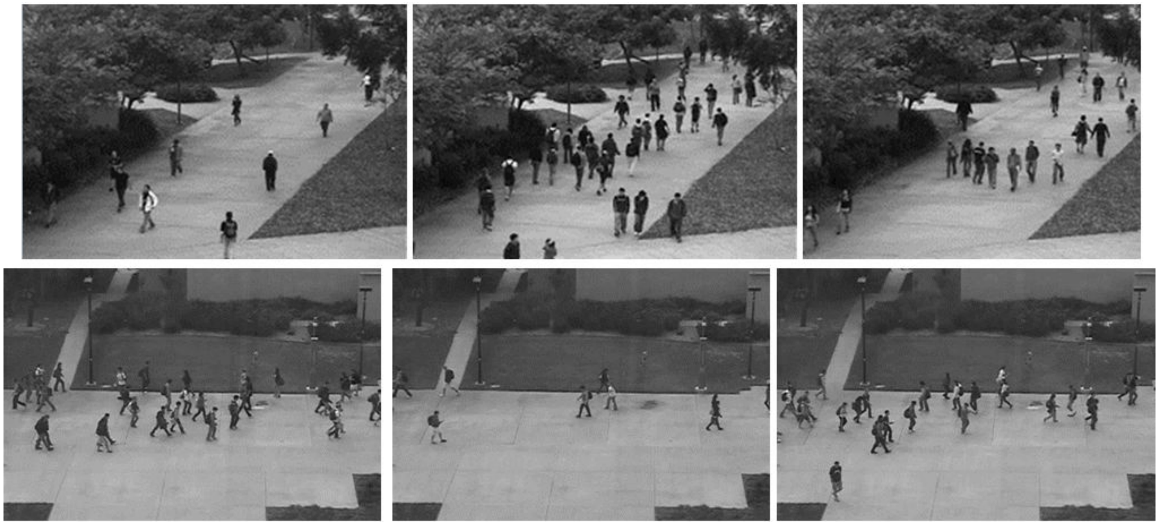


Figure 3.4: Frames from the *UCSDPed1* (upper row) and *UCSDPed2* (bottom row) training sets.

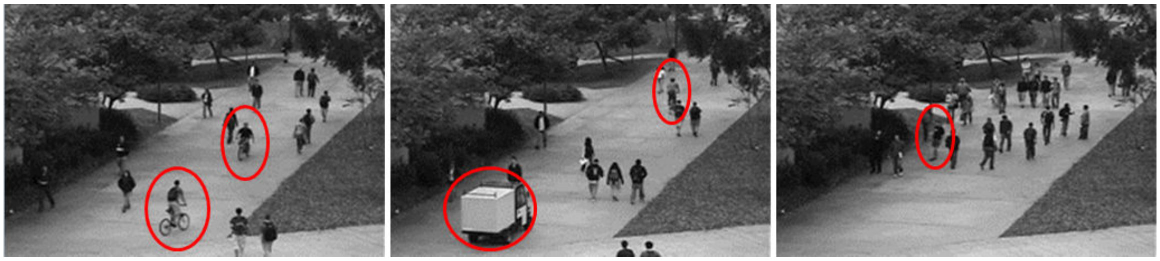


Figure 3.5: Anomalous frames from the *UCSDPed1* data set (anomalies are circled).

of my approaches, I vary the coefficient k to find the EER and to compare it with a few state-of-the-art methods.

The number of states K and mixture components M is determined using a k-means clustering of the training data, with the number of clusters varying from 2 to 20, and then considering the percentage of variance explained, which is expressed as:

$$V_{explained} = 100 \times \frac{d_{total} - d_{within}}{d_{total}}, \tag{33}$$

where d_{total} is the sum of the Euclidean distances of all features vectors to all the clusters centroids and d_{within} the distance of all features vectors to their closest centroid. Figure 3.6 reports the average curve of the normalized percentage of variance explained in function

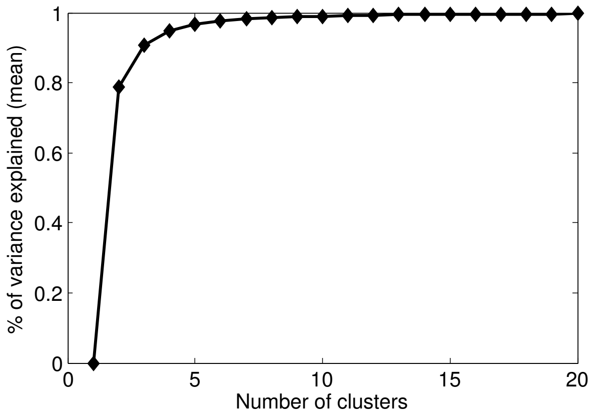


Figure 3.6: Normalized percentage of variance explained in function of the number of clusters (k-means), averaged over all cuboids' locations.

of the number of clusters for all cuboids' locations. Beyond 6 clusters, the percentage of variance explained does not significantly increase, while the addition of clusters makes the model computationally more demanding (as it increases the number of parameters). The number of clusters actually corresponds to the product $K \times M$. I choose to set $K = 2$ and $M = 3$ as the number of states and mixture components, respectively. The setting of K and M can also be done using the Bayesian Information Criterion [87], the Akaike Information Criterion [88], or the Minimum Message Length method [10,60]. However, these methods involve likelihood calculations and are thus computationally more demanding than the simple method that I have applied.

The UCSD data sets provide specific video sequences for training. However, though these sequences are supposed to be reference sequences for normal events, I picked out 3 sequences containing anomalies (bikers) in the Ped1 data set. As it is clear in my framework that the training set has to be clear of any abnormal event, the training sequences numbered 2, 23, and 25 are discarded from my model training.

The HMM-based methods improve the results of [52] while using a unique scale (see Table 3.2). The implementation of a second scale would lead to a significant increase of the computational time which is prohibitive. For the Ped1 data set, the EER does not give any preponderant model within the different HMMs proposed. However, the EER only represents one point of performance, that is seen as optimal in most applications as it

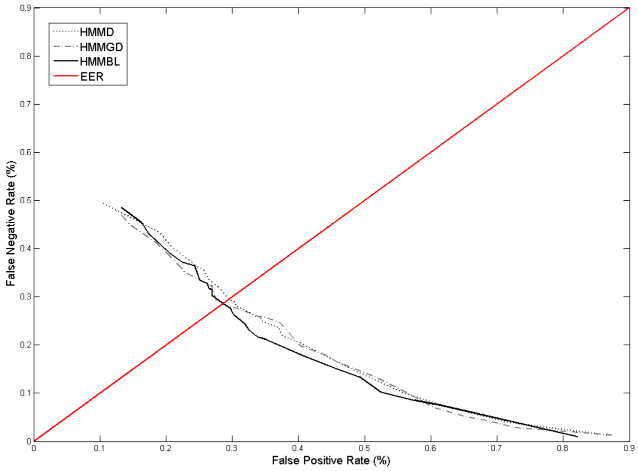


Figure 3.7: Example of False Negative Rate as a function of False Positive Rate for the three studied methods for one run of the experiment on the Ped1 data set. k varying from 1 to 10^{20}.

Method	EER-Ped1	EER-Ped2
[52] Single scale	34.0%	32.0%
[52] 2 scales + context	31.0%	30.0%
[89]	31.0%	42.0%
[22]	32.4%	28.5%
[59]	27.0%	26.9%
[54]	17.8%	18.5%
HMMD	28.9%	**18.5%**
HMMGD	29.0%	22.0%
HMMBL	29.0%	**16.6%**

Table 3.2: Equal Error Rate for the detection task on *UCSDPed1* and *UCSDPed2* data sets. Values in bold are the ones for which the EER is reached for a coefficient k value close to 1.

minimizes the two error types simultaneously. In order to discriminate the best model, I look at the overall performance of these three methods computing the Area Under the Curve (AUC). In this case, the training set is of significant size and contains large disparities at the crowd density level, therefore the coefficient k is kept larger than 1, in the range $[1, 10^{20}]$. The EER point is reached around $k = 10^6$. I compute the AUC over a range of k values varying from 1 to 10^{20} and calculate the area under the obtained curve using a trapezoidal numerical integration approximation. The AUC of the HMMD and HMMGD methods are respectively 8% and 12% larger than the HMMBL one. This means that out of the EER point, HMMBL will typically bring lower error rates than the two other models. This can be observed on Figure 3.7 which reports the performance of one run of the experiment on the Ped1 data set.

For the Ped2 data set, the HMMBL clearly works better than the other models. Moreover, it reaches the EER for a coefficient k close to 1. HMMD also gets to the EER point for k close to 1, whereas HMMGD models reach it for values much smaller than 1 (around 10^{-4}), which could be expected as the training set is much smaller than the Ped1 one. Ped2 testing set is however completely unbalanced in terms of normal/abnormal frames ratio: 84.2% of the frames contain an anomaly. The Ped1 testing set is much more balanced with 56.2% of abnormal frames. In the case of an unbalanced data set with only two categories involved, a fairer measure of the performance of a model is the Matthews correlation coefficient (MCC), that takes into account the four terms of the confusion matrix. An MCC equal to 1 means a perfect correlation between the ground truth and the results, a result close to 0 means the classifier is not better than randomness, and a coefficient close to -1 means a total opposition between the ground truth and the results. Over the range of thresholds specified earlier, the average maximal MCC reached is 0.55 for HMMD, 0.50 for HMMGD, and 0.62 for HMMBL, which confirms the better performance of the HMMBL models.

Influence of the crowd density

The models are trained from video sequences representing different crowd densities. However, once trained, it is interesting to look at the performance of my models depending on the crowd density of an unknown video sequence. For this experiment, I worked only with the Ped1 testing set which is substantially larger than the Ped2 one (only 12 video sequences). I split up the Ped1 data set into 2 subsets of approximately the same size. The first subset, denoted LD, gathers 19 video sequences featuring *low density* crowds and the second one, denoted HD, gathers 17 video sequences featuring *high density* crowds. Table 3.3 reports the EER for this two subsets.

Though trained from various crowd densities, it is clear than all models perform better when the crowd density is low, which confirms the intuition that anomalies are easier to

Subset	Ped1-LD	Ped1-HD
HMMD	25.4%	36.2%
HMMGD	25.6%	33.4%
HMMBL	24.7%	35.8%

Table 3.3: Equal Error Rate for the detection task on *UCSDPed1* LD and HD subsets.

detect in less crowded environments. A suggested improvement could be the addition of a crowd density estimator and the training of several models (for different densities). This possibility is not examined in the present work and left for future investigation.

Localization task

Over the 36 and 12 test sequences available in the Ped1 and Ped2 data sets, respectively, 10 (Ped1) and 9 (Ped2) are precisely annotated in order to evaluate the precision of the anomaly localization[4]. For each frame in which an anomaly has been detected, I first look at the presence or absence of a true anomaly and, in the case there is one, I look at the percentage of well-detected abnormal pixels. If this percentage is higher than 40%, the anomaly is considered as well-localized. The three proposed methods show quite similar results and perform much better than most of the state-of-the-art methods. The localization results are obtained by using the settings leading to the EER. Table 3.4 provides a comparison of different methods ([22] does not provide any result for the localization task).

Discussion

Beyond the direct comparison that can be done with [52] as only slightly modified features have been used, I also compare my approach with other state-of-the-art methods. [89] proposed a *social force* model based on the study of the interaction between multiple moving particles. Frames are represented using optic flow and the normal behaviors are modeled with a bag-of-words approach. The results over the UCSD data sets are taken from the results presented in [52]. Recently, a probabilistic approach for the detection and localization

[4] A full annotation for missing ground truth has been added later on in [3] but has not been used in these experiments. The results used for comparison in Table 3.4 are the ones corresponding to the same partially annotated ground truth. In Chapter 4 however, the fully annotated data set is used.

<table>
<tr><th>Method</th><th>Ped1</th><th>Ped2</th></tr>
<tr><td>[52] Single scale</td><td colspan="2">27.0%</td></tr>
<tr><td>[52] 2 scales and context</td><td colspan="2">29.0%</td></tr>
<tr><td>[89]</td><td colspan="2">21.0%</td></tr>
<tr><td>[59]</td><td>78.9%</td><td>74.9%</td></tr>
<tr><td>[54]</td><td>64.8%</td><td>70.1%</td></tr>
<tr><td>HMMD</td><td>52%</td><td>77.2%</td></tr>
<tr><td>HMMGD</td><td>51%</td><td>74.7%</td></tr>
<tr><td>HMMBL</td><td>52%</td><td>73.5%</td></tr>
</table>

Table 3.4: Good detection rate at pixel-level (or true localization rate) on *UCSDPed1* and *UCSDPed2* data sets.

of anomalies in which normal behaviors are modeled by the use of mixtures of dynamic textures has been proposed [54]. Several spatial scales are used within a hierarchical framework. Spatio-temporal features based on a texture map and 3D Harris functions have been implemented in a Gaussian-based HMM in [22]. Finally, the work of [59] shows the efficiency of features based on a combination of histograms of oriented swarms (for the dynamics) with histograms of oriented gradients (for the appearance). The anomaly detection is then performed within a Support Vector Machine (SVM) framework.

My models perform better than both [52] and [89] in all cases. On the Ped2 data set, for the detection task, the HMMBL model outperforms all other approaches. However, [54] achieves better detection performance on the Ped1 data set. For the localization, the three proposed methods achieve the best detection rate around 75% for the Ped2 data set along with the method of [59] and, the best score for the Ped1 data set are the ones of [54] and [59]. This lower performance has however to be interpreted considering the computing time required for a frame. The three presented methods can process a frame within 0.2 seconds on a computer with 5GB RAM and a 3.4GHz CPU, working with Matlab. The video sequences of the UCSD data sets have been recorded at a rate of 10 frames per seconds which is a usual rate for video surveillance designs. The proposed methods seems thus faster than the ones presented in [54] and [59] that both takes around 1.2 second per frame, with a 2.8 GHz CPU, 2GB RAM and `C` programming, and with a 3.5 GHz CPU, 16 GHz RAM, and `C++` programming, respectively. Since the training step is executed offline, my approaches are not far from running in real-time and might be able to do so with an optimized coding and some operations realized in parallel (especially in the features extraction step). Overall, the three models work fine for this type of applications with a slight advantage for the BL-based HMM. As a side note, more recent publications results over these data sets are presented in the next chapter of this dissertation, in Section 4.4.

3.4.4 Boat detection

As it is, my approach is applicable to any type of video sequence recorded with a still camera if training data, corresponding to what is considered as the *normal* situation, are available. To assess this point, I used the *Boat-Sea* video sequence from the Anomalous Behaviour Data Set [56]. In this video, a still capture of an empty pier is recorded when a boat arrives from the left side of the camera field. The aim is to detect, localize, and track this event. This video sequence does not represent a big challenge for the detection task itself as the environment is uncluttered and as there is no camera recording issue (i.e., jitter, white stripes,...). However, it is interesting to look at the anomaly localization performance of the proposed models.

The HMMs are trained using the first part of the video sequence, i.e., until the boat appears on the left side of the image, while the rest of the sequence is used as the testing set. The threshold k is set to 1, according to the analysis provided in Section 3.4.2. All three models fully detect and track the boat, however, the HMMBL method involves less false positive cuboids at other spots within a frame. Figure 3.8 reports the number of cuboids featuring the boat as well as the number of abnormal cuboids detected by each of the three studied models for every temporal unit (i.e., 8 frames).

The fact that all anomalous cuboids are not detected does not mean the boat is not detected because of the cuboids' spatial overlap. Table 3.5 reports the average number of truly abnormal cuboids and those that are detected as abnormal cuboids by my models, respectively. Figure 3.9 shows examples of the boat detection and precise localization for the HMMBL model.

All three approaches succeed in detecting, localizing and tracking the boat whether it

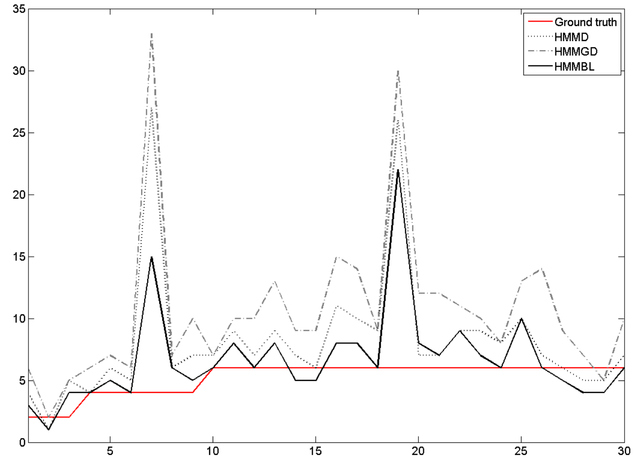


Figure 3.8: Number of cuboid detected as abnormal by the three models in comparison with the true number of abnormal cuboids by temporal unit of 8 frames.

Method	Average
Ground truth	5.2
HMMD	8.2
HMMGD	10.6
HMMBL	6.7

Table 3.5: Average number of cuboids detected as abnormal in the *Boat-Sea* video sequence compared to the ground truth.

is moving or still as it is in the last part of the video sequence. This last point is due to the dynamic environment as my features are entirely movement-based. Even when still, the presence of the boat disturbs the pattern of the sea (waves) and the stillness of the image in this area raises a low model likelihood result, which allows the detection of the event (presence of the boat).

3.4.5 Detecting anomalies at a security check point

The last proposed application aims at detecting people going in the wrong direction in an airport security line-up. The video sequences, extracted from the Anomalous Behavior Data Set [56], are recorded from a surveillance camera hung up to the ceiling and filming vertically downwards. The first sequence, of about 200 frames, is clear of any anomaly and thus used for the training step, while the rest of the sequences composes the testing set.

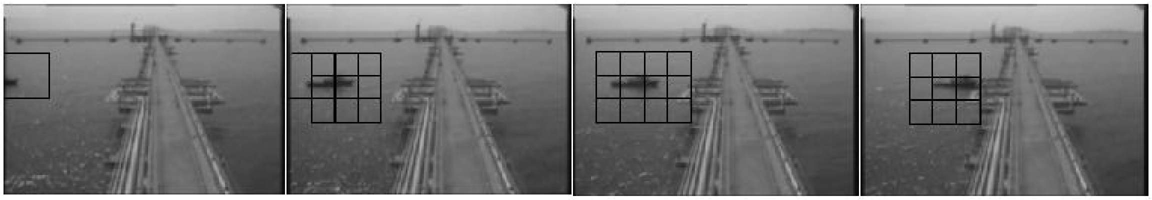


Figure 3.9: Frames from the *Sea-boat* video sequence with superposition of the cuboids detected as anomalous by the HMMBL model.

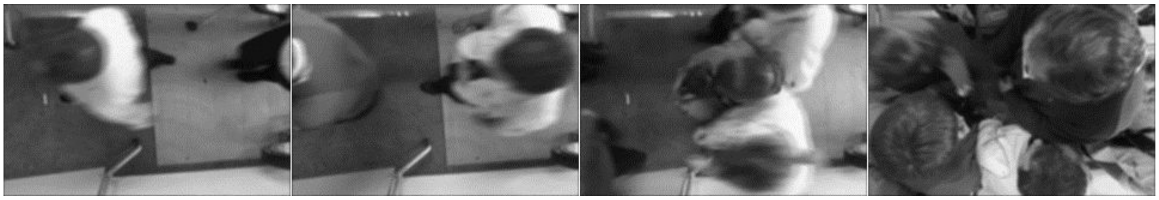


Figure 3.10: Sample frames from the *Airport-WrongDir* video sequences. From left to right: two normal situations (people walking from the right to the left), a woman going the wrong way, people stuck on their way. The two last frames are considered as anomalous.

Figure 3.10 shows a few frames from the data set.

From Figure 3.10, one can notice that the anomalies are of larger scale than in the two previous applications. It seems thus logical to increase the size of the cuboids in order to limit the number of false positive cuboids. I propose to use cuboids that are twice and four times larger than the previously used ones, i.e., size 80 × 80 and 160 × 160 pixels. Each frame is then entirely covered by 15 cuboids and 2 cuboids, respectively. The results are reported in Table 3.6

The best performance in this case is achieved by the HMMGD model either with (k values far from 1) or without tuning coefficient k. However, the performance of the three methods dramatically varies depending on the scale of the cuboids. The use of cuboids of

Scale	HMMD	HMMGD	HMMBL
40 × 40	18.6	13.9	24.8
80 × 80	23.0	15.8	**45.7**
160 × 160	40.9	**28.6**	53.8

Table 3.6: EER over the *Airport-WrongDir* data set depending on the cuboids size and the model used. Results in bold are the ones for which the EER is reached for a coefficient k close to 1.

size 40×40 results in a large proportion of false positive that asks for a coefficient k several orders below 1 in order to reach the EER. Cuboids of scale 80×80 suffer this issue to a smaller extent for the HHMD and HMMGD models. The HMMBL allows to reach the EER for a k value close to 1 though the overall EER is high (compared to what can be achieved with a precise tuning of k). Finally, contrary to smaller cuboids, cuboids of size 160×160 lead to a significant amount of missed anomalies for the HMMBL model and, to a smaller extent, for the HMMD model. However, the behavior of the HMMGD model without tuning k is the best obtained for this application. In this latter case, the EER is reached for k close to 1 and has a higher value than the one obtained with smaller cuboids along with a precise tuning of k yet is still acceptable. The processing time is greatly reduced because of the smaller number of cuboids, hence the number of HMMs to estimate.

3.5 Conclusion

In this chapter, two new variants of the HMM, based on the generalized Dirichlet and the Beta-Liouville distributions have been proposed and their learning equations via an Expectation-Maximization procedure derived. Moreover, it provides the first results for classification and anomaly detection of the Dirichlet-based HMM, only tested over synthetic data before this work. The preliminary study confirmed the expectations and corroborated the multiple studies that already proved that, in the case of mixture models and for proportional data, the use of Dirichlet distributions provides a certain advantage. The two proposed variants of the HMM allowed the improvement of the anomaly detection and localization performance compared with a simpler variant based on Dirichlet mixtures. This provides new alternatives for proportional data modeling that arise in numerous situations when compact data representation is needed. Based on the experimental results presented in this chapter, these new models have the potential to enhance the results obtained with the Dirichlet-based HMMs in a wide range of applications, while demanding the tuning of few extra parameters in the case of the generalized Dirichlet version and only one extra parameter for the HMMBL. Furthermore, as the tuning of the HMM is usually done offline, this extra parameters' tuning does not have a significant impact on the model efficiency. These new approaches allow the modeling of more general data as there are fewer restrictions on the data covariance structure, and as they can efficiently work with data of slightly higher dimension than the Dirichlet-based HMM can.

The derivations provided for these two new models have been presented in such a way that they can be easily re-used to derive HMM variants with other exponential probability distributions. In this way, it should be possible to derive and use HMMs with probability distributions that closely match the properties of the data that one need to model. I also determined some easily applicable guidelines for the choice of the threshold to be used for anomaly discrimination. I finally showed that the proposed methods could detect and localize anomalies in a near real-time fashion without any code optimization, in a Matlab implementation. The most time-consuming step is the computation of the features and the use of features that are faster to compute would also help to reach a real-time processing.

Chapter 4

Variational learning of HMMs for proportional data

4.1 Introduction & Related Work

This chapter is in essence similar to the previous one but presents an alternative way for reaching the same goal, and does it with a better performance. As already mentioned, everyday, we are in the scope of multiple video surveillance cameras, when using public transportation, parking lots, walking nearby governmental buildings, or simply buying groceries. The rapid growth of these recording systems has been mostly driven by global public security concerns with respect to thieves, personal attacks, or terrorism, along with an always decreasing cost of hardware with a steady increase of the embedded features quality (greater resolution, storage capacity,...) [58, 90]. The realm of data that can be extracted from these systems and the goals that can be achieved through their smart use is huge and various. They are today installed and used by governments, commercial companies, and private citizens, all with different preoccupations and requirements [91]. Data processing methods specifically adapted to these systems are required to answer these various challenges.

I only briefly recall hereafter the different types of approaches that one can find in the literature as details have been exposed in the previous chapter in Section 3.2. In a few words, early approaches were mostly based on tracking that becomes very challenging when many entities are present at the same time in the camera field, especially due to occlusions. Furthermore, tracking methods, when used alone, involve the loss of all appearance-related information, and reveal themselves unable to discriminate between a car and a pedestrian if both follow the same trajectory at the same speed for example. Most of these tracking methods rely on the distinction between the background and moving foreground objects, achieved through background subtraction methods which become somewhat unreliable when working with a dynamic backgrounds.

Methods that avoid the drawbacks of the tracking and background subtraction tasks by focusing on higher information levels such as optical flow, gradients-based quantities, or dynamic textures, and on a global understanding of the situation, are rising increasing interest. Finally, some recent methods aim at modeling the relations between spatio-temporal points of interest (STIPs) through graph representations [92, 93]. The most recent approaches leading to the current state-of-the-art results are presented in more detail in Section 4.4.

Finally, in order to detect when a normal situation becomes abnormal, a binary or two-class classifier is needed. As exposed in Chapter 3, the discriminative power of the classifier is as important as having appropriate features and HMMs can be a very discriminative classifier when used with the right assumptions. As mentioned ealier in this dissertation, the recent years have seen their learning equations adapted to a variety of continuous non-Gaussian data types (e.g., mixed [75], proportional [28], normal inverse Gaussian [78], Student's t [77] data) for application in numerous different fields.

[28] introduced the equations for learning an HMM based on mixtures of Dirichlet emissions. The main modification compared to the typical Gaussian-based model occurs in the M-step of the Expectation-Maximization (EM) algorithm, as it is in this step that the emission distribution parameters are updated. In Chapter 3, the EM algorithm has been derived for HMM based on mixtures of generalized Dirichlet. Estimating all the parameters using an EM algorithm is the typical approached, the most widely used. This estimation process is based on the computation of the likelihood of the available observation time-series with respect to the model. This quantity is unfortunately computationally intractable

as it involves a summation over all possible combinations of hidden states and mixture components. The maximum likelihood approach, which aims at maximizing it, can lead to overfitting and convergence towards a local maximum given the multimodality of the likelihood function. Bayesian methods such as MCMC provide a way to approximate the likelihood but involve extremely long computations that are prohibitive if many models have to be learnt. More recently, variational Bayesian approaches have been successfully proposed as a computationally tractable way of tackling this approximation. It has been first studied in [94, 95] for discrete data, and then in [96] for Gaussian data, and in [97] for Student's-t data. Variational learning has also been applied to other machine learning algorithms such as probability mixture models [11, 98, 99] for various types of distributions, neural networks [100], and graphical models [101].

