Online learning for audio clustering and segmentation

Alberto Bietti

École Normale Supérieure de Cachan
Master Mathématiques, Vision et Apprentissage

Mines ParisTech
Option MAREVA

INRIA, IRCAM

Online learning for audio clustering and segmentation

Author:
Alberto Bietti

Supervisors:
Arshia Cont
Francis Bach

September 2014

Abstract

Audio segmentation is an essential problem in many audio signal processing tasks which tries to segment an audio signal into homogeneous chunks, or segments. Most current approaches rely on a change-point detection phase for finding segment boundaries, followed by a similarity matching phase which identifies similar segments. In this thesis, we focus instead on joint segmentation and clustering algorithms which solve both tasks simultaneously, through the use of unsupervised learning techniques in sequential models. Hidden Markov and semi-Markov models are a natural choice for this modeling task, and we present their use in the context of audio segmentation. We then explore the use of online learning techniques in sequential models and their application to real-time audio segmentation tasks. We present an existing online EM algorithm for hidden Markov models and extend it to hidden semi-Markov models by introducing a different parameterization of semi-Markov chains. Finally, we develop new online learning algorithms for sequential models based on incremental optimization of surrogate functions.

Résumé

Le problème de la segmentation audio, essentiel dans de nombreuses tâches de traitement du signal audio, cherche à décomposer un signal audio en courts segments de contenu homogène. La plupart des approches courantes en segmentation sont basées sur une phase de détection de rupture qui trouve les limites entre segments, suivie d'une phase de calcul de similarité qui identifie les segments similaires. Dans ce rapport, nous nous intéressons à une approche différente, qui cherche à effectuer les deux tâches – segmentation et clustering – simultanément, avec des méthodes d'apprentissage non supervisé dans des modèles séquentiels. Les modèles de Markov et de semi-Markov cachés sont des choix naturels dans ce contexte de modélisation, et nous présentons leur utilisation en segmentation audio. Nous nous intéressons ensuite à l'utilisation de méthodes d'apprentissage en ligne dans des modèles séquentiels, et leur application à la segmentation audio en temps réel. Nous présentons un modèle existant de online EM pour les modèles de Markov cachés, et l'étendons aux modèles de semi-Markov cachés grâce à une nouvelle paramétrisation des chaines de semi-Markov. Enfin, nous introduisons de nouveaux algorithmes en ligne pour les modèles séquentiels qui s'appuient sur une optimisation incrémentale de fonctions surrogées.

Contents

Acknowledgements

First, I would like to thank my advisors, Arshia Cont and Francis Bach, whose support and expertise have been of invaluable help throughout my internship. I also would like to thank them for giving me the opportunity to conciliate my two main passions – machine learning and music – with this project. Being a musician myself, working at IRCAM has been a wonderful experience. I enjoyed being exposed to the world of contemporary classical music as well as learning about state of the art scientific research on music and audio. I also learned a lot from the various seminars and group meetings I attended in the SIERRA group at INRIA, which has shown to be an incredible environment for machine learning research.

I would like to thank Philippe Cuvillier, with whom I shared many fruitful and insightful discussions. I also want to thank Olivier Cappé and Mathieu Lagrange for helpful discussions. Finally, I am very grateful of the excellent academic education I received, both in the MVA master's at ENS Cachan, and at Mines ParisTech.

Merci aux stagiaires de l'IRCAM pour les bons moments qu'on a passés ensemble. Merci Charles et Lénaïc pour les bons souvenirs que je garderai de cette année. Grazie mamma, papà, Elettra, per avermi fatto crescere nel modo in cui sono cresciuto.

Chapter 1

Introduction

The task of audio segmentation has been an active area of research in the audio signal processing community since the 1980s. It aims to decompose an audio signal into homogeneous chunks, or *segments*, such that neighboring segments contain what a listener would consider different audio content. Perhaps the most common use case of audio segmentation is musical note segmentation, where an algorithm attempts to detect the start and end of each note in a musical piece. This can be extended to discovering musical structures, where segments can be more than just single notes, but rather short structures, such as chords or vibratos, which are homogeneous or close to being so. The problem of audio segmentation is however more general than musical segmentation, and could also be applied to segmenting auditory scenes in an audio file. Example applications of audio segmentation include music indexing for information retrieval, audio summarization (where the summary can then be synthesized), fingerprinting, or analyzing higher-level musical structures from the symbolic representation given by the segmentation.