When estimated with the Baum-Welch algorithm, the HMMs parameters are considered as unknown but fixed values. The estimation starts from the initial guess which is iteratively refined via E- and M-steps. At each iteration, the expected complete data likelihood is maximized with a guarantee to converge. Opposite, in Bayesian frameworks, all parameters are considered as random variables. A prior distribution is chosen over each parameter and the posterior distribution is inferred using Bayes' rule. The marginal likelihood of the data (or evidence) is obtained by integrating out the parameters as expressed later in Equation (36). This way, I define a *family* of models associated to a set of probability scores. The computation of this marginal likelihood is however computationally intractable. Some approximation methods exist but are either computationally expensive (e.g., MCMC methods) or give poor results [96]. The variational Bayesian framework has proved to be able to approximate the quantity accurately while being computationally tractable. Similar to the Baum-Welch algorithm, it is an iterative technique alternating E- and M-steps and guaranteed to converge. Compared to the variational Bayesian learning of HMMs for discrete, Gaussian, or Student's-t data, the assumption that the data follow mixtures of Dirichlet distributions implies the introduction of an extra approximation over the Dirichlet conjugate prior. It is therefore interesting to study how the increased flexibility of the variational approach and the introduction of this approximation will balance in the overall performance of the algorithm, with respect to the Maximum Likelihood approach.

Indeed, the variational Bayesian learning of the Dirichlet parameters is not as straightforward as its conjugate prior distribution exists but is computationally intractable. [11] showed that an approximate form could however be used with promising results for finite Dirichlet mixtures modeling. Basing my work on this latter reference as well as other previous works on variational estimation, I propose here to derive the equations of the variational Bayesian learning of the Dirichlet-based HMM (VBHMMD). This is the first contribution presented in this chapter.

Although the Dirichlet models show good modeling capabilities for proportional data, they rely on the assumption that the data have negative covariance [60]. To relax this assumption and reach higher accuracy, generalized Dirichlet (GD) models can be used. These distributions are both part of the exponential family and the Dirichlet is also a GD distribution [61]. GD models have also been studied in some machine learning applications, and have been shown to be in most cases more effective than the Dirichlet-based ones. For example, it has been used for document topic representation in a Latent Dirichlet Allocation framework [79], for texture classification [60] and web service intrusion detection [99] with finite mixture models, and for the design of generative kernels for Support Vector Machine [80]. The latter has shown enhanced results compared to the use of the Dirichlet distribution for object recognition and content-based image classification.

From these observations, I also propose an extension of the VBHMMD model towards a variational Bayesian generalized Dirichlet-based HMM (VBHMMGD). The extension of the HMM to the generalized Dirichlet case will unravel a double advantage. Indeed, on top of having a more flexible covariance structure, the use of this distribution, via the projection of the data into a specific space, relaxes an assumption made on the independence of the Dirichlet parameters when choosing the approximate conjugate prior for the Dirichlet model. This makes this model of high interest when working with compositional data. To the best of my knowledge, no work has developed so far the variational learning of the Dirichlet and GD-based HMMs. My approach is validated in a crowd anomaly detection application using the Pedestrians UCSD data sets, giving among the best results to date in a near real-time fashion. Furthermore, I show that my approach perfectly handles generated tampering events on one of these data sets. Finally, applying my method to a tampering detection data set from the Visual Analysis of People [102] with a moving camera allowed me to leverage its strength and weaknesses.

The contributions presented in this chapter can be summarized as follows:

- The complete derivation of the equations for the variational Bayesian learning of the Dirichlet-based HMM.

- The extension of the variational model to the case of the generalized Dirichlet.

- The application of these new models to the surveillance of public areas under real conditions and the direct comparison to the EM-based learning.

- The analysis of the strength and weaknesses of the proposed approach and their probable causes, giving meaningful hints for future directions of improvement.

The work presented in this chapter is currently under review for publication in IEEE Transactions on Neural Networks and Learning Systems under the title *Variational Bayesian learning of generalized Dirichlet-based hidden Markov models*.

In what follows, Section 4.2 derives the variational Bayesian learning of the Dirichlet-based HMM and Section 4.3 extends the model to the generalized Dirichlet case. Experiments over real-world data sets are carried out in Section 4.4 and 4.5 and conclusions are drawn in Section 4.6.

4.2 Dirichlet-based variational Bayesian HMM

In this section, I derive the equations of the variational Bayesian learning of the Dirichlet-based HMM. The main difference between the variational approach and the typical Baum-Welch approach as introduced in the previous chapter is the fact that in the variational approach, all the HMM parameters (i.e., transition matrix, mixing matrix, and initial state vector coefficients, as well as the parameters of the emission distributions) are considered as being random variables. Therefore, when in the Baum-Welch approach the HMM parameters are initialized and iteratively updated, in the variational approach the assumptions are made over the prior distribution that the different HMM parameters follow. The parameters of these prior distributions (called hyperparameters) are initialized and then iteratively updated. The values of the HMM parameters are inferred from the estimated values of these hyperparameters.

I first recall the general expression of the D-dimensional Dirichlet distribution with parameters $(\alpha_1, ..., \alpha_D)$,

$$\mathcal{D}(X|\alpha_1, ..., \alpha_D) = \frac{\Gamma(\sum_{d=1}^{D} \alpha_d)}{\prod_{d=1}^{D} \Gamma(\alpha_d)} \prod_{d=1}^{D} x_d^{\alpha_d - 1} \; . \tag{34}$$

with, $\alpha_i > 0$, $x_i > 0$, $i \in [1, D]$, $\sum_{i=1}^{D} x_i = 1$, and $\Gamma(t) = \int_0^\infty x^{t-1} e^{-x} dx$, the Gamma function.

Following the general description of the HMMs provided in Chapter 1, the likelihood of a sequence of observations (or time-series) X given the model is typically expressed as:

$$p(X|A, C, \pi, \alpha) = \sum_{S} \sum_{L} \pi_{s_1} \left[\prod_{t=2}^{T} a_{s_{t-1}, s_t} \right] \left[\prod_{t=1}^{T} c_{s_t, m_t} p(x_t | \alpha_{s_t, m_t}) \right] , \tag{35}$$

where S is the set of hidden states, L the set of mixtures' components. $\alpha_{ij} = (\alpha_{1ij}, \ldots, \alpha_{Dij})$, with $i \in [1, K]$ and $j \in [1, M]$, where K stands for the number of states and M the number of mixture components (which is assumed to be the same for each state without loss of generality). For clarity's sake, I derive the model for a unique observation sequence. When more observations sequences are available (which is highly recommended to prevent overfitting), a summation over the observation sequences has to be logically added in all the equations involving the data sequences.

The exact computation of the likelihood is intractable as it involves the summation over all possible combinations of states and mixtures components. The typical approach consists in the maximization of the likelihood of the data with respect to the model, as done with the Baum-Welch algorithm [26]. However, this procedure has drawbacks such as overfitting and is not guaranteed to converge towards the global maximum as the likelihood function is in general multimodal.

The variational Bayesian approach to the model estimation problem uses the posterior probability by assigning a prior to the parameters, and integrating it out to compute the marginal likelihood of the data. All the model's parameters are seen as random variables. Using the complete data likelihood, it is expressed as

$$p(X) = \int d\pi dAdCd\alpha \sum_{S,L} p(A,C,\pi,\alpha)p(X,S,L|A,C,\pi,\alpha) \; . \tag{36}$$

The exact computation of this quantity is still intractable but a lower bound can be derived by introducing an approximate distribution $q(A,C,\pi,\alpha,S,L)$ of the true posterior $p(A,C,\pi,\alpha,S,L|X)$. Using Equation (36), along with the Jensen's inequality I obtain

$$\begin{aligned} \ln(p(X)) = & \ln\left\{ \int dAdCd\pi d\alpha \sum_{S,L} p(A,C,\pi,\alpha)p(X,S,L|A,C,\pi,\alpha) \right\} \\ & \geq \int d\pi dAdCd\alpha \sum_{S,L} q(A,C,\pi,\alpha,S,L) \ln\left\{ \frac{p(A,C,\pi,\alpha)p(X,S,L|A,C,\pi,\alpha)}{q(A,C,\pi,\alpha,S,L)} \right\} \; . \end{aligned} \tag{37}$$

The inequality is tight when q equals the true posterior. Denoting the lower bound $\mathcal{L}(q)$, one can easily find that

$$\ln(p(X)) = \mathcal{L}(q) - \mathrm{KL}(q(A,C,\pi,\alpha,S,L)||p(A,C,\pi,\alpha,S,L|X)) \; , \tag{38}$$

where KL is the Kullback-Leibler distance between the true posterior and its approximate distribution [11].

As the true posterior distribution is computationally intractable, I consider a restricted family of distributions. Following the assumption made in [11,94–97], I consider that q can be written in a factorized form, i.e., $q(A,C,\pi,\alpha,S,L) = q(A)q(C)q(\pi)q(\alpha)q(S,L)$, and this holds for p too. Then, the lower bound can be written as

$$\begin{aligned}\ln(p(X)) \geq \sum_{S,L} \int dA dC d\pi d\alpha q(\pi) q(A) q(C) q(\alpha) q(S,L) \Big\{ &\ln(p(\pi)) + \ln(p(A)) \\ + \ln(p(C)) + \ln(p(\alpha)) + \ln(\pi_{s_1}) + \sum_{t=2}^{T} \ln(a_{s_{t-1},s_t}) + \sum_{t=1}^{T} \ln(c_{s_t,m_t}) \\ + \sum_{t=1}^{T} \ln(p(x_t|\alpha_{s_t,m_t})) - \ln(q(S,L)) - \ln(q(\pi)) - \ln(q(A)) - \ln(q(C)) - \ln(q(\alpha)) \Big\} \\ = F(q(\pi)) + F(q(C)) + F(q(A)) + F(q(\alpha)) + F(q(S,L)) \ . \end{aligned} \tag{39}$$

This lower bound is not convex and there will in general exist multiple maxima, which implies a dependence of obtained solution on the initialization.

The priors of all the parameters have to be defined in order to evaluate Equation (39). A natural choice for the prior of parameters A, C, and π is the Dirichlet distribution. Indeed, all the coefficients of these matrices and vector are strictly positive, less than 1, with each row summing up to one.

I therefore logically define

$$\begin{aligned} p(\pi) &= \mathcal{D}(\pi|\phi^{\pi}) = \mathcal{D}(\pi_1, ..., \pi_K|\phi_1^{\pi}, ..., \phi_K^{\pi}) \ , \\ p(A) &= \prod_{i=1}^{K} \mathcal{D}(a_{i_1}, ..., a_{i_K}|\phi_{i_1}^{A}, ..., \phi_{i_K}^{A}) \ , \\ p(C) &= \prod_{i=1}^{M} \mathcal{D}(c_{i_1}, ..., c_{i_M}|\phi_{i_1}^{C}, ..., \phi_{i_M}^{C}) \ . \end{aligned} \tag{40}$$

A conjugate prior has also to be defined over the Dirichlet parameters α. As for any distribution belonging to the exponential family, it can be expressed as [11,103,104]:

$$p(\alpha) = f(\nu, \mu) \left[\frac{\Gamma(\sum_{l=1}^{D} \alpha_l)}{\prod_{l=1}^{D} \Gamma(\alpha_l)} \right]^{\nu} \prod_{l=1}^{D} e^{-\mu_l(\alpha_l - 1)} \ , \tag{41}$$

where $f(\nu, \mu)$ is a normalization coefficient and (ν, μ) are hyperparameters. However, due to the difficulty to evaluate the normalization coefficient, this exact prior is intractable and an approximation has to be used in order to carry out the variational inference. Following the proposition of [98] and [11], a conjugate prior for the Beta distribution is used along with the assumption that the Dirichlet parameters are statistically independent. The Beta distribution is the special case of the unidimensional Dirichlet distribution, which makes this approximation meaningful. A good approximation to the conjugate prior of the Beta distribution is the Gamma distribution $\mathcal{G}$ expressed as follows

$$p(\alpha_{ijl}) = \mathcal{G}(\alpha_{ijl}|u_{ijl}, v_{ijl}) = \frac{v_{ijl}^{u_{ijl}}}{\Gamma(u_{ijl})} \alpha_{ijl}^{u_{ijl}-1} e^{-v_{ijl}\alpha_{ijl}} \ , \tag{42}$$

with $l \in [1, D]$, $i \in [1, K]$ and $j \in [1, M]$. The hyperparameters u and v are strictly positive.

$$\begin{aligned} p(\{\vec{\alpha_{ij}}\}_{i,j=1}^{K,M}) &= \prod_{l=1}^{D} \prod_{i=1}^{K} \prod_{j=1}^{M} \mathcal{G}(\alpha_{ijl}|u_{ijl}, v_{ijl}) \\ &= \prod_{l=1}^{D} \prod_{i=1}^{K} \prod_{j=1}^{M} \frac{v_{ijl}^{u_{ijl}}}{\Gamma(u_{ijl})} \alpha_{ijl}^{u_{ijl}-1} e^{-v_{ijl}\alpha_{ijl}} \ . \end{aligned} \tag{43}$$

The variational inference consists in iteratively alternating two steps, namely the M-step and the E-step. In the M-step, I consider the sequence of hidden states and mixture components to be fixed. Therefore, the terms in Equation (39) that are function of (S, L) are ignored in the following equations.

I first study the optimization of $q(A)$, $q(C)$, and $q(\pi)$. This specific part of the optimization, as independent from the emission distributions used, is common to other continuous HMM and has therefore already been studied in [96] and [97]. For keeping the theory clear, I only give the main equations and refer the reader to the aforementioned references for more details. Gathering all quantities related to A in Equation (39), I obtain

$$F(q(A)) = \int dA q(A) \ln \left[\frac{\prod_{i=1}^{K} \prod_{j=1}^{K} a_{ij}^{w_{ij}^{A}-1}}{q(A)} \right] , \tag{44}$$

with

$$w_{ij}^{A} = \sum_{t=2}^{T} \gamma_{ijt}^{A} + \phi_{ij}^{A} \ , \tag{45}$$

and

$$\gamma^A_{ijt} \triangleq q(s_{t-1} = i, s_t = j) \ . \tag{46}$$

The latter quantity is a local probability, typically computed in the HMM framework using a Forward-Backward algorithm [26]. Gibbs inequality leads to the following expression of $q(A)$ that maximizes $F(q(A))$:

$$q(A) = \prod_{i=1}^{K} \mathcal{D}(a_{i1}, \ldots, a_{iK} | w^A_{i1}, \ldots, w^A_{iK}) \ . \tag{47}$$

In the same fashion, for π, I have

$$q(\pi) = \mathcal{D}(\pi_1, \ldots, pi_K | w^\pi_1, \ldots, w^\pi_K) \ , \tag{48}$$

with

$$w^\pi_i = \gamma^\pi_i + \phi^\pi_i \ , \tag{49}$$

and

$$\gamma^\pi_i \triangleq q(s_1 = i) \ . \tag{50}$$

And for C:

$$q(C) = \prod_{i=1}^{K} \mathcal{D}(c_{i1}, ..., c_{iM} | w^C_{i1}, ..., w^C_{iM}) \ , \tag{51}$$

with

$$w^C_{ij} = \sum_{t=1}^{T} \gamma^C_{ijt} + \phi^C_{ij} \ , \tag{52}$$

and

$$\gamma^C_{ijt} \triangleq q(s_t = i, m_t = j) \ . \tag{53}$$

I now aim at optimizing $F(q(\alpha))$, making use of the approximation presented in Equation (42). From Equation (39), I have

$$F(q(\alpha)) = \int d\alpha q(\alpha) \ln \left\{ \frac{\prod_{i=1}^{K} \prod_{j=1}^{M} p(\alpha_{ij}) \prod_{t=1}^{T} p(x_t | \alpha_{ij})^{\gamma^C_{ijt}}}{q(\alpha)} \right\} . \tag{54}$$

Using Equation (43), the log-evidence maximization is given by

$$q(\alpha) = \prod_{i=1}^{K} \prod_{j=1}^{M} q(\alpha_{ij}) \ , \tag{55}$$

with,

$$q(\alpha_{ij}) = \prod_{l=1}^{D} \mathcal{G}(\alpha_{ijl} | u^{\star}_{ijl}, v^{\star}_{ijl}) \ . \tag{56}$$

Further, quantities marked with a $\star$ superscript refer to the optimized parameters. To this point, the problem is equivalent to the one of finding the variational solution of the parameters of a finite Dirichlet mixture model. The estimation of this finite mixture has been studied in [11] (precisely in Section B of Appendix A) and yields to the following solutions for the hyperparameters u and v.

$$u^{\star}_{ijl} = u_{ijl} + \mathcal{U}_{ijl} \ , \tag{57}$$

with, for i and j fixed and for P observation vectors

$$\mathcal{U}_{ijl} = \sum_{p=1}^{P} \langle Z_{pij} \rangle \bar{\alpha}_{ijl} \left[\Psi\left(\sum_{d=1}^{D} \bar{\alpha}_{ijd} \right) - \Psi(\bar{\alpha}_{ijl}) + \sum_{d=1, d \neq l}^{D} \Psi'\left(\sum_{d=1}^{D} \bar{\alpha}_{ijd} \right) \bar{\alpha}_{ijd} (\langle \ln(\alpha_{ijd}) \rangle - \ln(\bar{\alpha}_{ijd})) \right] . \tag{58}$$

and

$$v^{\star}_{ijl} = v_{ijl} - \mathcal{V}_{ijl} \ , \tag{59}$$

with

$$\mathcal{V}_{ijl} = \sum_{p=1}^{P} \langle Z_{pij} \rangle \ln(X_{pl}) \ . \tag{60}$$

The responsibilities (or weight of the data samples with respect to each mixture component) are defined within the HMM framework: if X_{pt} belongs to state i and mixture component j, then $Z_{pij} = 1$. Otherwise, $Z_{pij} = 0$. Therefore $\langle Z_{pij} \rangle = \sum_{t=1}^{T} \gamma^{C}_{pijt} = p(s = i, m = j | X)$ and the responsibilities are computed via a simple forward-backward procedure [26]. This last optimization completes the M-step.

In the E-step, the previously estimated parameters are kept fixed and $q(S, L)$ is estimated. As noticed in [96], Equation (39) can be re-arranged as

$$\mathcal{L}(q) = F(q(S, L)) - \mathrm{KL}(q(A, C, \pi, \alpha)|p(A, C, \pi, \alpha)) \ , \tag{61}$$

where

$$\begin{aligned} F(q(S, L)) = &\sum_{S} q(S) \int q(\pi) \ln(\pi_{s_1}) d\pi \\ &+ \sum_{S} q(S) \int q(A) \sum_{t=2}^{T} \ln(a_{s_{t-1},s_t}) dA \\ &+ \sum_{S,L} q(S, L) \int q(C) \sum_{t=1}^{T} \ln(c_{s_t,m_t}) dC \\ &+ \sum_{S,L} q(S, L) \int q(\alpha) \sum_{t=1}^{T} \ln(f(x_t|\alpha_{s_t,m_t})) d\alpha \\ &- \sum_{S,L} q(S, L) \ln(q(S, L)) \ , \end{aligned} \tag{62}$$

and the second term is fixed in this E-step.

By identification, I naturally define

$$\begin{aligned} \pi_i^\star &\triangleq \exp[\langle \ln(\pi_i) \rangle_{q(\pi)}] \ , \\ \pi_i^\star &= \exp[\Psi(w_i^\pi) - \Psi(\sum_i w_i^\pi)] \ , \\ a_{jj'}^\star &\triangleq \exp[\langle \ln(a_{jj'}) \rangle_{q(A)}] \ , \\ a_{jj'}^\star &= \exp[\Psi(w_{jj'}^A) - \Psi(\sum_{j'} w_{jj'}^A)] \ , \\ c_{ij}^\star &\triangleq \exp[\langle \ln(c_{ij}) \rangle_{q(C)}] \ , \\ c_{ij}^\star &= \exp[\Psi(w_{ij}^C) - \Psi(\sum_j w_{ij}^C)] \ . \end{aligned} \tag{63}$$

Ψ denotes the Digamma function and $\langle . \rangle$ denotes an expectation with respect to the quantity indicated as a subscript.

The last quantity to be optimized is

$$\ln(p^{\star}(X_t|\alpha_{s_t,l_t})) = \int q(\alpha)\ln(p(X_t|\alpha_{s_t,l_t}))d\alpha \ , \tag{64}$$

with

$$p(X_t|\alpha_{s_t,l_t}) = \left[\frac{\Gamma(\sum_{l=1}^{D}\alpha_{ijl})}{\prod_{l=1}^{D}\Gamma(\alpha_{ijl})}\prod_{l=1}^{D}X_{tl}^{\alpha_{jl}-1}\right]^{\gamma_{ijt}^{C}} . \tag{65}$$

Substituting Equation (65) into Equation (64) yields

$$\ln(p^{\star}(X_t|\alpha_{s_t,l_t})) = \gamma_{ijt}^{C}\int q(\alpha)\ln\left(\frac{\Gamma(\sum_{l=1}^{D}\alpha_{ijl})}{\prod_{l=1}^{D}\Gamma(\alpha_{ijl})}\right)d\alpha + \gamma_{ijt}^{C}\int q(\alpha)\sum_{l=1}^{D}(\alpha_{ijl}-1)\ln(X_{tl})d\alpha \ . \tag{66}$$

The second integral of Equation (66) can be expressed as

$$\sum_{l=1}^{D}\ln(X_{tl})\langle\alpha_{ijl}-1\rangle_{q(\alpha)} = \sum_{l=1}^{D}\ln(X_{tl})\left(\frac{u_{ijl}}{v_{ijl}}-1\right) , \tag{67}$$

while the first integral can be expressed in the form of an expectation that I denote $J(\alpha_{ijl})$

$$J(\alpha_{ijl}) = \left\langle \ln\left(\frac{\Gamma(\sum_{l=1}^{D}\alpha_{ijl})}{\prod_{l=1}^{D}\Gamma(\alpha_{ijl})}\right)\right\rangle_{q(\alpha)} . \tag{68}$$

This expression is analytically intractable and I choose to use the approximation proposed in [11] in which a lower bound of this quantity is derived and equals to

$$\begin{aligned} J(\alpha_{ijl}) \geq \bar{\alpha}_{ijl}\ln(\alpha_{ijl})\Bigg\{\Psi\Bigg(\sum_{d=1}^{D}\bar{\alpha}_{ijd}\Bigg) - \Psi(\bar{\alpha}_{ijl}) \\ + \sum_{d=1,d\neq l}^{D}\bar{\alpha}_{ijd}\Psi'\Bigg(\sum_{d=1}^{D}\bar{\alpha}_{ijd}\Bigg)(\langle\ln(\alpha_{ijd})\rangle - \ln(\bar{\alpha}_{ijd}))\Bigg\} , \end{aligned} \tag{69}$$

with

$$\bar{\alpha}_{ijl} = \frac{u_{ijl}}{v_{ijl}} , \tag{70}$$

and

$$\langle \ln(\alpha_{ijd}) \rangle = \Psi(u_{ijd}) - \ln(v_{ijd}) \ . \tag{71}$$

Substituting Equations (63) and (69) into Equation (62), I obtain

$$F(q(S,L)) = \sum_{S,L} q(S,L) \ln \left(\frac{\pi^\star_{s_1} \prod_{t=2}^{T} a^\star_{s_{t-1},s_t} \prod_{t=1}^{T} c^\star_{s_t,m_t} f^\star(X_t|\alpha_{s_t,m_t})}{q(S,L)} \right) . \tag{72}$$

The optimized $q(S,L)$ then is expressed as

$$q(S,L) = \frac{1}{W} \pi^\star_{s_1} \prod_{t=2}^{T} a^\star_{s_{t-1},s_t} \prod_{t=1}^{T} c^\star_{s_t,l_t} p^\star(X_t|\theta_{s_t,l_t}) \ , \tag{73}$$

with the normalizing constant W,

$$W = \sum_{S,L} \pi^\star_{s_1} \prod_{t=2}^{T} a^\star_{s_{t-1},s_t} \prod_{t=1}^{T} c^\star_{s_t,l_t} p^\star(X_t|\theta_{s_t,l_t}) \ . \tag{74}$$

Equation (74) is actually the likelihood of the optimized model $(A^\star, C^\star, \pi^\star, \alpha^\star, S, L)$ and can be computed using a forward-backward algorithm [26, 96].

Typically, the learning of all parameters is achieved through a succession of M-steps followed by E-steps. $F(q)$ is estimated at the end of each iteration until convergence is reached.

Precisions about the implementation

Initial values for all priors have to be set. A natural choice for the Dirichlet prior parameters for A, C and π, is to take uniform hyperparameters. Therefore, no state or mixture component is favored before the beginning of the learning scheme. For accurate convergence u and v have to be set more carefully. A method of moments is used over the available data in order to get a rough estimation of the Dirichlet parameters. I then set u equal to these roughly estimated values of α and set v equal to 1. Therefore, the ratio of the two hyperparameters matches with the mean of the Dirichlet parameters values.

The difference between the initialization of the Baum-Welch algorithm (non-variational) and the variational approach are crucial. In the variational case, only the parameters of the distributions that the different HMM parameters follow (i.e., the hyperparameters) are initialized. In the Baum-Welch approach, each HMM parameter has to be carefully initialized. Assumptions are thus made at a higher level in the variational approach. It makes the initialization more straightforward as for instance, only one parameter is needed for the initialization of A, C, and π (with the assumption of uniform priors). Using a Gamma prior for the Dirichlet parameters requires the initialization of two hyperparameters u and v. However, the Dirichlet parameters are related to these parameters through their ratio (that is the mean of the Gamma distribution), which makes their initialization reduced to the initialization of a single parameter. This makes the entire variational initialization less restrictive than the one of the previous chapter and the results less dependent on the initialization. Similar conclusions are drawn for mixture models with the use of the variational approach with respect to a non-variational one in [11].

However, as specified earlier, I found important in the experiments to use a rough approximation of the Dirichlet parameters α_{init} for the initialization of the ratio u/v in order to prevent the convergence towards a local maximum.

Furthermore, the estimation of the hyperparameters does not allow for a direct estimation of the Dirichlet parameters but only of their average values. Therefore, some simple manipulations of the quantities have to be performed in order to get accurate estimations of the α_i's. To start with, by definition of the Gamma distribution I have

$$u/v = \langle \alpha \rangle \ , \tag{75}$$

where the dimension subscripts have been omitted for the sake of clarity.

By approximating α by its mean, I can write the following

$$\frac{u/v}{\sum u/v} \approx \frac{\alpha}{\sum \alpha} \, . \tag{76}$$

In the multidimensional case, when estimating the updated value of α, the sum over the dimensions is not available. However, it can be approximated by the sum of the initial vector of α parameters (obtained with the method of moments). Therefore, the updated value of α, denoted α', is approximated as follows

$$\alpha' = \frac{u/v}{\sum u/v} \times \sum \alpha_{init} \, . \tag{77}$$

This approximation is especially used when computing the responsibilities of the HMM γ and ξ, in the forward-backward algorithms.

In summary, the pseudo-code for the variational Bayesian learning of the Dirichlet-based HMM is presented as Algorithm 1.

Algorithm 1 Variational Bayesian Learning of Dirichlet-based HMM

```
function vbhmmDirLearn($X$, $\alpha_{init}$, $K$, $M$, tol, maxIter)
    ## Initialize hyperparameters ##
    $\phi^A = ones(1, K) \times 1/K$
    $\phi^C = ones(1, M) \times 1/M$
    $\phi^\pi = ones(1, K) \times 1/K$
    $v_{ijl} = 1, \forall i, j, l$
    $u_{ijl} = \alpha_{init_{ijl}}, \forall i, j, l$
    ## Initialize HMM parameters ##
    Draw the initial responsibilities $\gamma^A$, $\gamma^C$, and $\gamma^\pi$ from prior distributions with Eq. (40)
    Compute $w^A$, $w^C$, and $w^\pi$ with Eqs. (45), (49), and (52)
    Initialize $A$, $C$, and $\pi$ with coefficients computed with Eq. (63)
    ## Initialize HMM likelihood results and iteration count ##
    $hlik^{old} = 10^6$; $hlik^{new} = 10^5$; $iter = 0$
    while $|hlik^{old} - hlik^{new}| \geq tol \; \& \; iter \leq maxIter$ do
        ## E-Step ##
        Compute data likelihood $dlik$ using $X$, $u$, $v$, and $\alpha_{init}$ with Eqs. (34) and (77)
        Compute responsibilities $\gamma^A$, $\gamma^C$, and $\gamma^\pi$ with forward-backward procedure using $dlik$, $A$, $C$, and $\pi$. Eqs. (46), (50), and (53)
        Update $u$ and $v$ with Eqs. (57) to (60)
        ## M-Step ##
        Update $w^A$, $w^C$, and $w^\pi$ using responsibilities $\gamma^A$, $\gamma^C$, and $\gamma^\pi$ with Eqs. (45), (49), and (52)
        Update $A$, $C$, and $\pi$ using $w^A$, $w^C$, and $w^\pi$ with Eq. (63)
        ## Update stopping criteria ##
        $hlik^{old} \leftarrow hlik^{new}$
        Compute $hlik^{new}$ with Eq. (74) and forward-backward procedure
        $iter += 1$
```

4.3 Generalized Dirichlet-based variational Bayesian HMM

The model presented in Section 4.2 can be extended into a generalized Dirichlet based model. The generalized Dirichlet distribution extends the Dirichlet one by relaxing the constraint on the sign of the data covariance at the cost of being represented by more parameters. A D-dimensional generalized Dirichlet distribution is defined as:

$$GD(\vec{x}|\vec{\alpha},\vec{\beta}) = \prod_{d=1}^{D} \frac{\Gamma(\alpha_d+\beta_d)}{\Gamma(\alpha_d)\Gamma(\beta_d)} x_d^{\alpha_d-1}\left(1-\sum_{r=1}^{d} x_r\right)^{\nu_d}, \tag{78}$$

where Γ denotes the Gamma function and $\vec{\alpha} = (\alpha_1, ..., \alpha_D)$, $\vec{\beta} = (\beta_1, ..., \beta_D)$, $\vec{\alpha}$ and $\vec{\beta}$ the distributions' parameters, all real and strictly positive. ν_n is a combination of these parameters and equals to $\beta_d - \alpha_{d+1} - \beta_{d+1}$, if $d \neq D$, and to $\beta_D - 1$, otherwise. The definition holds for positive data that sum up to less than one: $\vec{x} \in \mathbb{R}_+^D$ and $\sum_{d=1}^{D} x_d < 1$. Therefore $\vec{x}$ can be extended to a proportional vector of dimension $(D+1)$, where the $(D+1)$-th element completes the vector to 1.

This distribution has an interesting and convenient property, which calls the need to express the problem in a transformed space that I refer to as the W-space. Each observation vector $\vec{x}$ is projected to the W-space through the following bijective function [60, 61]:

$$W_l = \begin{cases} x_l & \text{for } l = 1 \\ x_l/(1-\sum_{i=1}^{l-1} x_i) & \text{for } l \in [2, D] \ . \end{cases} \tag{79}$$

One can mathematically show that $\forall l, W_l \sim \text{Beta}(\alpha_l, \beta_l) = \text{Dir}(\alpha_1 = \alpha_l, \alpha_2 = \beta_l)$. The estimation problem is thus simplified as D smallest estimation problems of unidimensional Dirichlet distributions. This problem splitting has the advantage of improving the precision of the overall estimation. Indeed, the parameters being estimated one by one, an estimation error for one of them has no impact over the others. Also, problems of smaller dimension are in general easier to solve with precision than problem of higher dimensions. The combination of these two advantages brings a substantial enhancement of the model fitting to the data, and efficiency for further classification tasks.

The framework is can be summarized as the following steps, a more complete pseudo-code is given as Algorithm 2:

1. The data are projected onto the W-space.
2. For each dimension, the projected data are complemented to 1.
3. The VB-HMM with the Dirichlet assumption is applied to the complemented data, one dimension at a time.
4. The estimated parameters are kept as the GD parameters for the original data.
5. The estimation of the other HMM parameters is still ran with the original data.

The interesting point with this extension is that it actually brings back all parameters to unidimensional Beta. This is the special case for which the approximate conjugate prior chosen for the Dirichlet actually matches the exact prior. Therefore, I can expect significant improvements of the performance of this version of the algorithm compared to the Dirichlet version.

4.4 Application: Anomaly detection in crowds

A first set of experiments over the public UCSD data sets *Ped1* and *Ped2* is carried out, with the goal of detecting unusual behaviors in crowds of pedestrians from video surveillance recordings on an academic campus [55]. I refer the reader to the previous chapter, Sections 3.4.2 and 3.4.3 for the details about these data sets and the features I extract from them.

For having comparable results with [52] and Chapter 3, all frames are converted to grayscale and resampled to the size 160 × 240 pixels, using a bicubic interpolation. A Gaussian filter of size $[3, 3]$ with $\sigma = 1.1$ is also applied for reducing the noise. To keep the comparison of the HMM learning methods as relevant as possible, the features that I use are the same as in the previous chapter and therefore, variations in the final detection

Algorithm 2 Variational Bayesian Learning of generalized Dirichlet-based HMM

```
function vbhmmGDLearn($X$, $\alpha_{init}$, $\beta_{init}$ $K$, $M$, tol, maxIter)
    ## Initialize hyperparameters and transform data ##
    Initialize $\phi^A$, $\phi^C$, and $\phi^\pi$ as in Algorithm 1
    Project $X$ in W-space as $W_l, \forall l$ with Eq. (79) and complement projected data to 1 $(W_l, 1 - W_l)$
    Initialize $(u_{ijl}, v_{ijl}), \forall i, j, l$
    ## Initialize HMM parameters ##
    Initialize $A$, $C$, and $\pi$ as in Algorithm 1 using Eqs. (40), (45), (49), (52), and (63)
    ## Initialize HMM likelihood results and iteration count ##
    $hlik^{old} = 10^6$; $hlik^{new} = 10^5$; $iter = 0$
    while $|hlik^{old} - hlik^{new}| \geq tol$ & $iter \leq maxIter$ do
        ## E-Step ##
        Compute data likelihood $dlik$ using $X$, $u$, $v$, $\alpha_{init}$, and $\beta_{init}$ with Eqs. (77) and (78)
        Compute responsibilities $\gamma^A$, $\gamma^C$, and $\gamma^\pi$ with forward-backward procedure using $dlik$, $A$, $C$, and $\pi$. Eqs. (46), (50), and (53)
        for $l \in [1, \ldots, D]$ do
            Update $u_{ijl}$ and $v_{ijl}$, $\forall i, j$ with Eqs. (57) to (60) where $W_l$ is used in place of $X$
        ## M-Step ##
        Perform M-Step as in Algorithm 1
        ## Update stopping criteria ##
        Update stopping criteria as in Algorithm 1
```

accuracy can only be attributed to the HMM parameter estimation method (EM versus variational).

I recall that an HMM is trained for each cuboid location using all the available observations from the training video sequences. When there is a new query sequence to be classified, the likelihood of each cuboid in this new sequence is computed with respect to the HMM that has been estimated at this location. This result is compared to a pre-defined threshold which is tuned for each location using the minimum likelihood value of the training samples multiplied by a factor k. The multiplicative factor prevents potential outliers in the training data to corrupt the classification results. Section 3.4.2 of Chapter 3 discusses this multiplicative factor influence.

The cuboids used are of spatial size 40×40 pixels and have an 8-frame depth. Therefore each frame is spread across 77 cuboids, each modeled by an independent HMM. As explained before, video sequences to be classified are processed such as all their feature vectors are computed. Then, the likelihood of the series of feature vectors (i.e., histograms) with respect to their corresponding model are computed and compared to the pre-tuned threshold. The anomaly detection part is run several times with different values of k in order to obtain the performance at EER.

For comparison on the learning method only, I adopt the same initialization procedure as in Chapter 3. As no significant improvement can be observed beyond 6 clusters, following the Occam's razor principle, the product $M \times K$ is set to 6, with $M = 3$ and $K = 2$. Finally, training sequences numbered 2, 23, and 25 are not used for the training phase as unexpected anomalies have been spotted in them.

Each experiment is run 10 times and the results are averaged and analyzed at two different levels.

- At the frame-level: The information I am looking at is whether a frame contains or not an anomaly. The localization of the anomaly is not taken into account.

- At the pixel-level: An anomaly is considered as well-detected if and only if it is detected at the correct location within the frame. Following the rules established

> by [55] for this data set, an anomaly is correctly located if at least 40% of its pixels are detected as anomalous. As for the frame-level detection, if a frame does not contain an anomaly but one pixel of the frame is detected as such by my algorithm, the frame is considered as a false positive. Pixel annotations of each anomaly for every abnormal frame are publicly available [3, 55].

The True Positive and False Positive Rates can therefore be expressed as:

$$\text{TPR} = \frac{\#\text{of true positive frames}}{\#\text{of positive frames}}\,, \tag{80}$$

$$\text{FPR} = \frac{\#\text{of false positive frames}}{\#\text{of negative frames}}\,. \tag{81}$$

Performance is then expressed in terms of EER for the frame-level analysis, which is the ratio of misclassified frames at which the two error types equal, i.e., $FPR = 1 - TPR$, and in terms of True Detection Rate (TDR), i.e., $1 - EER$, at pixel-level. The EER at pixel-level will differ form the one at frame-level due to the "lucky guess" detections, which are anomalous frames detected as such but at the wrong spatial location. The pixel-level results seem to me more relevant but for fair comparison with other approaches I present both results in Table 4.1.

The most relevant and meaningful comparison of this work is the one with the results of the previous chapter. Indeed, the implementation of the experiments over these data sets only differs by the learning method used for the HMMs. The improvement in detection accuracy and good localization brought by the use of the variational approach is clear, especially under the generalized Dirichlet assumption. The comparison with [52], shows the importance of the classifier choice for equivalent features.

Focusing on papers of the last few years leading to the current best results, I compare my approach with 7 other methods. [22] used a Gaussian-based HMM along with spatio-temporal features based on a texture map and 3D Harris functions. As my approach also uses spatio-temporal features and HMMs, it provides an interesting comparison on how the

Method	EER-Ped1	TDR-Ped1	EER-Ped2	TDR-Ped2	Processing time (inferring), Config., Language
[52]	31.0%	$\star$ 29%	30.0%	29%	0.1s/fr, CPU: 2.6GHz, RAM: 3GB
[22]	32.4%	N/A	28.5%	N/A	5.1s/fr, CPU: 2GHz, RAM: 4GB, Matlab
[92]	19.9%	68.2%	N/A	N/A	1.3s/fr, CPU: 3.4GHz , RAM: 4GB, Matlab
[93]	2.9%	N/A	9.9%	N/A	N/A
[59]	27.0%	78.9%	26.9%	74.9%	1.2s/fr, CPU: 3.5GHZ, RAM: 16GB, C++
[54]	17.8%	64.8%	18.5%	70.1%	1.2s/fr, CPU: 3.5GHZ, RAM: 16GB, C++
[105]	24.0%	81.3%	24.4%	81.9%	0.4s/fr, CPU: 2.8GHz, RAM:128GB
[106]	N/A	N/A	19%	76%	0.04s/fr, CPU: 3.5GHz, RAM: 8GB, Matlab
HMMD	28.9%	$\star$ 52.0%	18.5%	77.2%	0.2s/fr, CPU: 3.4Ghz, RAM: 5GB, Matlab
HMMGD	29.0%	$\star$ 51.0%	22.0%	74.7%	0.2s/fr, CPU: 3.4Ghz, RAM: 5GB, Matlab
HMMBL	29.0%	$\star$ 52.0%	16.6%	73.5%	0.2s/fr, CPU: 3.4Ghz, RAM: 5GB, Matlab
VBHMMD	31.4%	57.4%	12.5%	74.8%	0.2s/fr, CPU: 3.4Ghz, RAM: 5GB, Matlab
VBHHMGD	29.0%	61.8%	13.8%	80.3%	0.2s/fr, CPU: 3.4Ghz, RAM: 5GB, Matlab

Table 4.1: EER for the detection task and TDR at pixel-level for the localization task over the *UCSDPed1* and *UCSDPed2* data sets. The $\star$ symbol indicates results obtained over the original partially annotated ground truth (later completed by [3])

combination features/emission probability can affect the global performance. [93] devised an original approach by using points of interest detected using 3D Harris corner functions at which histograms of gradients and optical flow are computed for appearance and motion modeling. These information along with entities interactions are formulated as a graph which is used along with an SVM for binary classification. However, the anomaly detection is only performed at the frame-level. In the same trend, the method in [92] is based on describing the frequent geometric patterns between spatio-temporal points of interest and makes use of 3D-SIFT features and of Gaussian process regression for the modeling. [59] combined histograms of oriented swarms (dynamics modeling) with histograms of oriented gradients (appearance modeling) along with an SVM, and [54] proposed a hierarchical approach using several spatial scales and mixtures of dynamic textures to build a normalcy model. [105] presents an original use of spatial-temporal convolutional neural networks applied to small spatio-temporal video volumes selected using optical flow. The networks are fed with raw data bypassing the feature design step while capturing appearance and motion information. Finally, [106] proposes to use a combination of two local self-similarity descriptors (spatial and temporal) with a global descriptor learned using auto-encoders. Reliable detection is obtained by fusing the local and global results.

To strengthen results comparison across all methods, Table 4.1 reports the frame processing time, the computer configuration, and the implementation language used. My method counts among the fastest ones allowing near real-time processing. As a side note, the reported processing times are given as a simple indication of what one can expect as frame processing rate. One has to keep in mind that such processing times are heavily coupled to the employed programming methods such as the use of parallel computing, vectorization, and optimization techniques at large.

These results also clearly illustrate how the use of the generalized Dirichlet distribution can drastically enhance the global performance of the variational Bayesian HMMs. These improvements can have three sources:

- The more flexible covariance structure of the GD distribution (not restricted to be negative) [61].
- The relaxation of the assumption over the statistical independence of the distribution parameters, making the approximation of the conjugate prior over the distribution parameters tighter.
- The splitting of the main estimation problem into independent sub-problems of lower dimension via Equation (79)

The results over the Ped1 data set can seem a bit far from the best obtained results in the last few years. However, this has to be analyzed with respect to the frame processing time which is definitely lower than these more accurate methods. Only [105] and [106] show equivalent or greater potential for efficient, real-time anomaly detection. My proposed approach provides the second to best results up to date over the Ped2 data set while being faster than the best one in the literature to the best of my knowledge. The combination of the chosen features with the HMMs seems to work clearly better over sequences featuring pedestrians walking across the camera field. This is opposite to the results of [55] (not reported in Table 4.1) that used optic flow based features, which is rather close to gradient based features, but associates them with a bag-of-word representation of the video sequences. This added to the fact that I use features that are similar to [52] which did not get this difference of performance between the two data sets, tend to indicate this behavior might be induced by the HMMs themselves. The results presented in the previous chapter comfort this interpretation. Overall, my system seems to be especially sensitive to motion as a sideways view visually enhance speed variations compared to a front view.

4.5 Application extension: Tamper detection with still and active camera

4.5.1 Synthetic tampering event detection with still camera

Tamper detection refers to the task of detecting when the video surveillance device experiences a material issue such as defocusing, lens breaking, drive mechanism failure, voluntarily occlusion of the visual field, among others. When such event occurs, the surveillance mission cannot be successfully performed anymore and, in most cases, a human operation is needed in order for the system to work again. Furthermore, when a malicious individual wants to operate in the camera field, one of his first act is likely to be directed towards the destruction or at least impairment of the surveillance system. For these reasons, it is highly desirable to have a solution working for both unusual events detection and tamper detection, as these are complementary tasks aiming towards a same goal of public security.

The UCSD data set does not contain any example of tampering events. I synthetically generated a set of 4 different types of tampering actions over the 10 first testing video sequences of the *Ped1* data set, taking as model some of the tampering events proposed in [102]:

- *Total Occlusion*: Simulates the case in which the camera is fully covered by an opaque object. I took frames from videos of the tamper detection data set presented in [102] (see next section) that I introduced over the UCSD video sequences.

- *Partial Occlusion*: Simulates the case in which the camera field is partially occluded by an opaque object. I simply added black patches at different spots in the video sequences.

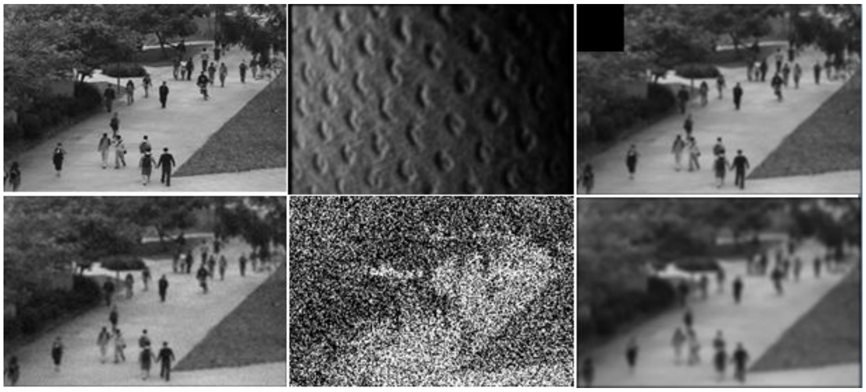


Figure 4.1: Examples of tampering events on UCSD Ped1 frames. Upper row: normal frame (left), total occlusion (center), partial occlusion (right). Lower row: light jitter (left), strong jitter (center), defocus (right)

Table 4.2: Number of tamper events detected over the total number of tamper events.

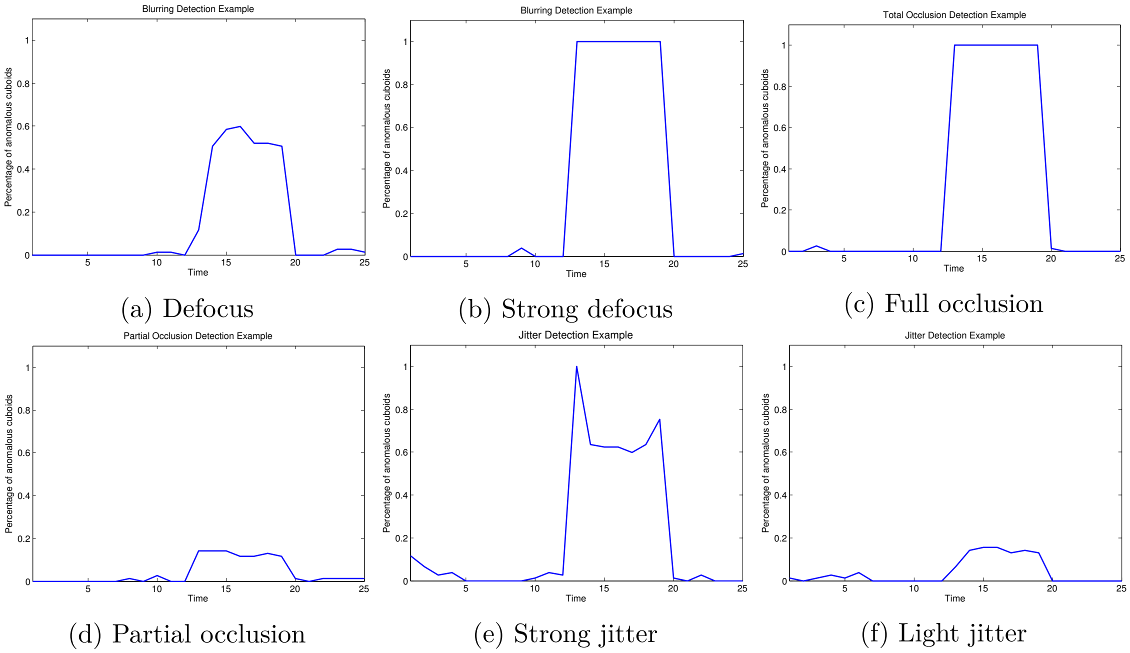
Tamper	HMMD	HMMGD	VBHMMD	VBHMMGD
Total Occlusion	10/10	10/10	10/10	10/10
Partial Occlusion	9/10	10/10	10/10	10/10
Jitter	10/10	10/10	10/10	10/10
Defocus	10/10	10/10	10/10	10/10

- *Jitter*: Camera jitter impairs the quality of the images recorded by the camera. I used the method [107], based on [108] and [109], for adding this effect to the video sequences, with different settings.

- *Defocus*: When the camera is brutally defocused, a blurry effect appears on the video frames. I added a rough Gaussian filtering (with window size from 5 × 5 to 50 × 50) to simulate this tampering event.

Samples of the different tampering events are reported in Figure 4.1.

The results presented in Table 4.2 are in line with my expectations. Detecting an anomaly occurring at a larger scale that the one of a pedestrian can be done flawlessly by the proposed approach. The same framework but with a non-variational learning fails at detecting the partial occlusion when it occurs in the bottom right corner. This may be due to the fact that no dynamic event occurs in this part of the frame in most of the



(a) Defocus

(b) Strong defocus

(c) Full occlusion

(d) Partial occlusion

(e) Strong jitter

(f) Light jitter

Figure 4.2: Detection results for random runs with the HMMVBD (GD version gives similar results).

training samples. Figure 4.2 reports some of the detection graphs in which the percentage of abnormal cuboids is plotted with respect to time.

I can conclude that my approaches makes the video surveillance system robust to tampering events while being efficient at detecting abnormal behaviors of pedestrians.

4.5.2 Tampering event detection with active camera

The lack of publicly available tamper detection data sets with a still camera setting did not allow me to experiment more over real video sequences. However, the authors of [102] built a tamper detection data set with video sequences recorded using a moving camera. Though my framework has been designed for a still camera viewpoint, I found interesting to give a try over this data set. It especially gives me insights about the strengths and weaknesses of the features I used for the main application of this chapter and the previous one.

I used the *Foyer* data set, part of the bigger *Visual Analysis of People* (VAP) data set [102]. The scenes are recorded in cycles from left to right and right to left with highly variable illumination conditions. The tamper events are grouped into 3 large categories that include in total 8 sub-categories: Occlusion (total occlusion, partial occlusion with fabric, partial occlusion from the camera positioning with respect to the room at the end of the cycle), Displacement (reduced camera field, moving up, and moving fast), and Focus (blur and motion blur). Between three and five samples for each event type are available in addition to a long training video sequence for the two surveillance areas.

In [102], specific features are designed for each type of tamper event to be detected. The design of such features is out of the scope of this paper and the only aim here is to investigate the capabilities and weaknesses of the proposed approach when applied to a situation in which the camera viewpoint is moving. In order to have a reference about the usefulness of the features I use with respect to the classification model, I also used the histograms of the intensities of the raw pixels for each full frame, still using a temporal length of 8 frames.