Many approaches to audio segmentation rely on detecting abrupt changes in the audio content (such as musical onsets) which determine the boundaries of each segment, a task known as *change detection*. This is often accomplished by looking at specific audio features such as spectral centroid, spectral flux or zero crossings (Tzanetakis and Cook, 1999). Another important technique for change-point detection is that of sequential hypothesis testing, using e.g. the CUSUM algorithm (Tartakovsky et al., 2014; Basseville et al., 1993). Recently, Dessein and Cont (2013); Dessein (2012) applied these techniques to real-time audio segmentation using the information geometry of exponential families. These approaches don't generally assign a label to each segment directly, and one usually needs to compute similarity matrices between segments in order to find similar segments and discover the underlying structure, which is not adapted to real-time settings. Lostanlen (2013); Cont et al. (2011) show how segmentations and similarities can be computed in an information-geometric context by finding the right centroids for each segment.

In many cases, the audio segmentation task is cast to a supervised learning task, in which detecting a change point or a specific segment is done using a classifier which needs to be trained from labeled data. This makes the algorithm depend heavily on the training examples used, and might not easily adapt to new auditory environments.

We will focus on unsupervised learning approaches to audio segmentation, where no training data is required and we rely on the expressiveness of the model to discover the correct segmentation. Rather than separately detecting changes and finding similarities, we wish to do both at the same time, by having different models for each class of segments and inferring the segment transitions from observations. This naturally leads to the framework of hidden Markov models (HMMs) (Cappé et al., 2005; Rabiner, 1989) and hidden semi-Markov models (HSMMs) (Yu, 2010), which are powerful modeling tools and have had great success in audio signal processing, particularly speech recognition. In Chapter 2, we introduce the representation we use for audio signals, the use of Bregman divergences for computing similarities, and present the main

algorithms for (offline) learning in various models of interest, including HMMs and HSMMs.

Online learning has proven to be an effective way to improve learning, especially in large-scale settings (Bottou and Bousquet, 2008; Bottou, 1998). Most of its successes have been in the context of independent observations, e.g., (Mairal et al., 2010; Cappé and Moulines, 2009), and little work has been done to apply online learning to sequential models, such as hidden Markov or semi-Markov models. O. Cappé proposed an online EM algorithm for HMMs based on a forward smoothing recursion (Cappé, 2011). Yildirim et al. (2012) proposed an online EM algorithm for changepoint models (including HSMMs), but they rely on sequential Monte Carlo sampling techniques as in (Cappé, 2009), which aren't always desirable.

In Chapter 3, we present various online learning algorithms for HMMs and HSMMs. We propose an extension of the online EM algorithm of Cappé (2011) to HSMMs in Section 3.1.2, and various other online algorithms for HMMs (which can also be extended to HSMMs) based on incremental optimization of surrogate functions. Chapter 4 presents some experiments for the audio segmentation task on musical examples.

The online learning algorithms present in Chapter 3 could also be beneficial to the *Antescofo* score-following system built at IRCAM (Cont, 2010), which is a real-time audio-to-score alignment system based on inference in a hidden semi-Markov model. At present time, *Antescofo* involves no learning and its observation model relies on fixed templates artificially created for each note or chord that is observed. Because this observation model is quite weak, the model relies heavily on the duration criteria of the notes on the score, and isn't very robust to changes in sound, due to different instruments or different environments. Our online learning algorithms could be used to learn these observations models adaptively over time, starting from the hand-crafted templates as a prior model.

Our main contributions are the following:

- an extension of the online EM algorithm of (Cappé, 2011) to HSMMs, thanks to a two-variable parameterization of the HSMM
- various algorithms for online learning in HMMs and HSMMs based on incremental optimization schemes, and a comparison of these models
- applications of these algorithms to audio segmentation, and a potential application to the *Antescofo* score-following system for adaptively learning the observation templates.

Chapter 2

Representations, models and offline algorithms

In this chapter, we will provide some background on the building blocks we use for audio segmentation. Our focus will be on unsupervised learning techniques, where no labeled data is provided and we rely on the richness of the model to correctly identity the segments and musical structures. We will start by discussing the choice of representation for the audio signals. We then present the modeling tools based on Bregman divergences that we use on our representations, and finally we describe Hidden Markov and semi-Markov models, which are well-suited for the sequential nature of our task, and derive appropriate learning algorithms.