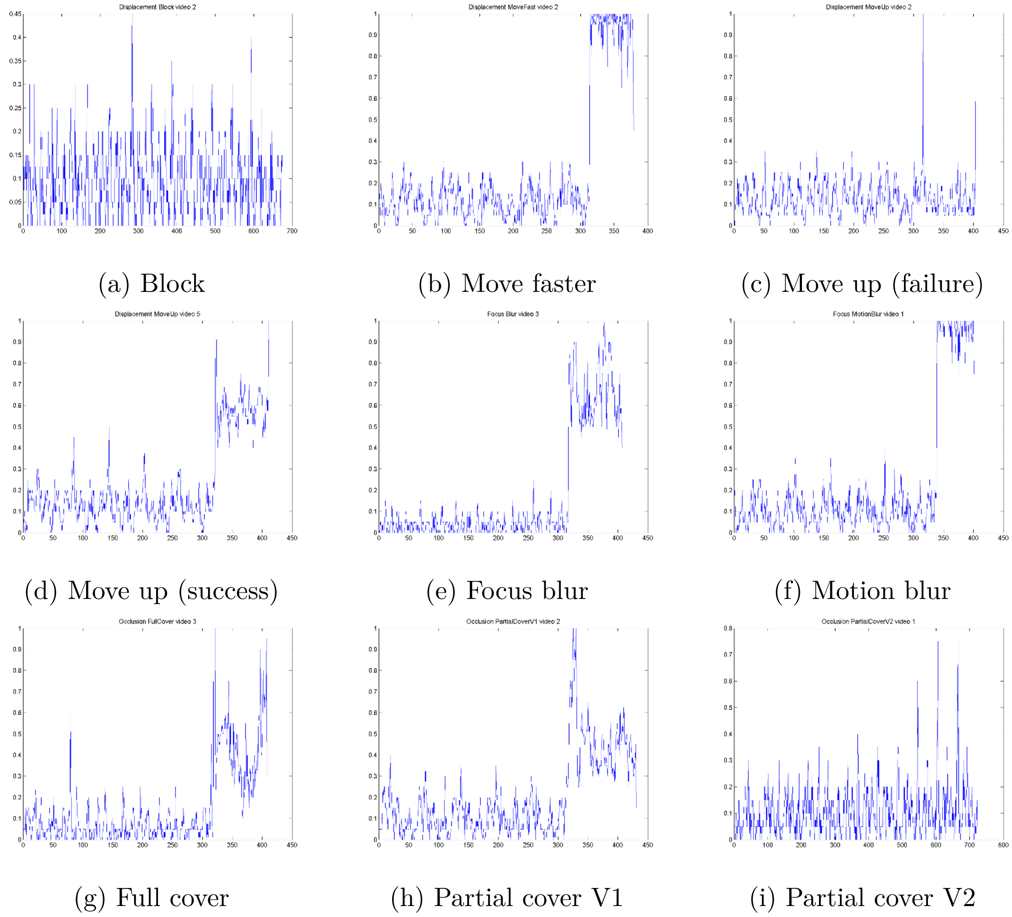
Table 4.3 gives the number of successful detections for the different tamper events for the 2 types of HMMs:

Table 4.3: Number of tamper events detected over the total number of tamper events.

Tamper	VBHMMD	VBHMMGD
Total Occlusion	0/3	3/3
Partial Occlusion V1	3/3	3/3
Partial Occlusion V2	0/3	0/3
Blur	5/5	5/5
Motion Blur	4/5	5/5
Block	0/5	0/5
Move Up	1/5	1/5
Move Fast	4/5	5/5

More specifically, in [102] SURF features for blurring events [110], histograms of oriented gradients (HOG) for occlusions, and features designed to track the displacement between successive frames for the abnormal displacement events. In an integrated system, it is difficult to imagine having a multitude of sophisticated features to be tracked in real time, each related to a specific type of tempering event (which can additionally take multiple forms).

Some successful and unsuccessful detection results are provided in Figure 4.3. In the light of the results presented in Table 4.3, I can draw the following observations.



(a) Block (b) Move faster (c) Move up (failure)

(d) Move up (success) (e) Focus blur (f) Motion blur

(g) Full cover (h) Partial cover V1 (i) Partial cover V2

Figure 4.3: Typical detection results for random runs with the VBHMMGD over the *Foyer* data set. The ordinates represent the percentage of cuboids being detected as abnormal.

- *Short span of the camera detection fails.* This is totally expected within my proposed framework. All seen situations are actually situation considered as normal by my model. For detecting such events, one can simply use the gradient-based features out of the histogram fashion, by tracking the average gradient along the x and y axes i.e., G_x and G_y, over time and comparing it to the pre-defined schedule movement.

- *Vertical move of the camera not well detected.* This is less expected but mostly comes from the fact I always look at the norm of the spatial gradients together. Therefore, transposing the horizontal move of the camera to a vertical one doesn't change the values of the features. Additionally, as I work with an active camera the range of appearance of training frames is wide and diverse, the difference in the frames appearance when moving vertically is not strong enough to raise any alarm. Tracking the average G_x and G_y over time and compare it to the pre-defined schedule movement should allow the detection of such events.

- *Partial cover at end of camera course (*PartialCoverV2*).* My features and model fail at detecting this situation. The reason is unclear but it is here a tamper event similar to a mix of the short span and partial occlusion *V1.* The event to be detected is short in length and most of the frame is normal.

The graphs reported in Figure 4.3 are raw and can be post-processed with a threshold and a moving average to be smoother. The moving average would allow some extremely short false detections that can occur regularly to be withdrawn. These are triggered in most case by the change of direction of the camera movement. The GD version of my approach allows the detection of more tamper events, especially the full cover one which is missed when using the Dirichlet.

4.6 Conclusion

As a conclusion, I derived the variational learning for the Dirichlet HMM and extended it to the generalized Dirichlet case. The combination of relaxing the constraint over the data, using an approximate conjugate prior, and splitting the main problem into independent lower dimensional sub-problems brought a clear improvement of the algorithm overall performance. With these approaches, I also propose a realistic solution to the problem of unusual event detection in crowds of pedestrians as they work in a near real-time fashion which can easily be improved towards real-time (with the training of the models done offline) using parallel programming. Their use yields convincing results over the UCSD data sets, among the current best state-of-the-art methods. The system is also robust to tamper detection which can be seen as anomalies at a larger scale. The application of the framework to video recorded using a moving camera showed surprisingly good results with respect to the fact the approach has been designed with the strong hypothesis that the camera is still (and thus the video frame split into cuboids). These latter experiments allowed to clarify the cases in which my method can fail.

Chapter 5

Extension to Hybrid HMMs for mixed data

5.1 Introduction

Along with the development of informatics and data collecting devices came data in the form of complex structures that classic tools can barely handle or to the cost of inaccurate modeling and extra complexity. Multivariate continuous and discrete mixed data became a topic of high interest in the last few years. These data structures especially arise in health studies, econometrics, genetics, and toxicology [111]. Basic methods such as the separate processing of the outcomes of different types, the discretization of the continuous outcomes, or the numerical scoring of the discrete outcomes suffer from both subjectivity (in the choice of the rules) and loss of information [112], and are thus not satisfactory. The separation of the outcomes, for instance, results in the loss of the correlation information and requires ad hoc methods to fuse the results if a classification or clustering task is performed.

As conventional tools do not allow easy processing of these data, new approaches and extensions of well-known methods have been developed. The estimation of mixed outcomes correlations have been studied in numerous papers and [111] provides a review of the main approaches that include methods based on joint probability factorization (ch. 6) and copula-based representation (ch. 10,11), pseudo-likelihood pair-wise factorization (ch. 9), and latent variable models (ch. 6). As for mixed data modeling, Bayesian Networks have been adapted in [113], and [114] generalized the latent variable analysis method. Clustering have been studied in [115] for continuous and count longitudinal data. A regularized classifier for discriminating between two classes with mixed continuous and categorical data is proposed in [116] and a minimum distance based classifier is defined in [117]. In [118], a classification model based on general mixed data models for mixtures of nominal, ordinal, and continuous variables is developed. This latter model is an extension of the general location model, one of the first proposed for mixed data analysis [119], in which categorical data are assumed to be marginally distributed as a multinomial. A normal distribution that controls the continuous variable is then assigned to each multinomial state. The main issue with this model, and with most of the models developed for health studies, is the poor performance or difficult generalization for multivariate data of more than a few dimensions.

In image processing, the pixels' intensity is typically considered as continuous. Therefore, continuous data of different types arise when different imaging systems are used to capture a same scene. This is the case when a scene is captured with both a color and an infrared cameras as reported in [120], a LiDAR (Light Detection and Ranging) and an optic imaging system [121], or by a radar and an optic imaging systems as reported in [1] and [2]. Especially, the noises corrupting each system can be of different natures and their joint processing has been seldom documented. SAR and ultrasound images typically contain speckle noise [122, 123], while optic images embed noise typically modeled as an additive Gaussian noise [124]. I found this topic to be understudied with only two approaches of radar-optic images joint processing for change detection that could be found in [2] and [1]. In the former work a method that is able handle images that have a slightly different point of view is proposed. It is based on the modeling of the dependence of the two images using the copulas theory. In the latter work a manifold is learned by taking into consideration the physical properties of the images, and especially the nature of the noise that they are corrupted by. On a close topic, Li et al. tackle the problem of conflict resolution in heterogeneous data from different sources that can arise due to transmission errors, high levels of noise, or malicious manipulation of the data [125]. They build an objective function that is solved as a joint optimization problem. The study of heterogeneous continuous/continuous data still needs a lot of studies to get a panoply of tools that can be chosen for processing them conveniently.

In this chapter, I propose to extend the hidden Markov model for multivariate data of mixed types. As presented so far, HMMs exist for continuous data of numerous types as well as for discrete data, but are not documented for multivariate mixed data. The proposed approach is by some points close to the multi-stream HMMs [126,127] theory that handles data coming from different sources but all modeled with distributions of the same type. My proposed framework can handle multivariate mixed continuous/continuous and discrete/continuous data. It does not make any assumption on the form of the probability mass functions (pmfs) of the discrete data, that can also be represented as mixtures of pmfs, which is rather unusual but essential in the case of mixed data modeling. A simple pmf mixtures estimation method is proposed, and can process outcomes taking their values into vocabularies (or ranges) of different sizes. This HMM can also fit continuous data by mixtures of several different probability distribution functions as long as the equations for the Baum-Welch algorithm (i.e., the EM procedure for the estimation of the HMM parameters) are derived. Modeling data using accurate pmfs and pdfs greatly reduces the number of parameters of the model compared with the more common fully Gaussian model.

The methods developed in the specific context of health studies often rely on basic knowledge of the physical meaning of the outcomes so their correlations, for instance, can be accurately modeled. However, and especially when working with images or videos, the physical meaning of the data to be processed is often hidden behind complex preprocessing operations (that lead to features) and it is thus essential to develop a model that does not require any prior knowledge on the data. My model takes implicitly into account the correlations between the different data types via the HMM parameters.

The work presented in this chapter has been presented at the 17th IEEE International Workshop on Multimedia Signal Processing (MMSP'15) in Xiamen, China, where it received one of the *Top 10% Paper Award*. The publication is referenced [33] in the Bibliography section.

In the following, Section 5.2 presents the approach and the EM-algorithm used for the parameters estimation. Section 5.3.1 reports the experiments led with synthetic data using the proposed hybrid HMM and a fully Gaussian HMM and Section 5.3.2 presents a real application for change detection in a pair of optic and SAR images corrupted by noises of different types (additive white Gaussian and multiplicative speckle noise, respectively). I finally conclude in Section 5.4.

5.2 Theory

I first set the HMM notations for this chapter: the hidden states are denoted $\{k_1, ..., k_T\}$, $k_j \in [1, K]$, with K the number of states, the pmf π controls the initial state, and the transition matrix B control the state transition at each time t. In the case of mixtures emission probabilities, the choice of the generating mixture m_t is driven by the mixing matrix C.

My hybrid HMM is able to handle sophisticated multivariate mixed data such as observations featuring different types of continuous outcomes (e.g., Dirichlet, Gamma, and Gaussian), as well as discrete ones. Different discrete outcomes can have a different number of categories, and the only assumption made in this work is the categorical or binary (two categories) nature of these discrete outcomes.

The HMM parameters estimation is done with an EM-algorithm relying on both *local* and *global* quantities, that relates to outcomes of one specific type and to all outcomes, respectively. The outcomes are assumed to be ordered by type i.e., the r_1-dimensional sub-vector formed by the r_1 first dimensions of the observation vectors follows an r_1-dimensional distribution and the following r_2-dimensional sub-vector follows another r_2-dimensional distribution (and so on if more than two distributions are present). When separate processing is done for each type of data/distribution, all the outcomes of this type are processed at once. The algorithm can be divided into four steps:

- **Initialization.** The continuous distributions parameters are separately initialized using methods of moments and the pmfs parameters by counting occurrences in the initial clusters, with the addition of a Dirichlet prior in order to avoid zero-values. Opposite to what is done in some work about HMMs such as in [28], the HMM parameters B, C, and π are not randomly initialized but also determined from the initial clustering (or part of it if the training set is too large), using a simple count of the transitions between the different clusters.

- **E-step.** In this step, the quantities

$$\begin{cases} \xi_{k_t,k_{t+1}} \triangleq p(k_t, k_{t+1}|x_0, ..., x_T) \ , & (82) \\ \gamma_{k_t,m_t} \triangleq p(k_t, m_t|x_0, ..., x_T) \ , & (83) \end{cases}$$

  are calculated with $\vec{x}$ a data sequence. For each type of outcome, *local* (ξ, γ) pairs are computed, as well as *global* pairs for all outcomes. To this purpose, the *local*, distribution-related, observation likelihoods are combined as a product that stands for the *global* likelihood used for the estimation of the *global* $(\xi_g$,$\gamma_g)$ pairs.

- **M-step.** In this step, all the parameters are updated. B, C, and π are updated with the *global* (ξ_g, γ_g) values with the usual formulas given in [28]. The different distributions parameters, denoted as θ_i's are computed using the *global* updated parameters B, C, π, along with the *local* quantities (ξ, γ).

- **Convergence criterion** Similarly to what presented in Chapter 3, the stopping criterion is the sum of an entropy and an energy that is computed as the sum of the energies contributions of the different types of outcome along with the HMM parameters. When this quantity evolves by less than a threshold set to 10^{-3} in my experiments or when a user-defined maximum number of iterations have been completed (10 in the following applications), the current parameters are kept.

Although this algorithm is extremely simple, it performs well and allows the processing of sophisticated data for which methods are severely lacking in the literature. The interlacement of *local* and *global* quantities allows to get a single HMM that fits mixed observations and is illustrated in Figure 5.1. In the E-step, the *global* likelihood computation, taken as the product of the different *local* likelihoods, is an approximation that assumes the different outcome types to be independent. However, the dependency between the outcomes is

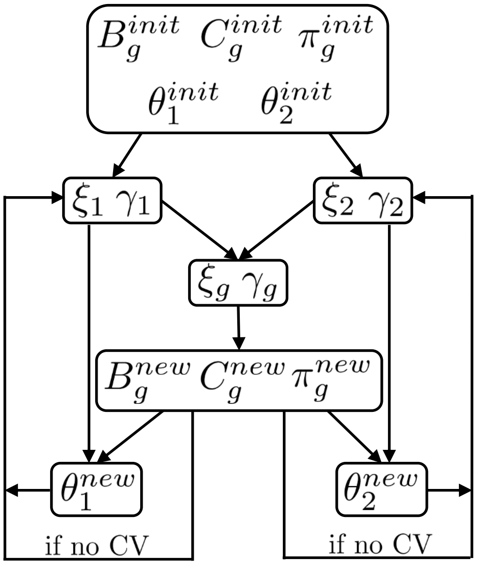

B_g^{init} C_g^{init} π_g^{init}
θ_1^{init} θ_2^{init}
ξ_1 γ_1
ξ_2 γ_2
ξ_g γ_g
B_g^{new} C_g^{new} π_g^{new}
θ_1^{new}
θ_2^{new}
if no CV
if no CV

Figure 5.1: *Local* and *global* variables dependencies. The loop is used only if convergence (CV) is not reached.

embedded into the HMM parameters B, C, and π, which seems to be sufficient for accurate modeling. Other approximations can be used, especially some developed for the multi-stream HMMs [126]. However, they are derived for a single type of distribution and their generalization to mixed data is not straightforward. I do not give any distribution type or outcome more relevance than another in the following experiments. Adding weights to the distribution types is possible though, and I refer the reader to the multi-stream HMMs theories for optimal weights estimation techniques [128–130].

I first experiment this hybrid HMM with synthetic data of three different mixture types namely, discrete, Gaussian, and Dirichlet. The use of pmfs mixtures into HMMs is rare and a single pmf is usually assigned to each state. I could not find in the literature clear indications about HMMs handling such mixtures and therefore developed my own method for their update. *Local* γ's are computed for each observation using the most recent parameters available with a forward-backward algorithm. I estimate the number of expected emissions for every discrete value, depending on the state and mixture component, as the sum of these *local* γ's. The results are then normalized to get valid pmfs. The Dirichlet implementation is done with the equations of [28] and makes use of the *global* γ value for the Dirichlet parameters estimation.

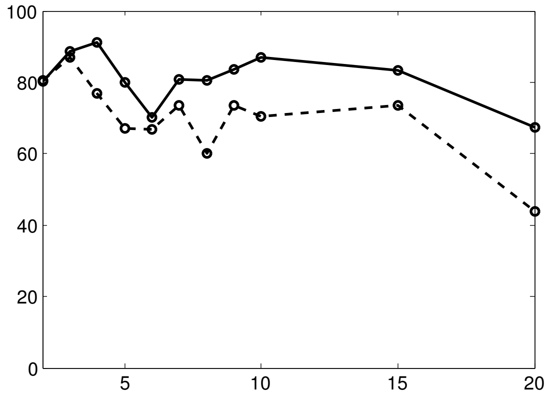


Figure 5.2: Retrieval rates for N Gaussian, N Dirichlet, and N discrete outcomes (5 categories) with $N \in [2, 20]$ for hybrid HMMs (plain line) and Gaussian HMMs (dashed line).

5.3 Experiments

5.3.1 Synthetic data

Random data are generated from combinations of the three aforementioned types with Gaussian means and Dirichlet parameters in the range [1,20], and Gaussian covariances in the range [1,5]. The distribution parameters are randomly generated, which penalizes the approach performance as the randomness does not insure the clusters to be actually very different from each other. When two clusters are too close, confusion is willing to occur, which lowers down the retrieval rate.

I use the same setting as in [28] and Chapter 3 of this thesis: 1000 sequences of length randomly chosen in the range [10,20] are generated from a known HMM. The source state and component of each observation sample is recorded. A fully Gaussian and a hybrid HMMs are trained from the same initial clustering and thus, the same initial parameters B, C, and π. I then try to retrieve the state and component that generated each observation sample. I fixed $K = 2$ and $M = 3$ and ran each specific setting (vocabulary length, dimension, distributions combination) at least 5 times. Results for a combination of the 3 outcome types are reported in Figures 5.2 and 5.3.

If N Gaussian, N Dirichlet, and N discrete outcomes are considered, hybrid HMMs outperform fully their Gaussian equivalents except for very low dimensions, whatever the number of categories the discrete outcomes can fall in. Figures 5.2 and 5.3 illustrate these

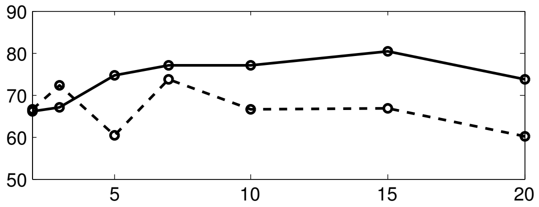


Figure 5.3: Retrieval rates for N Gaussian, N Dirichlet, and N discrete outcomes (10 categories) with $N \in [2, 20]$ for hybrid HMMs (plain line) and Gaussian HMMs (dashed line).

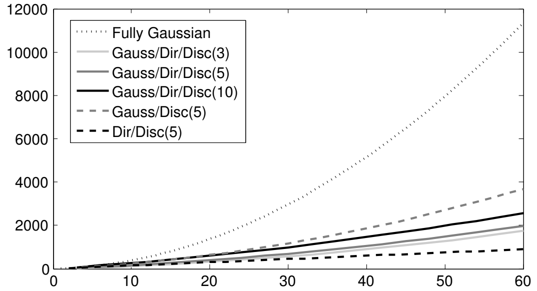


Figure 5.4: Number of model parameters as a function of the observations dimension for different HMM settings, assuming equal number of outcomes of each type. The number in parenthesis indicates the number of categories for the discrete outputs.

results. Moreover, the number of parameters of a Gaussian HMM grows with the square of the data dimension. The discrete and Dirichlet HMMs reduce this to a linear dependency. Therefore, whenever some parts of the outcome observations are discrete or proportional (i.e., strictly positive and summing up to 1), using appropriate mixture types to model them into an hybrid HMM is beneficial for the model compacity and thus, the processing time. Figure 5.4 illustrates the number of parameters growth for different mixed data settings.

The retrieval rates have also been computed for data different from the training data, giving same accuracies. Finally, the generation of samples from the trained HMM are retrieved by the original HMM at the same level of accuracy than the trained one. This shows the estimated HMM is accurate enough despite the approximations made in the computation of the *global* likelihood. Furthermore, it has to be noted that the computation of the retrieval rate at the sample level is highly penalizing as it does not take into account the sequences as a whole and thus, do not completely describe the classification potential of the model. Indeed, I found the trained hybrid HMMs to be able to classify unseen sequences among two classes almost perfectly, though their retrieval rates were around 75%.

The case of N Gaussian combined with N discrete outcomes, $N \in [2, 20]$, is not presented as a graph as I found the same retrieval rates either with hybrid HMMs or fully Gaussian ones. However, the number of parameters for the hybrid HMM is the lowest one, as shown in Figure 5.4, especially for high dimensions, and the Occam's razor principle will thus favor the choice of the hybrid model.

5.3.2 Change detection in satellite images

I validate my approach in a real situation by detecting changes in the pair of optic and SAR images presented in Figure 5.5. In the case of an exceptional event such as a natural disaster (e.g., flood, earthquake, drought, landslide,...), remote pictures of the area can be helpful for measuring the impacted area of the event and characterizing its evolution. To do so, a new image of the zone has to be obtained and then compared to a previously captured one. However, there are chances that the imaging sensor taking the second picture is not the same as the first one and can even have a totally different imaging system, especially in an emergency situation[1]. Keeping this context in mind, [1] proposed an approach to detect changes between an optic and a SAR image taken above the city of Gloucester. The SAR image has been acquired by the TerraSAR-X satellite in 2007 after a flood, while the optic image has been acquired before the flood. Figure 5.5 present the images used in this application. The SAR image is available at the address referenced [132]. The optic image as well as the hand-annotated change mask, are directly taken from the paper [1], so that I am sure that all the images are well registered.

SAR images are corrupted by speckle noise that is usually considered as a multiplicative Gamma noise [122], meanwhile optic systems are typically corrupted by an additive Gaussian noise [124]. I propose to train a hybrid HMM based on a univariate Gaussian distribution along with a univariate Gamma distribution. This choice of univariate distribution functions comes from the generalization of the Gamma distribution to a multivariate setting that is not straightforward. Indeed, it has been the topic of numerous studies [133], leading

[1] One has to keep in mind that I place myself in the context of satellite imagery. Satellites trajectories cannot be drastically changed (or at least not for the need of a single picture) and have a limited life span (typically 5 to 15 years depending on the mission) [131].

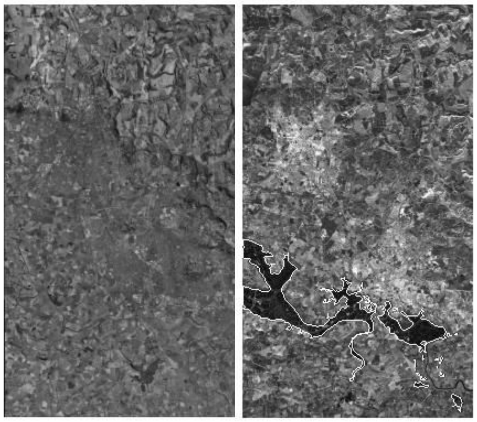


Figure 5.5: Optic image in gray levels (left) and SAR image with the change mask contour superimposed (right). The change area is the black area at the bottom of the image, surrounded by the white contour.

to various definitions. I thus keep the generalization of HMMs to multivariate Gamma distributions for future work. The use of univariate distributions imposes to describe the image as a sequence of overlapping vectors. Therefore, a horizontal vector and a vertical vector are built around each pixel i.e., the current pixel of interest lies at the center of the vectors.

Three zones of the images, clear of any change, are chosen for the HMM training. These patches correspond in total to less than 4% of the image and are reported in Figure 5.6. Within these zones, a total of 180 pairs of row vectors are used for the HMM training. One has to note here, that the use of columns vectors or of a combination of row and column vectors would be equivalent. As I work with land pictures taken at very-high altitude, no orientation carries more of less information than another. For the sake of clarity, the sequences used are then two-dimensional, the first dimension being pixels' intensities from the optic image, and the second dimension representing the same pixels' intensities from the SAR image. The chosen parameters for the method are a length of 21 for the vectors, and HMM parameters of $K = 4$ and $M = 1$. Typically, three landscape types can be seen in the images (cities, fields with relief, and flat fields). However, using only three states for the HMM is not satisfactory as the clustering of all data into three clusters does not explain enough variance of the training set to be reliable, as it can be seen in Figure 5.7. From this latter figure, $K = 4$ is inferred to be the acceptable minimum number of states.

Intuitively, the forth state can prevent the outliers, that can be due to extreme values of noise, for instance, to corrupt the estimation of the other states. Experiments with more than four states have shown a continuous degradation of the results. Finally, the choice of the vectors length is a trade-off between the detection precision (the longer the vector, the blurrier the contours of the detected changed area) and the multiplication of false detection (the smaller the vector, the more false detection there are).

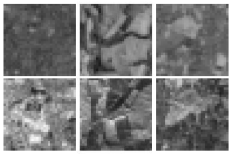


Figure 5.6: Patches taken from the optic image (top) and the corresponding patches from the SAR image (bottom).

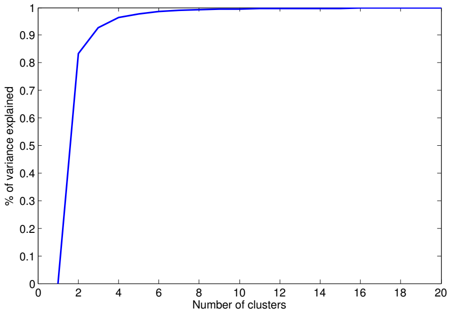


Figure 5.7: Percentage of variance explained in function of the number of clusters.

The equations for the update of the Gamma parameters are given in Appendix A. Once the model is trained, the images are then scanned horizontally (rows) and vertically (columns), the two-dimensional sequences are built, and the likelihood of each sequence with respect to the trained HMM is computed. This likelihood value is seen as the similarity measure of the vectors' central pixel between the two images. A high likelihood corresponds to a high similarity, while a low value corresponds to a low similarity. The intermediate likelihood results along the rows and the columns are reported in Figure 5.8.

I use a threshold to find the value of the False Alarm Rate (FAR) such that it equals $1 - \text{TD}$, TD being the True Detection rate. This value represents the Equal Error Rate. Figure 5.9 reports the similarity map obtained after summation of the log-likelihood maps obtained in the horizontal and vertical setting at the EER. The pixels at the border of the

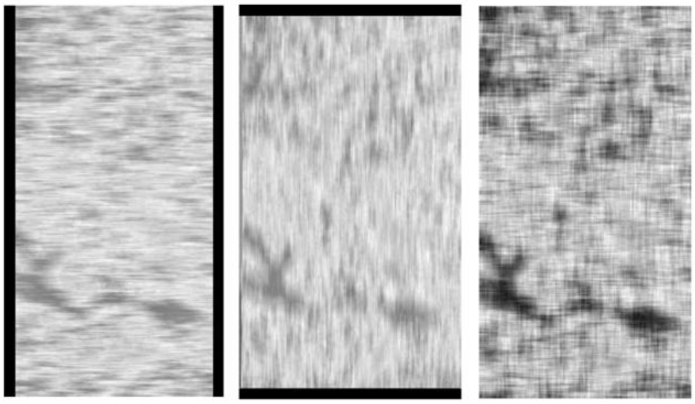


Figure 5.8: Normalized log-likelihood maps of the pixels processed in rows (left), columns (center), and the summation of these two log-likelihoods maps. The lowest the likelihood, the darker.

image cannot be the central pixel of both a vertical and a horizontal vector and, for fair comparison, I do not take them into account in the detection results. For fair comparison again, the training zones are excluded as well from the detection results. Averaged over 10 runs, the EER of this method for this pair of images is 16.75% with standard deviation 0.20. The processing of the pair of images, training included, takes less than 7 minutes.

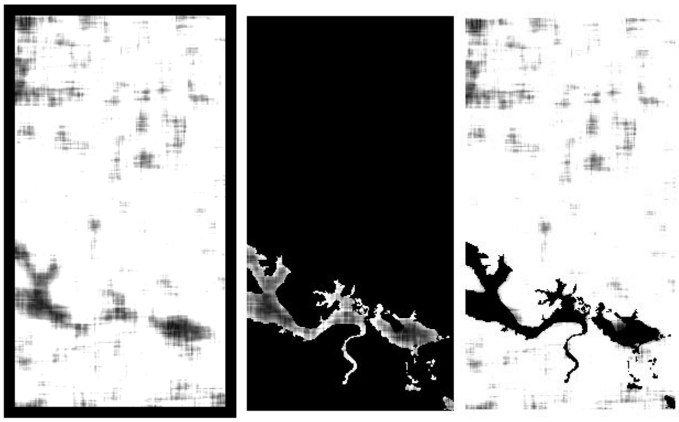


Figure 5.9: (left) Map of detected changes at FAR = 1−TD. The lowest the likelihood, the darker. The black contour helps for visualization. (center) Same map with the mask superimposed. (right) False detections.

This method raised artifacts, mostly isolated points detected with a low likelihood (most of them in light gray in Figure 5.9). In the case of a flood detection event, these points are more than unlikely to be true positives. A morphological operation over the resulting map is performed in order to remove these isolated points. With this extra operation, removing artifacts with a maximal size of 20 pixels and an 8-pixel connectivity, the EER drops to

16.11% with standard deviation 0.25.

I compare my method with the two recent state-of-the-art methods [1] and [2], and the following classic methods: mean pixel difference, mean pixel ratio, correlation coefficient, and mutual information. I did not re-implement these methods and directly took the results from [1]. Detection maps results are reported in Figure 5.10, and Table 5.1 summarizes the EER points for each method. These results have to be analyzed as being related to a unique pair of images. My approach performance is in line with the method recently proposed by Prendes et al. [1]. It outperforms the method of [2] and most of the classic methods. The good results of the Mean Pixel Ratio method can be attributed to the fact that the area to be detected is totally homogeneous [1]. In its current setting, the hybrid model I developed has only 22 degrees of freedom (the HMM parameters B and π, the Gaussian parameters, and the Gamma parameters), and thus provides a very compact representation of the similarity of the data.

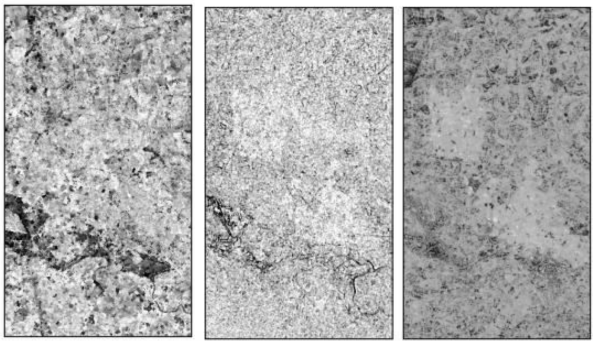


Figure 5.10: Raw results (no threshold) obtained with the methods from [1] (left), correlation coefficients (center), and [2] (right).

Method	FAR $= 1 -$ TD
Mine	16.75 ± 0.20
Mine - no artifacts	16.11 ± 0.25
Copulas [2]	23.96
Manifold learning [1]	14.58
Correlation coef.	31.19
Mutual Info.	23.13
Mean Px. Diff.	21.75
Mean Px. Ratio	18.61

Table 5.1: Comparison of different methods for change detection in the pair of images presented in Figure 5.5.

5.4 Conclusion

In this chapter, I proposed hybrid HMMs for mixed continuous/continuous and discrete/continuous data modeling as a first study on the topic to the best of our knowledge. I showed that while building these HMMs with a simple method, it is able to handle outcomes of 3 different types with a better accuracy and fewer parameters than fully Gaussian HMMs. The application to real data for change detection in a pair of optic and SAR images assessed that my method can compete with the current top state-of-the-art ones. While working on this topic, it became clear that to this day, tools that can easily handle mixed data are seldom and that this approach, with its adaptivity to numerous data types, gives a new alternative for processing such data. Numerous rich application such as the one used to illustrate the performance of the proposed method are waiting for such tools to be fully studied. As a side note, the use of a single pair of images in this chapter but also in the cited papers is due to the scarcity of the data in this domain, and to the fact most images are not made publicly available by the different space agencies over the world. Therefore, one can see the presented application as a proof-of-concept calling for a larger study involving more approaches and more data.

Chapter 6

Distances for Dirichlet and GD HMMs

6.1 Introduction

In the previous chapters we have seen that hidden Markov models are generative models which first mathematical foundations have been set off in the 1960's [25] and that are since then widely used in a variety of fields, from speech processing [134, 135] to image processing [136, 137], video processing [31, 138], and pattern recognition [139, 140] to name but a few. First developed for discrete and Gaussian data, I showed that they are still mainly used under these assumptions [22–24, 26], although more learning strategies have recently been proposed for multiple types of distributions such as the Poisson [138], Student's t [77], normal inverse Gaussian [78], contaminated Gaussian [141], Dirichlet [28] and, in this thesis, for the generalized Dirichlet (Chapters 3 and 4), Beta-Liouville (Chapter 3), and mixed distributions (Chapter 5). I recall that an HMM model can be denoted as $\lambda = (A, C, \pi, \theta)$, where A is the transition matrix defining the probability of transitioning from one state to another and C is the mixing matrix (only present when working with mixtures) defining the probability for each component within each mixture model. π is the probability mass function for the choice of the starting state and θ represents the parameters relative to the emission probability distributions.

Comparing the similarity of two HMMs has been first studied in [26] where a Kullback-Leibler (KL) divergence based on the limit of the log-likelihood of an infinitely long data sequence generated by one HMM is proposed. A good estimation is obtained when using a very long data sequence, which requires a lot of computations for the log-likelihood estimation. In this chapter, I carry out a comparative study of parametric distances for Dirichlet and generalized Dirichlet-based HMMs. The search of such distances relaxes many issues encountered when using data-dependent distances. Indeed, relying on data provides a non-deterministic distance while relying on parameters allows for deterministic distances to be built. Moreover, the availability of data is not granted in all cases and data generation can be difficult to achieve for some sophisticated distributions and is always time-consuming. Also, good accuracy with data-driven metrics is achieved to the cost of the use of very long data sequences. Finally, when working with distributions as the Dirichlet and the generalized Dirichlet, the variance is often underestimated leading to peaky distributions. Their likelihood values with respect to data samples go then beyond 1. In the forward algorithm used to estimate the HMM likelihood, these values are multiplied multiple times and, when the data sequence grows longer, computational overflow is often reached, making this method complex to implement and unreliable, as shown later in this chapter.

The literature about the design of deterministic metrics for continuous HMMs is scarce and most of the proposed distances or similarity measures require long data sequences generated from or modeled by the HMM to be computed [142–144]. Very few papers define such distances that can further generalize to mixture-based HMMs and all of them are defined in the context of the Gaussian. To the best of my knowledge, the only current approaches fulfilling these requirements are the approaches by Sahraeian and Yoon [145] and the approach by Zeng et al. [146]. The former defines similarity measures based upon the ability to match hidden states from the two HMMs and then measures the sparsity of the obtained correspondence matrix. This implies the choice of a distance to compare the emission probability distributions, taken as the Kullback-Leibler (KL) divergence in their study, which is transposed to a similarity measure by using its inverse or a negative exponential form of a multiple κ of it. How to tune this coefficient remains unclear. The original approach by Zeng et al. [146] relies on the computation of cumulative distribution functions for building a global cumulative function for each HMM. These cumulative functions that are then compared over the range of possible (or most probable) values for the observations. This metric, named HSD, is thus constrained to be used for unidimensional observations only.

A true distance is expected to verify the 4 following conditions but when working with sophisticated spaces, it is rather common to also define semi-distances that only verify the 3 first conditions. Denoting $(\lambda_1, \lambda_2, \lambda_3)$, three HMMs, $\forall\lambda_1, \forall\lambda_2, \forall\lambda_3$:

- Non-negativity: $dist(\lambda_1, \lambda_2) \geq 0$
- Identity: $dist(\lambda_1, \lambda_2) = 0 \iff \lambda_1 = \lambda_2$, where the equality between two models is defined by the equality of all their parameters, allowing permutations.
- Symmetry: $dist(\lambda_1, \lambda_2) = dist(\lambda_2, \lambda_1)$
- Triangle inequality:
  $dist(\lambda_1, \lambda_3) \leq dist(\lambda_1, \lambda_2) + dist(\lambda_2, \lambda_3)$

Furthermore I propose the following guidelines when designing a distance to which one shall pay attention for the defined distance or semi-distance to be useful and reliable:

- The distance shall evolves accordingly to what the user would logically expect
- The distance shall evolves smoothly
- The distance shall be sensitive to the variations of any parameters (in the case of the HMMs: the emission distributions parameters, the transition matrix, and the mixing coefficients)

In the specific case of the HMMs, and with respect to the fact that the data likelihood is often used as a decision/classification threshold, one shall also pay a special attention to how the distance behaves with respect to the KL divergence as defined by Juang and Rabiner in [147]:

$$D_{KL}(\lambda_1, \lambda_2) = \lim_{T\to\infty} \frac{1}{T}(\ln(p(O_T|\lambda_1)) - \ln(p(O_T|\lambda_2))) \ , \tag{84}$$

where O_T represents a time-series of T observations.

Dirichlet and generalized Dirichlet-based HMMs have only recently been proposed and applied to real-world situations. The learning equations of the former have been derived in [28] in 2007 and applied for the first time on a real-world data set for texture classification, action recognition, and anomaly detection in this thesis. The learning equations of the latter have been derived as part of this thesis work. To the best of my knowledge, no work on distances between these models has been done so far and this is the first comparative study for parameters-base distances for these models.

My contributions are the following, (1) the replication of the results of [145] with the addition of a third inner distance, the Probability Product Kernel [148] and of the results of [146] over Gaussian-based HMMs for comparison (never studied before) and for highlighting their sensitivity limitations in Section 6.2 ; (2) the non-trivial extension of the distance proposed in [146] to the multidimensional case for the Dirichlet and the GD in Section 6.3 ; (3) the proposition of two variants of a new distance, robust to mixture shuffling and to component shuffling for HMMD and HMMGD in Section 6.4 ; and (4) a thorough study of the behavior of the aforementioned distances with respect to variations of all parameters and permutations of states and components, including pointing out at the strengths weaknesses of some state-of-the-art distances with respect to each other through multiple experiments with synthetic data and over two real-world data set, showing the reliability of the new proposed distances in Section 6.5.

The overall goal of this comparative study is to give the option to anyone working with these models to choose the distance or similarity measure fitting their needs the most and to know what to expect from each one of them, as well as the influence of the tuning parameters when there are some. This opens up possibilities for designing distance-based algorithms in the HMM space such as hierarchical clustering (see Section 6.5.2), nearest neighbors methods, etc.

The work presented in this chapter is being submitted in a journal in the field under the title *Data-free metrics for Dirichlet and generalized Dirichlet mixture-based HMMs - A practical study.*

6.2 Preliminary results and problem setting

The motivation for the design of new parametric distances comes from the following preliminary work in which I re-implement and test the methods proposed in [145], while adding the Probability Product Kernel (PPK) from [148] as a distance measure between distributions in their framework and study the influence of the variation of each parameter in order to highlight an important limitation. I refer the reader to the original paper for the implementation details but recall the main steps here: a correspondence between the states is obtained from a similarity measure between the emission distributions of the HMMs. In the case of mixture-based HMMs, only the KL divergence is proposed in the form of its inverse or in the form of the inverse of its exponential multiplied by a factor κ. A sparcity score over the correspondence matrix is computed as a reflection of the similarity of the HMMs (the scarcer the matrix is, the more similar the HMMs are).

Following their work, I use 2-dimensional Gaussian HMMs. The transition matrices are fixed at: $A_1 = A_2 = T_1 = [.6\ .4; .4\ .6]$ and the Gaussian means are set to $\mu_1 = [1\ 1; 3\ 3]$ and $\mu_2 = [1\ 3 - d; 3\ 1 + d]$, with d varying from 0 to 2. Finally, the covariance matrices are set to the identity for the first dimension and to $C_{1,2} = [1\ .3; .3\ 1]$ and $C_{2,2} = [1\ .1; .1\ 1]$ for the second dimension.

Figure 6.1 shows that, as expected, the similarity increases with d and that the PPK similarity measure can be used in this framework if transformed into a negative exponential form. This approach is thus sensitive to the variations of the distributions' parameters.

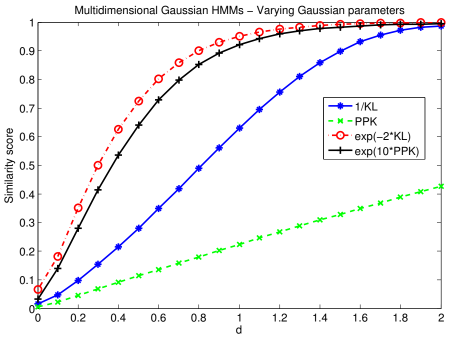


Figure 6.1: Varying Gaussian means with 2-dimensional Gaussian HMMs

Second, I study the sensitivity to the variations in the transition matrix while keeping the

Gaussian parameters similar (but slightly different to avoid divisions by 0). The parameters used are $A_1 = [.9 - d\ .1 + d; .9 - d\ .1 + d]$, $A_2 = T_2 = [.1\ .9; .1\ .9]$, $\mu_1 = [1\ 1; 3\ 3]$, and $\mu_2 = [1\ 1.1; 3\ 3.1]$. The variances are kept small and equal to 0.1 in order to have a clear difference between the components of the HMMs. I vary d from 0 to 0.8 and report the results in Figure 6.2.

Only the two PPK-based similarities give logical trends. This shows the method to be in general non-sensitive to changes in the transition matrix in the multidimensional Gaussian case. In [145], this sensitivity is only studied in the case of discrete HMMs and the related figure already showed a low sensitivity. An absence of sensitivity to changes in the transition matrix reduces HMMs to be seen as mixtures models, discarding their essential dynamic properties.

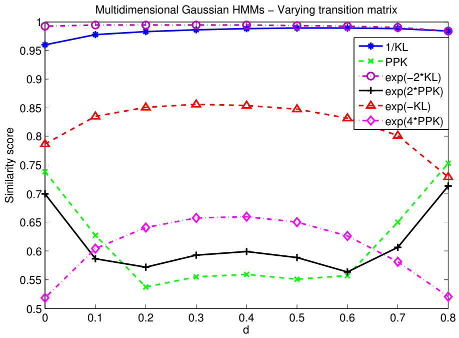


Figure 6.2: Varying transition matrices with 2-dimensional Gaussian HMMs

Additionally, I study the influence of coefficient κ on the computed distances by making it vary from 1 to 20 for the exponential forms of the approach (using the same parameters as the ones used for Figure 6.1). The results, in Figure 6.3, pinpoint a major flaw of the approach. The final similarity measure drastically varies, making the results non objective unless under a careful study of this coefficient's tuning.

With these results in mind, I study how the HSD approach [146] behaves compared to the previously tested methods. As already said, its main limitation resides in the fact that it only applies to unidimensional distributions. Its efficiency giving coherent distances when the Gaussian parameters are changed is clearly illustrated in the original paper and I only present the results for variations in the transition matrix. The parameters used are

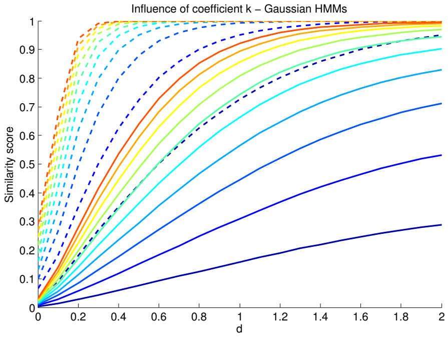


Figure 6.3: Varying κ with 2-dimensional Gaussian HMMs. Plain curves for PPK-based similarity and dashed curves for KL-based similarities. κ varies from 1 to 10, $\kappa = 1$ for the lowest curve of each network of curves.

$A_1 = [.9 - d \ .1 + d; .9 - d \ .1 + d]$, $A_2 = T_2$, $\mu_1 = \mu_2 = [1; 3]$, and the variances equal to 0.10 and 0.11. Here and in all subsequent graphs, I plot the HSD distance Δ as a similarity score by computing $exp(-\Delta)$, in order to be able to compare with the other approaches. In Figure 6.4, the HSD metric perfectly grasps the variations imposed to the transition matrix and, once again, the approach of [145], with whatever inner distance setting, does not achieve to grasp these variations.

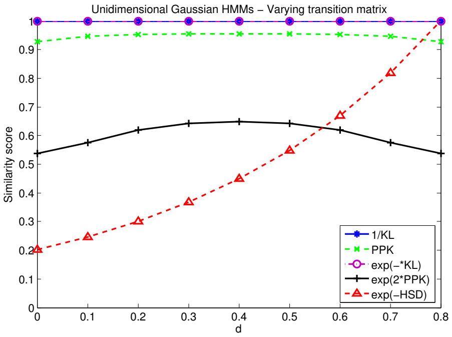


Figure 6.4: Varying the transition matrix for unidimensional Gaussian HMMs.

These results clearly show the need of designing new distances for multidimensional continuous HMMs that exhibit a sensitivity in changes of the distribution parameters, of the transition matrix, and of the mixing matrix. As most research is led on the Gaussian HMMs I shift the focus to HMMs designed for proportional data and relying on Dirichlet and generalized Dirichlet distributions which are the main topic of this thesis.

In the following, I extend the work of [146] to overcome the unidimensional limitation of the HSD distance for Dirichlet and the generalized Dirichlet using some of their natural mathematical properties. I also propose a distance based on several approximations of Kullback-Leibler divergences at the level of the distribution, the mixture, and the HMM. While many works make the assumption of mixtures composed of fixed components, and/or of HMM with ordered states, I add the steps to handle all sorts of permutation that can occur during the learning phase, ending up with the most robust parametric distance to the best of my knowledge.

6.3 Extension of the HSD distance

I recall that a D-dimensional Dirichlet distribution is expressed as

$$p(x|\alpha) = \frac{\Gamma(\sum_{d=1}^{D} \alpha_d)}{\prod_{d=1}^{D} \Gamma(\alpha_d)} \prod_{d=1}^{D} x_d^{\alpha_d - 1} \ , \tag{85}$$

with $\alpha = (\alpha_1, \ldots, \alpha_D)$, $\alpha_d > 0$, and $x = (x_1, \ldots, x_D)$, $\sum_{d=1}^{D} x_d = 1$. $\Gamma(t) = \int_0^\infty x^{t-1} e^{-x} dx$ is the Gamma function.

Similarly, a D-dimensional generalized Dirichlet distribution is expressed as

$$p(x|\alpha, \beta) = \prod_{d=1}^{D} \frac{\Gamma(\alpha_d + \beta_d)}{\Gamma(\alpha_d)\Gamma(\beta_d)} x_d^{\alpha_d - 1} \Big(1 - \sum_{r=1}^{d} x_r\Big)^{\nu_d} \ , \tag{86}$$

with $\alpha = (\alpha_1, \ldots, \alpha_D)$, $\alpha_d > 0$, $\beta = (\beta_1, \ldots, \beta_D)$, $\beta_d > 0$, and $x = (x_1, \ldots, x_D)$, $\sum_{d=1}^{D} x_d < 1$. ν_d is defined as $\nu_d = \beta_d - \alpha_{d+1} - \beta_{d+1}$ if $d \neq D$ and $\nu_D = \beta_D - 1$.

The limitation of the HSD distance to unidimensional distributions is due to the fact it relies on the computation of the cumulative distribution function (CDF) of the distributions composing the HMM. The concept of CDF is undefined for multidimensional distributions, hence the distance cannot apply to them.

However, as previously mentioned, the generalized Dirichlet distribution has the following property [60,61]:

Property 6.1: A D-dimensional generalized Dirichlet, $GD(\alpha_1, ..., \alpha_D, \beta_1, ..., \beta_D)$, is equivalent to a set of D independent Beta distributions with the same parameters $(\alpha_n, \beta_n), n =$

$1, \ldots, D$, in a particular transformed data space that is reached through a bijection. The bijective function linking the two data spaces is expressed as $W = \{W_n\}_{1:D}$ with:

$$W_n = \begin{cases} x_n \ , \quad \text{for } n = 1 \ , \\ \dfrac{x_n}{1 - \sum_{i=1}^{n-1} x_i} \ , \quad \text{for } n \in [2, D] \ . \end{cases} \tag{87}$$

Beta distributions, are unidimensional by definition and their CDF is easily computable. I can then make up a simple function that acts as an equivalent of the CDF for multidimensional generalized Dirichlet distributions and keep the rest of the distance computation untouched.

When working with the Dirichlet distribution, another transform is first required to express it into a generalized Dirichlet form. Indeed, the Dirichlet is a degenerate case of generalized Dirichlet [61].

Property 6.2: A D-dimensional generalized Dirichlet $GD(\alpha_1, ..., \alpha_D, \beta_1, ..., \beta_D)$, which parameters verify $\beta_n = \alpha_{n+1} + \beta_{n+1}$, for $n = 1, \ldots, (D-1)$, is a Dirichlet distribution with parameters $Dir(\alpha_1, \ldots, \alpha_D, \beta_D)$.

Reversing this expression allows to express a Dirichlet distribution in the form of a generalized Dirichlet one and thus to apply an extended form of the HSD distance computation to it.

In summary, Beta distributions are used to characterize the HMM in a transformed data space and the HSD measure can be deployed using them. The resulting distance is equivalent to the distance that could have been computed in the initial space as these two spaces are connected through a bijection.

The computation of the HSD distance for multidimensional Dirichlet and GD distribution-based HMMs follows the steps:

1. For each state of each HMM, express the Dirichlet distributions in their GD form [61]: $Dir(\alpha_1, ..., \alpha_{D+1}) \equiv GD(\alpha_1, ..., \alpha_D, \beta_1, ..., \beta_D)$, with $\beta_j = \alpha_{j+1} + \beta_{j+1}$ for $j = 1, \ldots, (D-1)$ and $\beta_D = \alpha_{D+1}$

2. Initialize the distance Δ and the value x to 0, and the step size to $s = 1/L$ (hereafter, $L = 100$)

3. Iteratively do L times the following steps:

    (a) For each state k, dimension d, and HMMs $i = 1, 2$, compute $\text{BetaCDF}_{i,k,d}(\alpha_{i,k,d}, \beta_{i,k,d}, x)$

    (b) For each state k of each HMM i, compute
    $\text{CDF}_{i,k} = \sum_{d=1}^{D} \text{BetaCDF}_{i,k,d}$

    (c) Compute the models' CDFs using a dot product $F_i = \langle \Pi_{s,i}, \text{CDF}_i \rangle$

    (d) Compute $\Delta = \Delta + s \times |F_1(x) - F_2(x)|$

    (e) Increment x by s

When the models are based on GD distributions, the first step is obviously omitted. Experimental results for this distance are reported in Section 6.5.

6.4 Proposed distance

6.4.1 General case

I propose to derive a parametric distance for HMMD and HMMGD under the assumption that mixtures are indivisible elements. This means that either these mixtures have some physical representation and that their components cannot be split up over different states, or that the components found while initializing the HMM have been ordered following some heuristic rules. The computation of this distance needs to take into account the potential permutation of the mixtures over the different states.

As I intend to compute a parameter-based distance similar in behavior to the Kullback-Leibler distance, I start from its definition for a given function f:

$$D(f_1||f_2) = \int f_1 \ln\left(\frac{f_1}{f_2}\right) . \tag{88}$$

When data samples $X = x_1, \ldots, x_T$ are available, a Monte-Carlo approximation of

Equation (88) gives:

$$D(f_1||f_2) \approx \frac{1}{T}\sum_{t=1}^{T}(\ln(f_1(x_t)) - \ln(f_2(x_t))) \ . \tag{89}$$

For this approximation to be accurate, T needs to be large enough. In the case of HMMs, f_1 and f_2 can be identified as the likelihood of the data with respect to the HMMs λ_1 and λ_2, respectively. As T increases, the computation of these quantities becomes heavier and at some point, even prohibitive (see Section 6.5.1 for more details).

[143] devised a method to approximate an upper bound to the Kullback-Leibler divergence for Dependence Trees and showed that it can be used for left-to-right HMMs that can be considered as a special case of dependence trees. I start from the approximation proposed as:

$$D(\lambda_1||\lambda_2) \leq \sum_{k=1}^{K} \pi'_{k_1}(D(a_j||\tilde{a}_j) + D(b_j||\tilde{b}_j)) \ , \tag{90}$$

where π' is the stationary distribution of λ_1.

The stationary distribution of an HMM is iteratively computed as proposed in [146] following the recursive equation:

$$\pi'_{t+1} = \pi'_t A \ , \tag{91}$$

starting from the initial state probability mass function $\pi'_0 = \pi$.

Using Equation (90) implies that the distance does not take into account the transitional phase of the HMM. However, my experiments show that even for HMMs trained on short sequences, the distance behaves as expected and gives good discriminative results (see Section 6.5.1).

The only experiments carried out in [143] on pure HMMs are with a simple discrete HMM (with pre-defined parameters), two states and data of 3 dimensions. Therefore, more extensive experiments with a similarly designed method are needed to assess the potential discriminative performance of such parameter-based approximation of the KL divergence.

In Equation (90), the term $D(a_j||\tilde{a}_j)$ refers to the rows of the transition matrices. Each row of a transition matrix is a probability mass function and therefore the KL divergence can be easily computed. However, given that the models I am working with do not have a left-to-right topology, I first need to pair up the states of the two models. I propose to see this task as a linear assignment problem and solve it using the Jonker-Volgenant algorithm [149], which provides a faster implementation of the well-known Hungarian algorithm. The Jonker-Volgenant algorithm provides a cost matrix for pairing up each state of λ_1 with each state of λ_2, as well as the sequence of pairs that minimizes the assignment cost. From this sequence of pairs, I build a permutation matrix $\mathcal{R} = r_{i,j}$, where $r_{i,j} = 1$ if state i of λ_1 is optimally matched to state j of λ_2 and 0 otherwise. The transition matrix of the HMM λ_2 is then permuted as $\tilde{A}' = \mathcal{R}\tilde{A}\mathcal{R}$. The mixtures assigned to each state are permuted accordingly.

The second term of Equation (90), $D(b_j||\tilde{b}_j)$ refers to the emission probability distributions assigned to each state which are, in my case, mixtures. The KL divergence of mixture models does not have a closed form expression and then requires to be approximated. Hershey and Olsen [9] proposed a full review of techniques to approximate the KL divergence between two mixtures of Gaussian. Studying the assumptions made, most of the approximations they proposed can be applied to mixtures of Dirichlet and generalized Dirichlet without restriction. The variational approximation they proposed is chosen here for the good results it showed for the Gaussian case in [9], especially as the criterion used in that study is the similarity to the classic data-based KL divergence estimation, which is also one of my criteria for the design of this HMM distance.

Denoting the mixtures as $P_1 = \sum_{m=1}^{M} w_{1,m} p_{1,m}$ and $P_2 = \sum_{m=1}^{M} w_{2,m} p_{2,m}$ The variational approximation is written as:

$$D(P_1||P_2) = \sum_{m=1}^{M} w_{1,m} \frac{\sum_{a=1}^{M} w_{1,a} e^{-D(p_{1,m}||p_{1,a})}}{\sum_{b=1}^{M} w_{2,b} e^{-D(p_{1,m}||p_{2,b})}} \,. \tag{92}$$

The latter equation involves the computation of the KL divergence between two Dirichlet (and generalized Dirichlet) distributions. The KL divergence between two D-dimensional

Dirichlet distributions $Dir_1(\vec{\alpha_1})$ and $Dir_2(\vec{\alpha_2})$ can be expressed as:

$$
\begin{aligned}
KL(Dir_1||Dir_2) = & \ln\left(\Gamma\Big(\sum_{d=1}^{D}\alpha_{1,d}\Big)\right) - \sum_{d=1}^{D}\ln(\Gamma(\alpha_{1,d})) - \ln\left(\Gamma\Big(\sum_{d=1}^{D}\alpha_{2,d}\Big)\right) \\
& + \sum_{d=1}^{D}\ln(\Gamma(\alpha_{2,d})) + \sum_{d=1}^{D}(\alpha_{1,d}-\alpha_{2,d})\Psi\left(\alpha_{1,d}-\Psi\Big(\sum_{j=1}^{D}\alpha_{1,j}\Big)\right) ,
\end{aligned} \tag{93}
$$

and the KL divergence between two D-dimensional generalized Dirichlet distributions $GD_1(\vec{\alpha_1}, \vec{\beta_1})$ and $GD_2(\vec{\alpha_2}, \vec{\beta_2})$ is expressed as [150]:

$$
\begin{aligned}
KL_{GD}(p||q) = & \sum_{d=1}^{D}\ln\left(\frac{\Gamma(\alpha_{1,d}+\beta_{1,d})\Gamma(\alpha_{2,d})\Gamma(\beta_{2,d})}{\Gamma(\alpha_{1,d})\Gamma(\beta_{1,d})\Gamma(\alpha_{2,d}+\beta_{2,d})}\right) \\
& - \sum_{d=1}^{D}(\alpha_{1,d}-\alpha_{2,d})\Big(\Psi(\alpha_{1,d}) - \Psi(\beta_{1,d}) - \sum_{s=1}^{d}(\Psi(\alpha_{1,s}+\beta_{1,s}) - \Psi(\beta_{1,s}))\Big) \\
& + \sum_{d=1}^{D}(\nu_{1,d}-\nu_{2,d})\sum_{s=1}^{d}(\Psi(\alpha_{1,s}+\beta_{1,s}) - \Psi(\beta_{1,s})) \, .
\end{aligned} \tag{94}
$$

The steps of the KL divergences computation are given in Appendices B and C, respectively.

The set of Equations (90) to (94), allows to compute a distance between two Dirichlet or generalized Dirichlet-based HMM without the need for generating data of any kind. In Section 6.5.1, I show how well this distance performs on HMMs with randomly generated parameters, even when the HMM states are permuted. However, when working with real-world data, in some cases, HMM are trained on abstract features extracted from the data prior to the training. Some sets of equations for learning the HMM model do not impose any constraint upon how the initial mixture components found in the data are assigned to the states [28]. In that case, the sole assumption of state permutation is not strong enough and would fail. Therefore, there is a need to design a simple method allowing for component permutation between mixture models. Such a method is presented in the next section.

6.4.2 Special case

HMMs based on mixtures of Dirichlet have been first introduced in [28] and served as a reference for the development of the HMMs based on mixtures of generalized Dirichlet in this thesis. The learning process requires initial values for all HMMs parameters, including the emission distributions. This initialization is based on a simple k-means clustering followed by a moment matching procedure. The estimated distributions are then grouped into mixtures depending on the chosen values for K and M. The k-means clustering has no constraint on the choice of the seeds, so does the grouping procedure and therefore, in general, HMMs trained from the same data will have different mixtures (i.e., mixtures composed of different components) assigned to different states. These HMMs are yet totally equivalent and will perform the same way, with equivalent accuracies in classification tasks.

In these cases, the approach devised in the previous section does not make sense as one of the assumptions made is not respected. In order to take into account all the possible permutations, another quantity needs to be defined that allows to find a distance close to 0 when HMMs are equivalent even if their parameters, at first look, are different. The *natural* KL divergence, achieves it by looking at the likelihood values directly.

In order to devise a new relevant quantity, I get inspired by this initialization process of the HMM learning algorithm that relies of a k-means clustering among $K * M$ clusters. As the subsequent grouping of components into mixture models impacts the values of the transition matrix, of the mixing matrix, and of the initial state probability mass function, I cannot rely on these parameters as is. In order to see how close two HMMs are, I need to somehow revert this process i.e., to combine these parameters in order to *decorrelate* them from the mixture models. The procedure can be illustrated with this question: What is the closest equivalent of a non-mixture HMM that I can get from this mixture-based HMM? Obviously this will be a loose equivalence and in no case a bijection. However, I propose here a quantity that I call the *flatten transition matrix* that is simple and efficient enough to compute discriminative distances as I show later on a small example using real-world data in Section 6.5.2.

Building the *flatten transition matrix* A' - This quantity reflects what the transition matrix of a K-state mixture-based HMM with mixture of M components *flatten* into a non-mixture HMM with $K*M$ component would be equivalent to. This approximation naturally depends on the transition matrix $A = \{a_{ij}\}_{K\times K}$ and the mixing matrix $C = \{c_{ij}\}_{K\times M}$ of the HMM. Given that I work under the assumption of stationary HMM, the initial state probability π is not involved. The *flatten transition matrix* is expressed as:

$$A'= \begin{bmatrix} a_{11}c_{11} & \dots & a_{11}c_{1M} & a_{12}c_{21} & \dots & a_{1K}c_{K1} & \dots & a_{1K}c_{KM} \\ \multicolumn{8}{c}{\text{repeat over (M-2) rows}} \\ a_{11}c_{11} & \dots & a_{11}c_{1M} & a_{12}c_{21} & \dots & a_{1K}c_{K1} & \dots & a_{K1}c_{KM} \\ a_{21}c_{11} & \dots & a_{21}c_{1M} & a_{22}c_{21} & \dots & a_{2K}c_{K1} & \dots & a_{2K}c_{KM} \\ \multicolumn{8}{c}{\text{repeat over (M-2) rows}} \\ a_{21}c_{11} & \dots & a_{21}c_{1M} & a_{22}c_{21} & \dots & a_{2K}c_{K1} & \dots & a_{2K}c_{KM} \\ \multicolumn{8}{c}{\vdots} \\ \multicolumn{8}{c}{\vdots} \\ a_{K1}c_{11} & \dots & a_{K1}c_{1M} & a_{K2}c_{21} & \dots & a_{KK}c_{K1} & \dots & a_{KK}c_{KM} \\ \multicolumn{8}{c}{\text{repeat over (M-2) rows}} \\ a_{K1}c_{11} & \dots & a_{K1}c_{1M} & a_{K2}c_{21} & \dots & a_{KK}c_{K1} & \dots & a_{KK}c_{KM} \end{bmatrix}. \tag{95}$$

The repetition of lines is due to the fact the transition matrix of mixtures-based HMMs only depends on the previous hidden state and not of the mixture component by which the observation is actually modeled. Therefore, even though I keep a square $KM \times KM$ matrix to match the shape of an HMM transition matrix, there are actually only K^2M different coefficients. All the rows sum up to one and thus A' is a valid transition matrix.

There is no need for a mixing matrix C' as no mixture are then involved, and an extended π' initial pmf is computed as follows:

$$\pi' = (\pi_{11}c_{11}, \dots, \pi_{11}c_{1M}, \pi_{12}c_{21}, \dots, \pi_{1K}c_{K1}, \dots, \pi_{1K}c_{KM}) \tag{96}$$

I now approximated a non-mixture HMM version of the original HMM. The single distributions (mixture components) are assigned accordingly to the way A' is constructed.

I can now apply the approach devised in the previous section to HMMs *flatten* this way, by directly applying the linear assignment matching algorithm at the component level (which are now the states of the *flatten* version of the HMM).

6.5 Experiments

6.5.1 Practical study on synthetic data

In order to lead a comparative study of the different distances I design two types of experiments. For the distance designed in Section 6.4, in which the assumption of mixture having a meaningful representation and thus being composed of components always grouped together, I carry out an extensive series of experiment over randomly generated HMMs making each set of parameters vary independently of the others. I also present quantities that are meaningful for comparing distances. Indeed, when working in a space where no *natural* physical distance exist but only artificially designed ones, which reference to use to compare how well is a distance doing? It mostly depends on the expectations of the one who uses it. For this reason, the behavior of the distance has to be characterized under different aspects.

I propose the following quantities to this purpose:

- The correlation to the parameters average variation which gives an idea of how the evolution of the distance follows the evolution of the individual parameters.

- The autocorrelation at lag 1 for a continuous variation of the parameters: Gives a measure of the smoothness of the distance function with respect to the evolution of the parameters. In the case of two models whose parameters continuously go further away to each other, a coefficient close to 1 means a very smooth function, -1 means an irregular/non-monotonic function which is not desirable.

- The average variation by unitary variation (for a variation of parameter d equal to 1) of the parameters. This gives an idea of how discriminative the measure is.

- The correlation to the KL divergence computed from generated data. This illustrates how the behavior of the parameter-based distances is compared to the reference data-based one, especially in terms of stability.

- The average distance to the KL divergence computed from generated data. This illustrates how the behavior of the parameter-based distances is compared to the reference data-based one, especially in terms of discriminability.

Among them, as the data-based KL divergence has some limitations, the points 1 to 3 are found to be the more reliable way of comparing distances. When the correlation of the data-based KL divergence to the parameters variation is not strong, points 4 and 5 are not relevant anymore.

As some works define similarities and not distances, the proposed distances are evaluated as similarities by taking the inverse of the exponential i.e., e^{-dist}. The data-based KL divergence is computed by generated a sequence of data of length $T = 100$ from the reference HMM. The value of T and its limitations in the case of the Dirichlet and generalized Dirichlet are discussed later.

In the following experiments, all parameters are randomly drawn from uniform distributions with Dirichlet parameters in the range $[0, 20]$. Therefore, the presented results are penalized by some occurrences or low discriminability between some components that do not occur in real scenarios (as the initial clustering would create a unique cluster for samples following this distribution). The HMM parameters are fixed to $K = 5$, $M = 2$, $D = 4$, these values are small enough to keep the component similarities occurrences low, and big enough to have some of the distances failing. In the following experiments, the sensitivity of the distances to the variation of each type of parameter is studied separately for a clear illustration of the strength and weaknesses of each of them.

Experiment 1 - Sensitivity to variations of the distribution parameters The parameters of the Dirichlet/GD distributions of one of the HMMs are varied by adding a constant d between 0 and 20 to the concentration parameters. I expect the similarity measures to start from 1 and rapidly decrease to 0 as the parameters variation is quite important, and the analysis, exponential. Tables 6.1 and 6.2 report the performance results of the approaches of [145], the proposed extension of the HSD distance, the data-based Kullback-Leibler divergence, and the proposed distance. Figures 6.5 and 6.6 show the results of a typical run of the experiment (each set of experiments is repeated 20 times at least).[1]

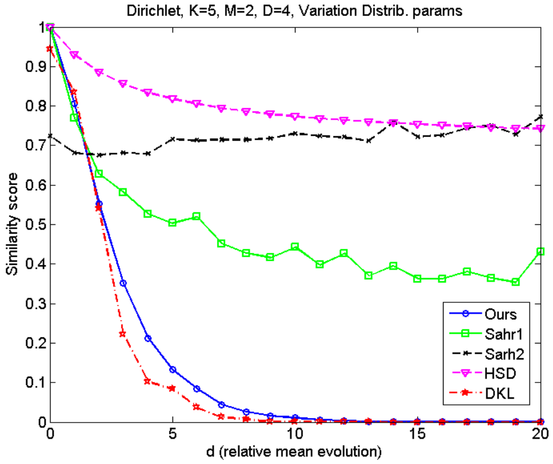


Figure 6.5: Varying the Dirichlet parameters between HMMs (typical run).

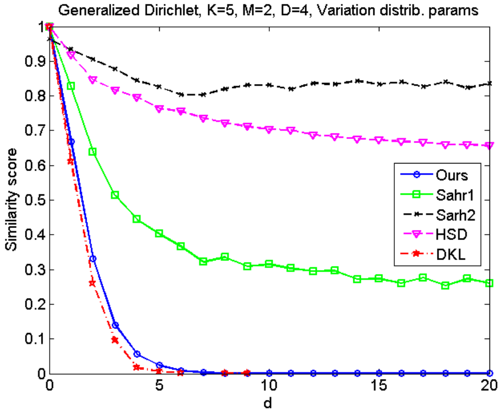


Figure 6.6: Varying the GD parameters between HMMs (typical run).

Besides the *Sahr2* similarity measure, all similarity measures are sensitive to distributions parameters variations. However, the extended HSD and the proposed similarity measure are smoother in their evolution, Though the HSD is more correlated to the variation of the parameters, its discriminative power is weak compared to the standard data-based KL-divergence and the proposed measure. These observations are valid for both the Dirichlet

[1] For all experiments the labels have to be read as follow: *DKL* is the data-based KL divergence. *Sahr1* and *Sahr2* are the methods of [145] with similarities computed as the inverse of the distance and the inverse exponential, respectively. *HSD* is the extended HSD distance presented in Section 6.3. *Ours* is the method proposed in Section 6.4.

Method	Corr params	Smooth	Amp var	Corr DKL	Avg dist DKL
DKL	-0.70	0.68	**-0.05**	1	0
Ours	-0.75	**0.72**	**-0.05**	0.99	0.06
HSD	**-0.86**	**0.72**	-0.01	0.95	0.19
Sahr1	-0.74	0.66	-0.04	0.97	0.12
Sahr2	-0.08	0.64	≤-0.01	0.48	0.17

Table 6.1: Comparative performance of distances for variation of the Dirichlet distributions parameters

Method	Corr params	Smooth	Amp var	Corr DKL	Avg dist DKL
DKL	-0.62	0.58	-0.04	1	0
Ours	-0.67	0.64	**-0.05**	0.95	0.06
HSD	**-0.90**	**0.75**	-0.02	0.86	0.18
Sahr1	-0.74	0.65	-0.04	0.94	0.13
Sahr2	-0.36	0.61	-0.01	0.80	0.19

Table 6.2: Comparative performance of distances for variation of the GD distributions parameters

and the GD cases. As the graphs of typical runs show, the proposed distance follows very well the evolution of the KL divergence while being deterministic and not relying upon any data.

Experiment 2 - Sensitivity to variations of the transition matrix Randomly drawing transition matrices T_1 and T_2, I make the transition matrix of the second HMM vary from T_1 to T_2, while the transition matrix of the first HMM remains equal to T_1. Therefore the transition of the second HMM is computed as $T_2^d = dT_2 + (1-d)T_1$. I expect the similarity measures to start from 1 and decrease as the transition matrices become less similar. Tables 6.3 and 6.4 report the performance results in the same manner as in Experiment 1. Figures 6.7 and 6.8 show the results of a typical run of the experiment. One should note that as the mixtures of distributions are perfectly equal, the inverse-based similarity measure of [145] is undefined.

Method	Corr params	Smooth	Amp var	Corr DKL	Avg dist DKL
DKL	-0.27	-0.03	-0.03	1	0
Ours	≤**-0.99**	**0.73**	**-0.28**	0.26	0.04
HSD	≤**-0.99**	**0.73**	-0.04	0.28	0.03
Sahr2	-0.21	0.08	≤-0.01	≥0.01	0.09

Table 6.3: Comparative performance of distances for variation of the transition matrices for Dirichlet-HMMs

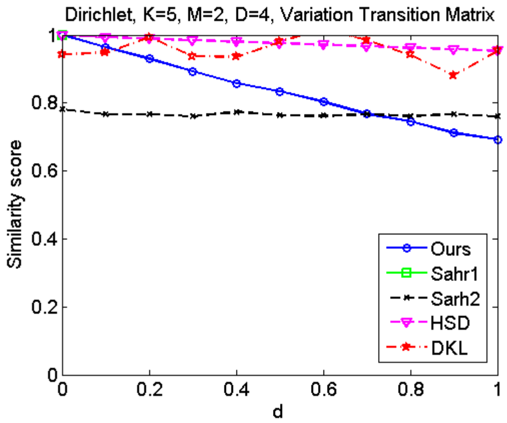


Figure 6.7: Varying the transition matrices between Dirichlet-HMMs (typical run).

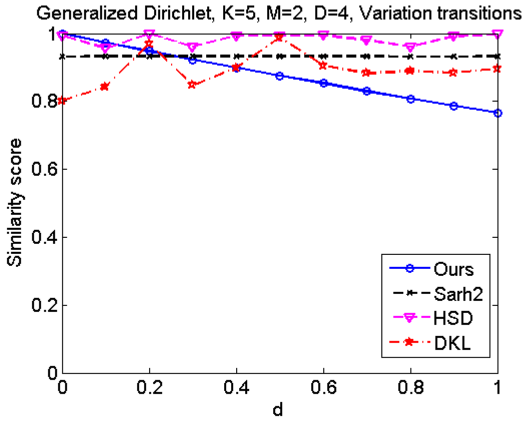


Figure 6.8: Varying the transition matrices between GD-HMMs (typical run).

Variations in the transition matrices are more subtle than variations within the distribution parameters. Indeed, it only impacts the way the time-series are ordered, not their potential values. The DKL and *Sahr2* similarity measures completely fail at detecting the slow drift of one HMM with respect to the other. DKL could potentially detect it using a bigger T value. However, as said earlier, this provokes overflow and make the distance slow to compute. This makes it an unreliable distance to work with unless fine tuning of T is

Method	Corr params	Smooth	Amp var	Corr DKL	Avg dist DKL
DKL	-0.21	-0.06	-0.02	1	0
Ours	$\leq$**-0.99**	**0.73**	**-0.27**	0.20	0.15
HSD	-0.30	-0.05	-0.02	0.45	0.34
Sahr2	-0.04	0.04	0.00	-0.01	0.27

Table 6.4: Comparative performance of distances for variation of the transition matrices for GD-HMMs

studied and a solution to overflow found (a simple scaling not solving the issue as, other distribution then reach the machine precision and set most results to 0).

Both the extended HSD and the newly proposed distance perform well in the Dirichlet case, being well correlated with the transition matrix variation and smooth. However the HSD is far less discriminative than the proposed distance. In the case of the GD, it fails and the proposed distance seems to be the only reliable option.

Experiment 3 - Sensitivity to variations of the mixing matrix Randomly drawing mixing matrices R_1 and R_2, I make the mixing matrix of the second HMM vary from R_1 to R_2, while the mixing matrix of the first HMM remains equal to R_1. Therefore, the mixing matrix of the second HMM can be computed as $R_2^d = dR_2 + (1-d)R_1$. I expect the similarity measures to start from 1 and decrease as the transition matrices become less similar. Tables 6.5 and 6.6 report the performance results in the same manner as in Experiment 1 and 2. Figures 6.9 and 6.10 show the results of a typical run of the experiment.

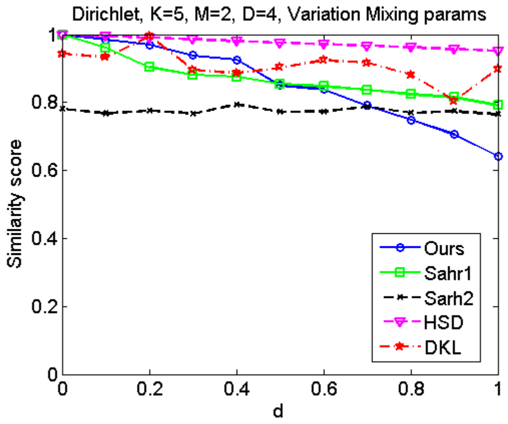


Figure 6.9: Varying the mixing matrices between Dirichlet-HMMs (typical run).

Variations of the mixing coefficients have a similar action on the generated data as a variation of the transition coefficients: it only impacts the way the time-series are ordered but not their values. It is therefore not surprising to see that the proposed approach allows good discrimination, good smoothness, and good correlation with the variation of the mixing coefficients. The extended HSD approach is valid here again in the Dirichlet case only but

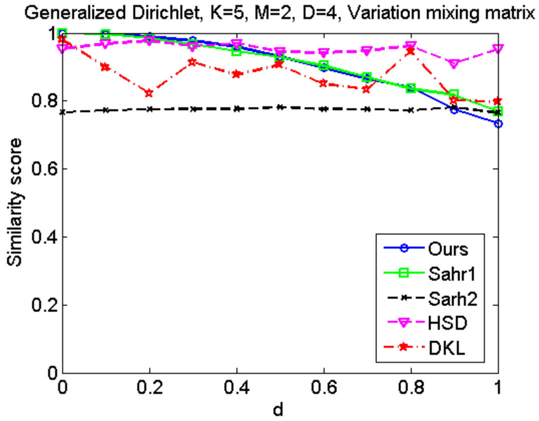


Figure 6.10: Varying the mixing matrices between GD-HMMs (typical run).

Method	Corr params	Smooth	Amp var	Corr DKL	Avg dist DKL
DKL	-0.57	0.16	-0.11	1	0
Ours	**-0.97**	**0.71**	**-0.20**	0.59	0.10
HSD	$\leq$**-0.99**	**0.73**	-0.05	0.57	0.12
Sahr1	**-0.98**	**0.72**	**-0.16**	0.58	0.11
Sahr2	-0.45	0.49	-0.02	0.27	0.07

Table 6.5: Comparative performance of distances for variation of the mixing matrices for Dirichlet-HMMs

with a weak discriminative potential. The *Sahr1* similarity measure works surprisingly well with just a bit less discriminative power than my proposed approach. However, it still relies on the tuning of the κ parameter which is not straightforward.

Overall, only the proposed approach shows itself successful to detect and logically reflect any kind of variation in the HMM model based on either Dirichlet or generalized Dirichlet, without requiring any data not any parameter tuning. The proposed extension of the HSD also reflects well the changes for Dirichlet-based HMMs but does not perform equally in the generalized Dirichlet case when the transition of mixing coefficients vary. Its discriminative power is lower which can also be the reason why it cannot achieve good performance when

Method	Corr params	Smooth	Amp var	Corr DKL	Avg dist DKL
DKL	-0.55	0.18	**-0.23**	1	0
Ours	**-0.97**	**0.71**	**-0.27**	0.57	0.13
HSD	-0.64	0.19	-0.05	0.69	0.14
Sahr1	**-0.98**	**0.72**	-0.14	0.59	0.14
Sahr2	-0.43	0.51	-0.01	0.35	0.48

Table 6.6: Comparative performance of distances for variation of the mixing matrices for GD-HMMs

minor parameters of the HMMs vary. The discriminative power of this distance could be enhance by adding a multiplicative coefficient when computing the approximate CDF while making the distance performance dependent of the tuning of that new parameter.

6.5.2 Illustration with real data

The extension of the method suits more HMMs that are trained as described in Chapters 3 and 4, using a component-by-component, k-means based initialization. As no bijective transformation is known between mixture-based HMMs, experiments validating my approach for the case when all components are assigned to different states are not possible with synthetic data.

I present hereafter, usages of the metric through clustering operations. Besides showing that the proposed metrics behaves in a logical way, I attempt to show the kind of information can be unraveled by its use. HMMs represent abstract features in a very abstract way. Therefore, the use of a ground truth of any sort to assess some clustering performance is not possible and the metric should rather be considered as a tool for exploring the data representation through HMMs.

A main constraint for clearly illustrating the proposed distance measure behavior is that, as just said, HMMs seldom represent something concrete that are itself measurable by a distance. Images appear to be a good way of getting some visual assessment of the performance. Therefore, I study the behavior of the designed distance with respect to HMMs trained over the UCSD Ped1 and Ped2 data sets, following the method presented in Chapter 3. In this study, the video sequences of the data sets that represent pedestrian walking on a university campus, are divided into 3D volumes. As the camera capturing the sequence is still, each volume represents a fixed spatial area of the campus i.e., grass, trees, walkway with pedestrians or a combination of two. An HMM is trained over each 3D volume location thus, I expect my designed distance to show high similarity between HMMs trained at similar locations (e.g., volumes representing the vegetation) and lower similarity between HMMs representing volumes featuring vegetation versus the busy walkway for example. In this application I fixed $K = 3$, $M = 2$, and $D = 12$ in coherence with Chapters 3 and 4.

Figure 6.11: Camera field for the UCSD Ped1 (left) and Ped2 (right) data sets

However, working with real-data requires a few adjustments. First of all, for the Dirichlet case, the parameters resulting from a training algorithm are oftentimes very high because of the variance which is badly estimated. In order to counter this artifact involved by some training methods, I use the mean of the Dirichlet (which is the normalized concentration vector) and rescale it in the range $[0, 20]$. [2]

After dividing the space into 77 overlapping patches (50% overlap) and training one HMM per location, I propose to compute the distances between these HMMs to unravel major patterns. I apply hierarchical clustering using my proposed similarity measure. I report hereafter in Figures 6.12 and 6.13, the two and three main clusters found across the trained Dirichlet HMMs.

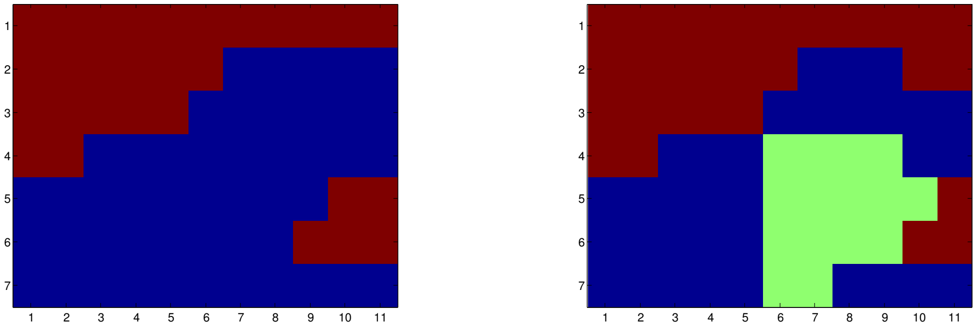


Figure 6.12: Two (left) and three (right) main clusters found in the UCSD Ped1 data set, Dirichlet-HMM case.

The proposed measure allows the clustering of the two main zones of the camera field (see Figure 6.11), the walkway versus the trees and grass where no dynamic action takes place. The clustering among three clusters seems to unravel the zones where less dynamic

[2] There is no risk of confusion with potential estimation of Dirichlet with parameters below 1, as Dirichlet distribution with such parameters exhibit several peaks on the "border" of the space they belong instead of a unique strong peak. The initial clustering performed for initializing the HMM naturally prevents this case to happen, as a distribution exhibiting two peaks would rather by approximate by two distributions with one peak each (minimizing the intra-class variance).

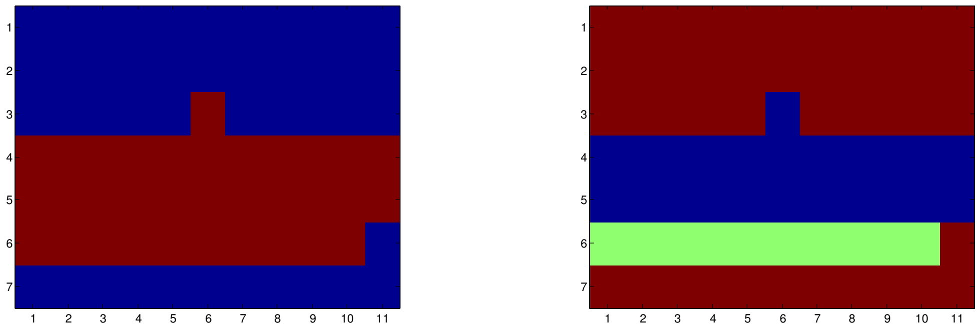


Figure 6.13: Two (left) and three (right) main clusters found in the UCSD Ped2 data set, Dirichlet-HMM case.

actions take place in the Ped2 data set. The meaning of the third cluster in the Ped1 data set is less clear. Looking at more cluster makes appear clusters whose visual meaning is not obvious. However, there similarity could allow approaches based on contextual information to refine the contextual areas.

Figures 6.14 and 6.15 report some clustering results over the same data sets using a GD-based HMM. One can see that the clustering results are somehow different on Ped1 but still make sense as the front view reduces the movements amplitudes that are far from it (Figure 6.14). It tends to show once more that my approach is more sensitive to movement than appearance. On the Ped2 data set, results similar to the Dirichlet case are found (Figure 6.14). However, interestingly, on some runs, I am able to detect, by increasing the number of cluster, a patch of the frame for which the HMM seems wrongly estimated (Figure 6.15), as it shows a single patch in a cluster. Such information, could be used to avoid using the corresponding HMM, and relying more on the neighboring ones as it is more willing to raise false positives or to miss some anomalies.

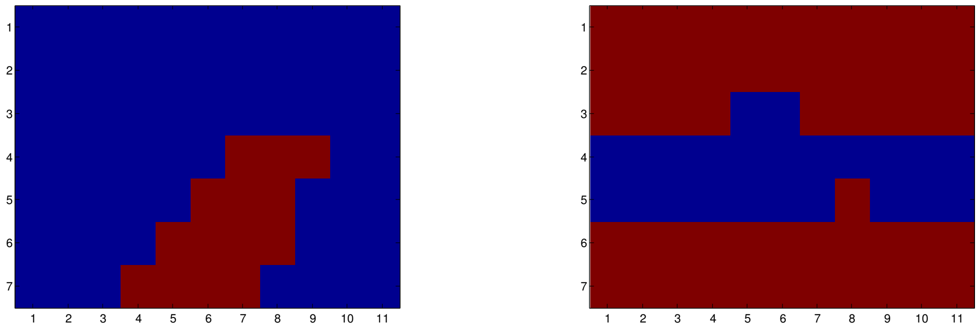


Figure 6.14: Two main clusters found in the UCSD Ped1 (left) and Ped2 (right) data sets, GD-HMM case.

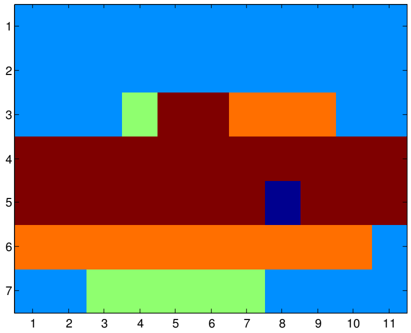


Figure 6.15: Five main clusters found in the UCSD Ped2 data set, GD-HMM case. The patch located at row 5 and column 8 seems erroneous.

6.6 Conclusion

In this chapter, I proposed the first parametric distances for the Dirichlet and generalized Dirichlet-based HMMs and, by extension, the Beta-based HMMs. I overcame the main limitation of the HSD distance proposed in [146] by extending it to the multidimensional case. Thought behaving as expecting for variations of the distributions and transition parameters, it failed at detecting changes in the mixing matrix. The approach I proposed, showed a great ability to detect any change of any HMM parameter, with good discriminative ability and without requiring any data. Its good correlation to parameters variation as well as its smoothness makes it the distance of choice for these models. By carrying out extensive experiments over synthetic data and providing a practical comparative performance of five similarity measures, a careful choice of measure that fits the expectations of the experimenter can be made. The extension of this distance for models trained by component is illustrated with real-data and shows coherent results and a potential for exploring the data representation through HMMs in order to detect erroneous estimations or to refine the concept of *neighbor* in some approaches using contextual information such as [52].

Chapter 7

Conclusion

The work presented in this thesis strove at adapting the broadly used HMMs to new data types with a focus on proportional data and to show their powerful capacity at modeling complex images and videos through applications for unusual event detection in videos and change detection in images. The study of two different learning approaches unraveled how considering the HMM parameters as random variables for estimation could substantially improve estimation precision hence, the models' performance. Wrongly estimated models can however still appear due to outliers or noise and the design of distances within the HMM space could be of great help for detecting them as well as for getting contextual information about the processed data. A lot of work remains to be done around these graphical models which, though a bit old and less used nowadays compared to extremely potent models such as deep neural networks, still have the advantage to be quickly trained, compact models that easily handle dynamical data, and that have a generative capacity.

In Chapter 2, showed that the model derived in [28] has good classifying capabilities and can be integrated in a framework for proportional data modeling. Moreover the work presented in this chapter proved that even very compact dictionaries in bag-of-words approaches could lead to good accuracies when classifying among couple of dozens of classes. The limitation in the dimensionality of data that can be modeled could however be an issue in some applications, and can be a subject of study in a future work.

In Chapter 3, I proposed the equations for the Baum-Welch learning algorithm for

HMM based on generalized Dirichlet and Beta-Liouville mixtures as emission distributions. Furthermore, application on synthetic and real data tends to show superior performance of these new models compared to the Dirichlet distribution. The full framework proposed for anomaly detection gives state-of-the-art results on one major data set for unusual event detection in crowds while having an inference time close to real-time. Among the directions that shall be explored as future work on the topic are an online setting for these models to perform well under varying conditions such as season cycles, or day/night cycles for outside video surveillance activities. Also, on a more technical point, the initialization of the Beta-Liouville parameters shall also be studied and improved as using an initial Dirichlet expressed in the form of a Beta-Liouville is neither satisfying from a theoretical or an empirical point of view. As in general, estimation algorithms are very sensitive to their initialization, closer initial estimates would most probably lead to an enhance performance of the Beta-Liouville-based HMM.

Chapter 4 presented the major contribution of this thesis that is the derivation of the variational learning for Dirichlet and generalized Dirichlet based HMMs. By considering the HMMs parameters as random variables and by carefully choosing the prior distributions to use, I showed that these models could outperform the models trained with a typical Baum-Welch algorithm for similar applications. Future work should logically include the Beta-Liouville distribution as well as a broader range of topics of application.

It was a good surprise to find in Chapter 5 that hybrid HMMs could be design following very simple rules and give very encouraging results. The modeling of mixed data is still fairly understudied and simple benchmarking methods are not easily found in the mainstream programming toolboxes of most languages. Studies like the one presented here helps at giving more tools for studying this complex data type and stop uniquely relying on the Gaussian assumption in such cases. The design of a unification framework between multi-stream and hybrid HMMs would be of great interest and could help integrating methods (such as the weighting methods) from that former field to the latter one. On a more technical note, a deeper study of the multivariate Gamma distribution in the HMM framework would be interesting as no consensus seems to exist for this distribution definition yet, is used as an hypothesis for modeling the noise corrupting some radar images (speckle noise).

Finally in Chapter 6, I proposed several parametric distances between HMMs based on the Dirichlet and the generalized Dirichlet distributions. Quantities for characterizing such distances are proposed and their behavior with respect to the gradual change of all HMM parameters are studied. Parametric distances for HMMs of all types are still seldom. However, some studies have shown how contextual information in images and videos could sometimes help reducing the false alarm rate, or detecting errors in the model estimation. The design of basic and more sophisticated distances between HMMs should help at designing more methods in this trend.

My hope is that the work realized in this thesis has shown the importance of studying the type of data to model before blindly modeling it following a Gaussian assumption. Though a Gaussian assumption may lead to good results, the use of appropriate tools/models will in general improve the performance of the approach. Using simple rules, one can derive tools around these adapted models for further comparing them to each other, or mixing them up to model even more sophisticated types of data.

Bibliography

[1] J. Prendes, M. Chabert, F. Pascal, A. Giros, and J.-Y. Tourneret. A newmultivariate statistical model for change detection in images acquired by homogeneous and heterogeneous sensors. *IEEE Trans. Image Process.*, 24(3):799–812, 2015.

[2] G. Mercier, G. Moser, and S. B. Serpico. Conditional copulas for change detection in heterogeneous remote sensing images. *IEEE Trans. Geosci. Remote Sens.*, 46(5):1428–1441, 2008.

[3] B. Antic and B. Ommer. Video parsing for abnormality detection. In *Computer Vision, IEEE International Conference on*, volume 0, pages 2415–2422, Los Alamitos, CA, USA, 2011. IEEE Computer Society.

[4] Big data infographic by ben walker. http://www.vcloudnews.com/every-day-big-data-statistics-2-5-quintillion-bytes-of-data-created-daily. Accessed: 2017-06-20.

[5] The top 20 valuable facebook statistics (may 2017). https://zephoria.com/top-15-valuable-facebook-statistics. Accessed: 2017-06-20.

[6] N. Bouguila. Count data modeling and classification using finite mixtures of distributions. *IEEE Transactions on Neural Networks*, 22(2):186–198, 2011.

[7] Y. Huang, K. B. Englehart, B. Hudgins, and A. D. C. Chan. A Gaussian mixture model based classification scheme for myoelectric control of powered upper limb prostheses. *IEEE Transactions on Biomedical Engineering*, 52(11):1801–1811, 2005.

[8] S. Bourouis, M. Al Mashrgy, and N. Bouguila. Bayesian learning of finite generalized inverted Dirichlet mixtures: Application to object classification and forgery detection. *Expert Syst. Appl.*, 41(5):2329–2336, 2014.

[9] J. R. Hershey and P. A. Olsen. Approximating the Kullback Leibler divergence between Gaussian mixture models. In *Acoust., Speech and Signal Proc., IEEE Int. Conf. on*, volume 4, pages 317–320. IEEE, 2007.

[10] M. A. T. Figueiredo and A. K. Jain. Unsupervised learning of finite mixture models. *IEEE Trans. Pattern Anal. Mach. Intel.*, 24(3):381–396, 2002.

[11] W. Fan, N. Bouguila, and D. Ziou. Variational learning for finite Dirichlet mixture models and applications. *IEEE Trans. Neural Netw. Learn. Syst.*, 23(5):762–774, 2012.

[12] W. Fan and N. Bouguila. Online variational learning of generalized Dirichlet mixture models with feature selection. *Neurocomputing*, 126:166–179, 2014.

[13] W. Fan and N. Bouguila. Online facial expression recognition based on finite Beta-Liouville mixture models. In *Computer and Robot Vision, International Conference on*, pages 37–44. IEEE, 2013.

[14] T. Bdiri and N. Bouguila. Learning inverted Dirichlet mixtures for positive data clustering. In *Rough Sets, Fuzzy Sets, Data Mining and Granular Computing, Proceedings of the 13th International Conference on (RSFDGrC'11), Moscow, Russia, June 25-27, 2011*, pages 265–272, 2011.

[15] J. C. Seabra, F. Ciompi, O. Pujol, J. Mauri, P. Radeva, and J. Sanches. Rayleigh mixture model for plaque characterization in intravascular ultrasound. *IEEE Transactions on Biomedical Engineering*, 58(5):1314–1323, 2011.

[16] E. E. Kuruoglu and J. Zerubia. Modeling SAR images with a generalization of the Rayleigh distribution. *IEEE Transactions on Image Processing*, 13(4):527–533, 2004.

[17] J. A. Hernandez and I. W. Phillips. Weibull mixture model to characterize end-to-end internet delay at coarse time-scales. *IEE Proceedings - Communications*, 153(2):295–304, 2006.

[18] T. M. Nguyen and Q. M. Jonathan Wu. Robust Student's-t mixture model with spatial constraints and its application in medical image segmentation. *IEEE Transactions on Medical Imaging*, 31(1):103–116, 2012.

[19] D. Karlis and E. Xekaliki. Mixed Poisson distributions. *International Statistical Review*, 73(1):35–58, 2005.

[20] O. Amayri and N. Bouguila. A Bayesian analysis of spherical pattern based on finite Langevin mixture. *Appl. Soft Comput.*, 38:373–383, 2016.

[21] T. Elguebaly and N. Bouguila. Simultaneous high-dimensional clustering and feature selection using asymmetric Gaussian mixture models. *Image Vision Comput.*, 34:27–41, 2015.

[22] F. B. Lung, M. H. Jaward, and J. Parkkinen. Spatio-temporal descriptor for abnormal human activity detection. In *Machine Vision Applications, 14th IAPR Int. Conf.*, pages 471–474. IEEE, 2015.

[23] E. L. Andrade, S. Blunsden, and R. B. Fisher. Hidden Markov models for optical flow analysis in crowds. In *Proc. ICPR*, pages 460–463, 2006.

[24] M. Bicego, U. Castellani, and V. Murino. A hidden Markov model approach for appearance-based 3D object recognition. *Pattern Recognition Letters*, 26(16):2588–2599, 2005.

[25] L. E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state Markov chains. *The Annals of Mathematical Statistics*, 37(6):1554–1563, 1966.

[26] L. R. Rabiner and B. H. Juang. An introduction to hidden Markov models. *IEEE ASSP Magazine*, 3(1):4–16, 1986.

[27] M. Cholewa and P. Glomb. Estimation of the number of states for gesture recognition with hidden Markov models based on the number of critical points in time sequence. *Pattern Recognition Letters*, 34(5):574–579, 2013.

[28] L. Chen, D. Barber, and J. M. Odobez. Dynamical Dirichlet mixture model. IDIAP-RR 02, IDIAP, 2007.

[29] E. Epaillard, N. Bouguila, and D. Ziou. Classifying textures with only 10 visual-words using hidden Markov models with dirichlet mixtures. In *Adaptive and Intelligent Systems - Third International Conference, ICAIS 2014, Bournemouth, UK, September 8-10, 2014. Proceedings*, pages 20–28, 2014.

[30] E. Epaillard and N. Bouguila. Hidden Markov models based on generalized Dirichlet mixtures for proportional data modeling. In *Artificial Neural Networks in Pattern Recognition - 6th IAPR TC3 International Workshop, ANNPR 2014, Montreal, QC, Canada, October 6-8, 2014. Proceedings*, pages 71–82, 2014.

[31] E. Epaillard and N. Bouguila. Proportional data modeling with hidden Markov models based on generalized Dirichlet and Beta-Liouville mixtures applied to anomaly detection in public areas. *Pattern Recognition*, 55:125–136, 2016.

[32] E. Epaillard and N. Bouguila. Variational Bayesian learning of generalized Dirichlet-based hidden Markov models applied to unusual events detection. *IEEE Transactions on Neural Networks and Learning Systems*, in revision.

[33] E. Epaillard and N. Bouguila. Hybrid hidden Markov model for mixed continuous/continuous and discrete/continuous data modeling. In *Multimed. Signal Proc., 17th IEEE Int. Workshop on*, pages 1–6. IEEE, 2015.

[34] B. van Ginneken, S. Katsuragawa, B. M. ter Haar Romeny, K. Doi, and M. A. Viergever. Automatic detection of abnormalities in chest radiographs using local texture analysis. *IEEE Transactions on Medical Imaging*, 21(2):139–149, 2002.

[35] M. Bertalmío, L. A. Vese, G. Sapiro, and S. Osher. Simultaneous structure and texture image inpainting. *IEEE Transactions on Image Processing*, 12(8):882–889, 2003.

[36] S. Xu, T. Fang, D. Li, and S. Wang. Object classification of aerial images with bag-of-visual-words. *IEEE Geoscience and Remote Sensing Letters*, 7(2):366–370, 2010.

[37] B. R. Povlow and S. M. Dunn. Texture classification using noncausal hidden Markov models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(10):1010–1014, 1995.

[38] G. Fan and X.-G. Xia. Wavelet-based texture analysis and synthesis using hidden Markov models. *IEEE Transactions on Circuits and Systems 1: Fundamental Theory and Applications*, 50(1):106–120, 2003.

[39] Q. Xu, J. Yang, and D. Siyi. Color texture analysis using the wavelet-based hidden Markov model. *Pattern Recognition Letters*, 26(11):1710–1719, 2005.

[40] E. O. T. Salles and L. L. Ling. Texture classification by means of HMM modeling of AM-FM features. In *Proc. ICIP*, pages 182–185, 2001.

[41] L. Liu, P. W. Fieguth, D. A. Clausi, and G. Kuang. Sorted random projections for robust rotation-invariant texture classification. *Pattern Recognition*, 45(6):2405–2418, 2012.

[42] L. Qin, Q. Zheng, S. Jiang, Q. Huang, and W. Gao. Unsupervised texture classification: Automatically discover and classify texture patterns. *Image and Vision Computing*, 26(5):647–656, 2008.

[43] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer Vision - Volume 2, Proc. of the Int. Conf. on*, pages 1150–1157. IEEE Computer Society, 1999.

[44] X. Yang and Y. Tian. Texture representations using subspace embeddings. *Pattern Recognition Letters*, 34(10):1130–1137, 2013.

[45] Y. Xu, H. Ji, and C. Fermüller. Viewpoint invariant texture description using fractal analysis. *International Journal of Computer Vision*, 83(1):85–100, 2009.

[46] S. Dubuisson. The computation of the Bhattacharyya distance between histograms without histograms. In *Proc. IPTA*, pages 373–378, 2010.

[47] N. Bouguila, D. Ziou, and J. Vaillancourt. Unsupervised learning of a finite mixture model based on the Dirichlet distribution and its application. *IEEE Transactions on Image Processing*, 13(11):1533–1543, 2004.

[48] N. Bouguila and D. Ziou. Using unsupervised learning of a finite Dirichlet mixture model to improve pattern recognition applications. *Pattern Recognition Letters*, 26(12):1916–1925, 2005.

[49] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1265–1278, 2005.

[50] Y. Sasaki. The truth of the F-measure. In *School of Computer Science (Ed.): University of Manchester*, 2007.

[51] M. Varma and A. Zisserman. A statistical approach to material classification using image patch exemplars. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):2032–2047, 2009.

[52] M. Bertini, A. D. Bimbo, and L. Seidenari. Multi-scale and real-time non-parametric approach for anomaly detection and localization. *Computer Vision and Image Understanding*, 116(3):320–329, 2012.

[53] F. Jiang, Y. Wu, and A. K. Katsaggelos. Abnormal event detection from surveillance video by dynamic hierarchical clustering. In *Proc. ICIP*, pages 145–148, 2007.

[54] W. Li, V. Mahadevan, and N. Vasconcelos. Anomaly detection and localization in crowded scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(1):18–32, 2014.

[55] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos. Anomaly detection in crowded scenes. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1975 –1981, 2010.

[56] A. Zaharescu and R. Wildes. Anomalous behaviour detection using spatiotemporal oriented energies, subset inclusion histogram comparison and event-driven processing. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *ECCV (1)*, volume 6311 of *Lecture Notes in Computer Science*, pages 563–576. Springer, 2010.

[57] B. C. Welsh and D. P. Farrington. Public area CCTV and crime prevention: An updated systematic review and meta-analysis. *Justice Quarterly*, 26(4):716–745, 2009.

[58] C. Norris. *A Review of the Increased Use of CCTV and Video-Surveillance for Crime Prevention Purposes in Europe.* EU Policy Department C - Citizens Rights and Constitutional Affairs - PE 419.588, 2009.

[59] V. Kaltsa, A. Briassouli, I. Kompatsiaris, L. J. Hadjileontiadis, and M. G. Strintzis. Swarm intelligence for detecting interesting events in crowded environments. *IEEE Trans. Image Process.*, 24(7):2153–2166, 2015.

[60] N. Bouguila and D. Ziou. High-dimensional unsupervised selection and estimation of a finite generalized Dirichlet mixture model based on minimum message length. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1716–1731, 2007.

[61] T.-T. Wong. Parameter estimation for generalized Dirichlet distributions from the sample estimates of the first and the second moments of random variables. *Computational Statistics and Data Analysis*, 54(7):1756–1765, 2010.

[62] A. Basharat, A. Gritai, and M. Shah. Learning object motion patterns for anomaly detection and improved object detection. In *CVPR*. IEEE Computer Society, 2008.

[63] M. T. Chan, A. Hoogs, J. Schmiederer, and M. Petersen. Detecting rare events in video using semantic primitives with HMM. In *ICPR (4)*, pages 150–154, 2004.

[64] C. Piciarelli, C. Micheloni, and G. L. Foresti. Trajectory-based anomalous event detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1544–1554, 2008.

[65] O. Arandjelovic. Contextually learnt detection of unusual motion-based behaviour in crowded public spaces. In *Computer and Information Sciences II, 26th Int. Symposium on Computer and Information Sciences, London, UK, 26-28 Sept. 2011*, pages 403–410, 2011.

[66] S. Ji, W. Xu, M. Yang, and K. Yu. 3D convolutional neural networks for human action recognition. In *Machine Learning, 27th Int. Conf.*, pages 495–502, Haifa, Israel, 2010. Omnipress.

[67] V. Reddy, C. Sanderson, and B. C. Lovell. Improved anomaly detection in crowded scenes via cell-based analysis of foreground speed, size and texture. In *Computer Vision and Pattern Recognition Workshops (CVPRW), Conference on*, pages 55–61, Colorado Springs, CO, USA, 2011. IEEE Computer Society.

[68] T. Li, H. Chang, M. Wang, B. Ni, R. Hong, and S. Yan. Crowded scene analysis: A survey. *IEEE Trans.Circuits Syst. Video Technol.*, 25(3):367–386, 2015.

[69] Z. Ghahramani and M. I. Jordan. Factorial hidden Markov models. *Machine Learning*, 29(2-3):245–273, 1997.

[70] P. M. Djuric and J.-H. Chun. An MCMC sampling approach to estimation of nonstationary hidden Markov models. *IEEE Transactions on Signal Processing*, 50(5):1113–1123, 2002.

[71] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[72] H. Hjalmarsson and B. Ninness. Fast, non-iterative estimation of hidden Markov models. In *Proc. IEEE Conf. Acoustics, Speech and Signal Process*, volume 4, pages 2253–2256. IEEE, 1998.

[73] S. Andersson and T. Ryden. Subspace estimation and prediction methods for hidden Markov models. *The Annals of Statistics*, 37(6B):4131–4152, 2009.

[74] H. Holzmann, A. Munk, M. Suster, and W. Zucchini. Hidden Markov models for circular and linear-circular time series. *Environmental and Ecological Statistics*, 13(3):325–347, 2006.

[75] A. Maruotti. Mixed hidden Markov models for longitudinal data: An overview. *International Statistical Review*, 79(3):427–454, 2011.

[76] A. Maruotti and R. Rocci. A mixed nonhomogeneous hidden Markov model for categorical data, with application to alcohol consumption. *Statist. Med.*, (31):871–886, 2012.

[77] S. P. Chatzis, D. I. Kosmopoulos, and T. A. Varvarigou. Robust sequential data modeling using an outlier tolerant hidden Markov model. *IEEE Trans. Pattern Anal. Mach. Intel.*, 31(9):1657–1669, 2009.

[78] S. P. Chatzis. Hidden Markov models with nonelliptically contoured state densities. *IEEE Trans. Pattern Anal. Mach. Intel.*, 32(12):2297–2304, 2010.

[79] K. L. Caballero, J. Barajas, and R. Akella. The generalized Dirichlet distribution in enhanced topic detection. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 773–782, New York, NY, USA, 2012. ACM.

[80] N. Bouguila. Deriving kernels from generalized Dirichlet mixture models and applications. *Information Processing and Management*, 49(1):123–137, 2013.

[81] W. Fan and N. Bouguila. Variational learning of finite Beta-Liouville mixture models using component splitting. In *Neural Networks, International Joint Conference on*, pages 1–8. IEEE, 2013.

[82] G. Ronning. Maximum-likelihood estimation of Dirichlet distribution. *Journal of Statistical Computation and Simulation*, 32:215–221, 1989.

[83] W.-Y. Chang, R. D. Gupta, and D. St. P. Richards. Structural properties of the generalized Dirichlet distributions. *Contemporary Mathematics*, 516:109–124, 2010.

[84] N. Bouguila. Hybrid generative/discriminative approaches for proportional data modeling and classification. *IEEE Transactions on Knowledge and Data Engineering*, 24(12):2184–2202, 2012.

[85] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, pages 1395–1402. IEEE Computer Society, 2005.

[86] R. Chaudhry, A. Ravichandran, G. D. Hager, and R. Vidal. Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *CVPR*, pages 1932–1939. IEEE, 2009.

[87] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

[88] H. Akaike. A new look at the statistical model identification. *IEEE Trans. Autom. Control*, 19(6):716–723, 1974.

[89] R. Mehran, A. Oyama, and M. Shah. Abnormal crowd behavior detection using social force model. In *CVPR*, pages 935–942. IEEE, 2009.

[90] A. Senior. *Protecting Privacy in Video Surveillance*. Springer-Verlag London Limited, 2009.

[91] T. C. Mack. Privacy and the surveillance explosion. *The Futurist*, 48(1), 2014.

[92] K.-W. Cheng, Y.-T. Chen, and W.-H. Fang. Gaussian process regression-based video anomaly detection and localization with hierarchical feature representation. *IEEE Trans. Image Process.*, 24(12):1288–1301, 2015.

[93] D. Singh and C. K. Mohan. Graph formulation of video activities for abnormal activity recognition. *Pattern Recognition*, 65:265–272, 2017.

[94] M. Beal. *Variational Algorithms for Approximate Bayesian Inference.* PhD thesis, Gatsby Computational Neuroscience Unit, University College London, 2003.

[95] D. J. C. Mackay. Ensemble learning for hidden Markov models. Technical report, Cavendish Laboratory, University of Cambridge, 1997.

[96] S. Ji, B. Krishnapuram, and L. Carin. Variational Bayes for continuous hidden Markov models and its application to active learning. *IEEE Trans. Pattern Anal. Mach. Intel.*, 28(4):522–532, 2006.

[97] S. P. Chatzis and D. I. Kosmopoulos. A variational Bayesian methodology for hidden Markov models utilizing Student's-t mixtures. *Pattern Recognition*, 44(2):295–306, 2011.

[98] Z. Ma and A. Leijon. Bayesian estimation of Beta mixture models with variational inference. *IEEE Trans. Pattern Anal. Mach. Intel.*, 33(11):2160–2173, 2011.

[99] W. Fan, H. Sallay, N. Bouguila, and S. Bourouis. Variational learning of hierarchical infinite generalized Dirichlet mixture models and applications. *Soft Computing*, 20(3):979–990, 2016.

[100] D. Barber and C. M. Bishop. Ensemble learning in Bayesian neural networks. In *Neural Networks and Machine Learning*, pages 215–237. Springer, 1998.

[101] H. Attias. A variational Bayes framework for graphical models. In *Advances in Neural Information Processing Systems 12*, pages 209–215. MIT Press, 2000.

[102] T. Tsesmelis, L. Christensen, P. Fihl, and T. B. Moeslund. Tamper detection for active surveillance systems. In *Advanced Video and Signal Based Surveillance, 10th IEEE Int. Conf.*, pages 57–62. IEEE, 2013.

[103] P. Diaconis and D. Ylvisaker. Conjugate priors for exponential families. *The Annals of Statistics*, 7(2):269–281, 1979.

[104] E. Castillo, A. S. Hadi, and C. Solares. Learning and updating of uncertainty in Dirichlet models. *Machine Learning*, 26(1):43–63, 1997.

[105] S. Zhou, W. Shen, D. Zeng, M. Fang, Y. Wei, and Z. Zhang. Spatial-temporal convolutional neural networks for anomaly detection and localization in crowded scenes. *Signal Processing: Image Communication*, 47:358–368, 2016.

[106] M. Sabokrou, M. Fathy, M. Hoseini, and R. Klette. Real-time anomaly detection and localization in crowded scenes. In *CVPR'15*. IEEE, 2015.

[107] Matlab code for jitter. http://www.mathworks.com/matlabcentral/fileexchange/10425-jitter. Accessed: 2017-06-08.

[108] J. M. Chambers, W. S. Cleveland, B. Kleiner, and P. A. Tukey. *Graphical Methods for Data Analysis*. Wadsworth, 1983.

[109] T. J. Hastie and D. Pregibon. *Statistical Models in S*. Wadsworth & Brooks/Cole, 1992.

[110] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[111] A. R. de Leon and K. Carriere-Chough. *Analysis of Mixed Data - Methods and Applications*. Taylor & Francis, 2012.

[112] W. J. Krzanowski. Distance between populations using mixed continuous and categorical variables. *Biometrika*, 70:235–243, 1983.

[113] M. J. McGeachie, H.-H. Chang, and S. T. Weiss. CGBayesNets: Conditional Gaussian Bayesian network learning and inference with mixed discrete and continuous data. *PLoS Comput. Biol.*, 10(6), 2014.

[114] R. P. Browne and P. D. McNicholas. Model-based clustering, classification, and discriminant analysis of data with mixed type. *J. Statist. Plann. Inference*, 142(11):2976–2984, 2012.

[115] A. Komarek and L. Komarkova. Clustering for multivariate continuous and discrete longitudinal data. *Ann. Appl. Stat.*, 7(1):177–200, 2013.

[116] C.-Y. Leung. Regularized classification for mixed continuous and categorical variables under across-location heteroscedasticity. *J. of Multivariate Anal.*, 93(2):358–374, 2005.

[117] M. Nunez, A. Villarroya, and J. M. Oller. Minimum distance probability discriminant analysis for mixed variables. *Biometrics*, 59(2):248–253, 2003.

[118] A. R. de Leon, A. Soob, and T. Williamson. Classification with discrete and continuous variables via general mixed-data models. *Journal of Applied Statistics*, 38(5):1021–1032, 2011.

[119] I. Olkin and R. F. Tate. Multivariate correlation models with mixed discrete and continuous variables. *Ann. Math. Stat.*, 32:448–465, 1961.

[120] Y.-N. Zhang, X.-M. Tong, X.-W. Zhang, J.-B. Zheng, J. Zhou, and S.-W. You. Pedestrian detection based on multi-modal cooperation. In *International Workshop on Multimedia Signal Processing, MMSP'08, October 8-10, 2008, Shangri-la Hotel, Cairns, Queensland, Australia*, pages 148–152. IEEE Signal Processing Society, 2008.

[121] X. Hu, Z. Zhang, Y. Duan, Y. Zhang, J. Zhu, and H. Long. LiDAR photogrammetry and its data organization. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3812:181–184, 2011.

[122] L. Gagnon and A. Jouan. Speckle filtering of SAR images - a comparative study between complex-wavelet-based and standard filters. In *SPIE Proc.*, volume 3169, pages 80–91, 1997.

[123] K. R. Joshi and R. S. Kamathe. SDI: new metric for quantification of speckle noise in ultrasound imaging. In *International Workshop on Multimedia Signal Processing, MMSP'08, October 8-10, 2008, Shangri-la Hotel, Cairns, Queensland, Australia*, pages 122–126. IEEE Signal Processing Society, 2008.

[124] K. R. Castleman. *Digital Image Processing.* Prentice Hall Press, Upper Saddle River, NJ, USA, 1996.

[125] Q. Li, Y. Li, J. Gao, B. Zhao, W. Fan, and J. Han. Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. In *International Conference on Management of Data, SIGMOD'14, Snowbird, UT, USA, June 22-27, 2014*, pages 1187–1198, 2014.

[126] O. Missaoui, H. Frigui, and P. Gader. Multi-stream continuous hidden Markov models with application to landmine detection. *EURASIP J. Adv. Sig. Proc.*, 2013.

[127] E. Kijak, G. Gravier, L. Oisel, and P. Gros. Audiovisual integration for tennis broadcast structuring. *Multimedia Tools Appl.*, 30(3):289–311, 2006.

[128] Y. L. Ng. Discriminative training of stream weights in a multi-stream hmm as a linear programming problem. Master's thesis, The Hong-Kong University of Science and Technology, 2008.

[129] S. Tamura, K. Iwano, and S. Furui. A stream-weight optimization method for multi-stream HMMs based on likelihood value normalization. In *Acoustics, Speech, and Signal Processing, Proceedings of the IEEE International Conference on (ICASSP'05), Philadelphia, PA, USA*. IEEE, 2005.

[130] M. Gurban, J.-P. Thiran, T. Drugman, and T. Dutoit. Dynamic modality weighting for multi-stream HMMs in audio-visual speech recognition. In *Multimodal Interfaces, Proceedings of the 10th International Conference on (ICMI'08), Chania, Crete, Greece*. ACM, 2008.

[131] ESA: Our missions. http://www.esa.int/ESA/Our_Missions. Accessed: 2017-07-04.

[132] The DLR website. http://www.dlr.de/eo/. Accessed: 2017-06-08.

[133] S. Kotz, N. Balakrishnan, and N. L. Johnson. *Continuous Multivariate Distributions*. John Wiley and Sons, 2000.

[134] O. Abdel-Hamid and H. Jiang. Fast speaker adaptation of hybrid NN/HMM model for speech recognition based on discriminative learning of speaker code. In *Acoust.,*

Speech and Signal Proc., IEEE Int. Conf. on, Vancouver, BC, Canada, pages 7942–7946. IEEE, 2013.

[135] K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, and Y. Oura. Speech synthesis based on hidden Markov models. *Proc. of the IEEE*, 101:1234–1252, 2013.

[136] F. Alvaro, J.-A. Sanchez, and J.-M. Benedi. Recognition of on-line handwritten mathematical expressions using 2D stochastic context-free grammars and hidden Markov models. *Pattern Recognit. Lett.*, 35:58–67, 2014.

[137] J. Baumgartner, A. G. Flesia, J. Gimenez, and J. Pucheta. A new image segmentation framework based on two-dimensional hidden Markov models. *Integr. Comput.-Aided Eng.*, 23:1–13, 2016.

[138] L. Rossi, J. Chakareski, P. Frossard, and S. Colonnese. A Poisson hidden Markov model for multiview video traffic. *IEEE/ACM Trans. on Netw.*, 23:547–558, 2015.

[139] A. Soualhi, H. Razik, G. Clerc, and D. D. Doan. Prognosis of bearing failures using hidden Markov models and the adaptive neuro-fuzzy inference system. *IEEE Trans. on Ind. Electron.*, 61:2864–2874, 2014.

[140] Y. Cao, Y. Li, S. Coleman, A. Belatreche, and T. M. McGinnity. Adaptive hidden Markov model with anomaly states for price manipulation detection. *IEEE Trans. on Neural Netw. and Learn. Syst.*, 26:318–330, 2015.

[141] A. Punzo and A. Maruotti. Clustering multivariate longitudinal observations: The contaminated Gaussian hidden Markov model. *J. of Comput. and Graph. Stat.*, 25:1097–1116, 2016.

[142] F. Cuzzolin and M. Sapienza. Learning pullback HMM distances. *IEEE Trans. on Pattern Anal. and Mach. Intell.*, 36(7):1483–1489, 2014.

[143] M. N. Do. Fast approximation of Kullback–Leibler distance for dependence trees and hidden Markov models. *IEEE Signal Proc. Lett.*, 10(4):115–118, 2003.

[144] L. Chen and H. Man. Fast schemes for computing similarities between Gaussian HMMs and their applications in texture image classification. *EURASIP J. on Appl. Signal Proc.*, 13:1984–1993, 2005.

[145] S. M. E. Sahraeian and B.-J. Yoon. A novel low-complexity HMM similarity measure. *IEEE Signal Proc. Lett.*, 18(2):87–90, 2011.

[146] J. Zeng, J. Duan, and C. Wu. A new distance measure for hidden Markov models. *Expert Syst. with Appl.*, 37:1550–1555, 2010.

[147] B.-H. Juang and L. R. Rabiner. A probabilistic distance measure for hidden Markov models. *AT&T Tech. J.*, 64(2):391–408, 1985.

[148] T. Jebara and R. Kondor. Bhattacharyya and expected likelihood kernels. In B. Schölkopf and M. K. Warmuth, editors, *16th Annu. Conf. on Learn. Theory and 7th Kernel Workshop, Proc.*, volume 2777 of *Lecture Notes in Computer Science*, pages 57–71. Springer Berlin Heidelberg, 2003.

[149] R. Jonker and A. Volgenant. A shortest augmenting path algorithm for dense and spare linear assignment problems. *Comput.*, 38:325–340, 1987.

[150] W. Masoudimansour and N. Bouguila. Generalized Dirichlet mixture matching projection for supervised linear dimensionality reduction of proportional data. In *Multimed. Signal Proc., IEEE 18th Int. Workshop on*, pages 1–6. IEEE, 2016.

Appendix A

Gamma Parameters Estimation

The Gamma distribution, parametrized with a shape α and a rate β parameters, is expressed as $\Gamma(x;\alpha,\beta) = (\beta^{\alpha}x^{\alpha-1}e^{-x\beta})/\Gamma(\alpha)$, with $\alpha, \beta > 0$ and $x \geq 0$. Using the same notations as in Chapter 3, the terms of the data log-likelihood function to be maximized for the estimation of the parameters of the univariate Gamma distributions associated to each state k and mixture component m can be written as

$$L_{Gam}(x;\alpha,\beta) = \sum_{t=0}^{T}\sum_{k=1}^{K}\sum_{m=1}^{M}\gamma_{k,m}^{t}\Big\{\alpha_{k,m}\ln(\beta_{k,m}) - x_t\beta_{k,m} + (\alpha_{k,m}-1)\ln(x_t) - \ln(\Gamma(\alpha_{k,m}))\Big\}\,. \tag{97}$$

x_t is the t-th element of the sequence of data of T elements (pixels in my case) and $\gamma_{k,m}$ is defined in Section 5.2. For each state and mixture component the maximization of this quantity is iteratively performed by the Newton-Raphson method which is expressed as

$$\begin{pmatrix}\alpha^{new}\\ \beta^{new}\end{pmatrix} = \begin{pmatrix}\alpha\\ \beta\end{pmatrix} - H^{-1}(\alpha,\beta)\frac{\partial L_{Gam}(x;\alpha,\beta)}{\partial(\alpha,\beta)}\,. \tag{98}$$

In the case of the univariate Gamma distributions, the involved quantities can be easily computed by hand.

Appendix B

KL divergence between two Dirichlet distributions

Hereafter are shown the steps to derive the Kullback-Leibler divergence between two multi-dimensional Dirichlet distributions. The usual notation $KL(p||q)$ is used for the divergence between a distribution p and another distribution q.

Let $p(x|\alpha)$ and $q(x|a)$ denote two D-dimensional Dirichlet distributions as defined in Equation (85) and derive the following quantity

$$KL_{dir}(p||q) = \int p(x) \ln \frac{p(x)}{q(x)} dx \; . \tag{99}$$

One typically recognizes the expression of an expectation with respect to p and I set the following notation for it

$$\begin{aligned} KL_{dir}(p||q) &= \left\langle \ln \frac{p(x)}{q(x)} \right\rangle_{p(x)} \\ &= \langle \ln(p(x)) - \ln(q(x)) \rangle_{p(x)} \; . \end{aligned} \tag{100}$$

Using Equations (85) and (100), I get

$$
\begin{aligned}
KL_{dir}(p||q) =& \Big\langle \ln\Big(\Gamma\Big(\sum_{d=1}^{D}\alpha_d\Big) - \sum_{d=1}^{D}\ln(\Gamma(\alpha_d))\Big) - \ln\Big(\Gamma\Big(\sum_{d=1}^{D}a_d\Big) + \sum_{d=1}^{D}\ln(\Gamma(a_d))\Big) \\
&+ \sum_{d=1}^{D}(\alpha_d - a_d)\ln(x_d)\Big\rangle_{p(x)} , \qquad (101)
\end{aligned}
$$

which can be simplified as

$$
\begin{aligned}
KL_{dir}(p||q) =& \ln\Big(\Gamma\Big(\sum_{d=1}^{D}\alpha_d\Big) - \sum_{d=1}^{D}\ln(\Gamma(\alpha_d))\Big) - \ln\Big(\Gamma\Big(\sum_{d=1}^{D}a_d\Big) + \sum_{d=1}^{D}\ln(\Gamma(a_d))\Big) \\
&+ \sum_{d=1}^{D}(\alpha_d - a_d)\langle\ln(x_d)\rangle_{p(x)} . \qquad (102)
\end{aligned}
$$

With the Dirichlet distributions parameters known, the only quantity which needs to be evaluated is $\langle\ln(x_d)\rangle_{p(x)}$. Making use of Equation (85),

$$
\begin{aligned}
\langle\ln(x_d)\rangle_{p(x)} &= \int p(x)\ln(x_d)dx \\
&= \frac{\Gamma(\sum_{d=1}^{D}\alpha_d)}{\prod_{d=1}^{D}\Gamma(\alpha_d)}\int \ln(x_d)\prod_{d=1}^{D}x_d^{\alpha_d-1}dx . \qquad (103)
\end{aligned}
$$

Using the property $\ln(x)x^t = \dfrac{d}{dt}(x^t)$ (and the fact the α_i's are independent) along with the Leibniz integral rule,

$$
\begin{aligned}
\langle\ln(x_d)\rangle_{p(x)} &= \frac{\Gamma(\sum_{d=1}^{D}\alpha_d)}{\prod_{d=1}^{D}\Gamma(\alpha_d)}\int \frac{\partial}{\partial\alpha_d}\Big(\prod_{d=1}^{D}x_d^{\alpha_d-1}\Big)dx \\
&= \frac{\Gamma(\sum_{d=1}^{D}\alpha_d)}{\prod_{d=1}^{D}\Gamma(\alpha_d)}\frac{\partial}{\partial\alpha_d}\int \prod_{d=1}^{D}x_d^{\alpha_d-1}dx . \qquad (104)
\end{aligned}
$$

Using the fact that by definition the integral of the Dirichlet distribution is equal to 1, I obtain

$$
\langle\ln(x_d)\rangle_{p(x)} = \frac{\Gamma(\sum_{d=1}^{D}\alpha_d)}{\prod_{d=1}^{D}\Gamma(\alpha_d)}\frac{\partial}{\partial\alpha_d}\Big(\frac{\prod_{d=1}^{D}\Gamma(\alpha_d)}{\Gamma(\sum_{d=1}^{D}\alpha_d)}\Big) . \qquad (105)
$$

By recognizing the typical form of the logarithm function derivative and the digamma function expression, I find

$$
\begin{aligned}
\langle \ln(x_d) \rangle_{p(x)} &= \frac{\partial}{\partial \alpha_d} \Big[\ln \Big(\frac{\prod_{d=1}^{D} \Gamma(\alpha_d)}{\Gamma(\sum_{d=1}^{D} \alpha_d)} \Big) \Big] \\
&= \frac{\partial}{\partial \alpha_d} \Big(\ln \Big(\prod_{d=1}^{D} \Gamma(\alpha_d) \Big) \Big) - \frac{\partial}{\partial \alpha_d} \Big(\ln \Big(\Gamma(\sum_{d=1}^{D} \alpha_d) \Big) \Big) \\
&= \frac{\partial}{\partial \alpha_d} (\ln(\Gamma(\alpha_d))) - \frac{\partial}{\partial \alpha_d} \Big(\ln \Big(\Gamma(\sum_{d=1}^{D} \alpha_d) \Big) \Big) \\
&= \Psi(\alpha_d) - \Psi \Big(\sum_{d=1}^{D} \alpha_d \Big) \, . \qquad (106)
\end{aligned}
$$

in which I made use of the fact that the α_i's are independent variables.

This last equation used in Equation (102) leads to the expression of Equation (93).

Appendix C

KL divergence between two generalized Dirichlet distributions

Hereafter are shown the steps to derive the Kullback-Leibler divergence between two multidimensional generalized Dirichlet distributions. The notations hereafter are the same as in Appendix B.

Let $p(x|\alpha, \beta)$ and $q(x|a, b)$ denote two D-dimensional generalized Dirichlet distributions as defined in Equation (86) and derive the following quantity

$$KL_{GD}(p||q) = \int p(x) \ln\left(\frac{p(x)}{q(x)}\right) dx \ . \tag{107}$$

I recall the equation of the GD distribution p:

$$p(x|\alpha, \beta) = \prod_{d=1}^{D} \frac{\Gamma(\alpha_d + \beta_d)}{\Gamma(\alpha_d)\Gamma(\beta_d)} x_d^{\alpha_d - 1} \left(1 - \sum_{s=1}^{d} x_s\right)^{\nu_d} , \tag{108}$$

with ν_d defined as in Equation (78) and denoting its equivalent in q as c_d.

Using Equation (108) in Equation (107), I get

$$
\begin{aligned}
KL_{GD}(p||q) &= \left\langle \ln\left(\frac{p(x)}{q(x)}\right)\right\rangle_{p(x)} \\
&= \sum_{d=1}^{D} \ln\left(\frac{\Gamma(\alpha_d+\beta_d)\Gamma(a_d)\Gamma(b_d)}{\Gamma(\alpha)\Gamma(\beta)\Gamma(a+b)}\right) + \sum_{d=1}^{D}(\alpha_d - a_d)\langle \ln(x_d)\rangle_{p(x)} \\
&+ \sum_{d=1}^{D}(\nu_d - c_d)\left\langle \ln\left(1-\sum_{s=1}^{d} x_s\right)\right\rangle_{p(x)} \ .
\end{aligned}
\tag{109}
$$

It would be possible to derive the full expression of this KL divergence by using steps similar to the ones presented in the case of the Dirichlet. However, the presence in this case of a second expectation makes this method being heavy in computation and I prefer using the following routine that is less straightforward, but less heavy to write to find the expressions of the two expectations left in Equation (109).

I start by computing the derivative of a GD distribution with respect to all its parameters.

$$
\frac{\partial p(x)}{\partial \alpha_d} = p(x)\left[\Psi(\alpha_d+\beta_d) - \Psi(\alpha_d) + \ln(x_d) - \ln\left(1-\sum_{s=1}^{d-1} x_s\right)\right] , \tag{110}
$$

is valid for all $d \in [1, D]$ if the last term is defined as equal to 0 in the case $d = 1$.

Similarly,

$$
\frac{\partial p(x)}{\partial \beta_d} = p(x)\left[\Psi(\alpha_d+\beta_d) - \Psi(\beta_d) + \ln\left(1-\sum_{s=1}^{d} x_s\right) - \ln\left(1-\sum_{s=1}^{d-1} x_s\right)\right] , \tag{111}
$$

is valid for all $d \in [1, D]$ if the last term is defined as equal to 0 in the case $d = 1$.

Integrating Equations (110) and (111) using the Leibniz rule and identifying the expectation expressions, I get the following system of equations:

$$
\begin{cases}
\Psi(\alpha_d+\beta_d) - \Psi(\alpha_d) + \langle \ln(x_d)\rangle_{p(x)} - \left\langle \ln\left(1-\sum_{s=1}^{d-1} x_s\right)\right\rangle_{p(x)} = 0 \, , & (112) \\
\Psi(\alpha_d+\beta_d) - \Psi(\beta_d) + \left\langle \ln\left(1-\sum_{s=1}^{d} x_s\right)\right\rangle_{p(x)} - \left\langle \ln\left(1-\sum_{s=1}^{d-1} x_s\right)\right\rangle_{p(x)} = 0 \, , & (113)
\end{cases}
$$

which is valid for all $d \in [1, D]$, with the last term of the left hand side being equal to 0 for $d = 1$.

This system of equations can recursively be solved and lead to the solution:

$$\begin{cases} \left\langle \ln\left(1 - \sum_{s=1}^{d-1} x_s\right)\right\rangle_{p(x)} = -\sum_{s=1}^{d} (\Psi(\alpha_s + \beta_s) - \Psi(\beta_s)) \,, & (114) \\ \langle \ln(x_d) \rangle_{p(x)} = \Psi(\alpha_d) - \Psi(\beta_d) - \sum_{s=1}^{d} (\Psi(\alpha_s + \beta_s) - \Psi(\beta_s)) \,. & (115) \end{cases}$$

Using (115) in (109), I obtain the final expression:

$$\begin{aligned} KL_{GD}(p||q) = & \sum_{d=1}^{D} \ln\left(\frac{\Gamma(\alpha_d + \beta_d)\Gamma(a_d)\Gamma(b_d)}{\Gamma(\alpha_d)\Gamma(\beta_d)\Gamma(a_d + b_d)}\right) \\ & - \sum_{d=1}^{D} (\alpha_d - a_d)\left(\Psi(\alpha_d) - \Psi(\beta_d) - \sum_{s=1}^{d} (\Psi(\alpha_s + \beta_s) - \Psi(\beta_s))\right) \\ & + \sum_{d=1}^{D} (\nu_d - c_d) \sum_{s=1}^{d} (\Psi(\alpha_s + \beta_s) - \Psi(\beta_s)) \,. \end{aligned} \tag{116}$$