2.1 Audio signal representation

An audio signal $x(t)$ is commonly described by frequency representations, given by the *Fourier transform*.

Definition 2.1 (Fourier transform)**.**

$$\hat{x}(\omega) = \int_{-\infty}^{+\infty} x(t)e^{-i\omega t}dt \tag{2.1}$$

The Fourier transform captures the frequency information of the entire signal, and therefore fails to capture local changes in frequency components. This problem is overcome by the *windowed Fourier transform*, or *short-time Fourier transform* (STFT), which uses a real and symmetric window $g(t)$ (typically bell-shaped) to localize the Fourier transform around each time t.

Definition 2.2 (Continuous-time STFT)**.**

$$\hat{x}(t,\omega) = \int_{-\infty}^{+\infty} x(u)g(u-t)e^{-i\omega u}du \tag{2.2}$$

Common choices for the window function g are the Gaussian window, the Hamming window or the Hanning window. If we now consider a discrete signal $x[t]$, where $t \in \mathbb{Z}$, we get the following discrete-time transform:

Definition 2.3 (Discrete-time STFT)**.**

$$\hat{x}(t,e^{i\omega}) = \sum_{u=-\infty}^{+\infty} x[u]g[u-t]e^{-i\omega u} \tag{2.3}$$

When g has compact support, each $\hat{x}(t, e^{iw})$ for fixed t can be obtained by taking the discrete Fourier transform of the signal $u \mapsto x[u]g[u-t]$ on a time window centered around t (typically using an FFT algorithm), and is thus given by a fixed number p of coefficients $\hat{x}_{t,1}, \dots, \hat{x}_{t,p} \in \mathbb{C}$.

In the following, we consider the audio signal to be represented by the modulus of its STFT coefficients: $x_t \in \mathbb{R}^p = (|\hat{x}_{t,1}|, \dots, |\hat{x}_{t,p}|)^\top$, where the subscript $t = 1, \dots, T$ denotes the index of the time windows – which have a constant offset and can have some overlap – and can potentially be unbounded in the online (streaming audio) case. In order to provide some invariance to the volume of the sound, we will consider the vectors x_t to be normalized, i.e. $\sum_i x_{t,i} = c > 0$ for all t.

Note that this representation is quite primitive and more informative representations could be used instead (e.g. MFCCs or scattering coefficients (Andén and Mallat, 2011)) to improve performance, however we will restrict ourselves to this simple representation, and will rely on the richness of our models as a means for improving results.

2.2 Bregman divergences

In order to cluster data points together, we need to be able to compare them using a similarity measure $D(x, y)$. In many cases the chosen measure is the Euclidian distance, or its square: $D(x, y) = \|x - y\|^2$. However, the choice of similarity measure makes some implicit assumptions about the geometry of the data, and in particular using the Euclidian distance assumes Euclidian geometry. Empirical evidence shows that alternative measures, such as the Kullback-Leibler (KL) or Itakura-Saito (IS) divergences perform better in practice for audio signals (Cont et al., 2011; Stylianou and Syrdal, 2001). In particular, when dealing with non-negative normalized vectors, the KL divergence is a natural choice since it corresponds to a multinomial distribution, as we will see in Section 2.3.3 (see, e.g., Banerjee et al. (2005b)).

2.2.1 Definition and examples

Definition 2.4 (Bregman divergence)**.** *Let $\psi : \Omega \to \mathbb{R}$ be a strictly convex function defined on a convex set $\Omega \subset \mathbb{R}^p$. The* Bregman divergence *associated with ψ is defined by*

$$D_\psi(x, y) = \psi(x) - \psi(y) - \langle x - y, \nabla\psi(y)\rangle. \tag{2.4}$$

It follows from the strict convexity of ψ that $D_\psi(x, y) \geq 0$, with equality if and only if $x = y$. Note that in general Bregman divergences are not symmetric and do not satisfy the triangle inequality, and hence are not distance metrics. We now give some classical examples of Bregman divergences relevant for audio.

Example 2.5. The squared Euclidian distance $\|x - y\|^2$ is a Bregman divergence, with $\psi(x) = \|x\|^2$. Indeed,

$$\begin{aligned} D_\psi(x, y) &= \|x\|^2 - \|y\|^2 - \langle x - y, 2y\rangle \\ &= \|x\|^2 + \|y\|^2 - 2\langle x, y\rangle \\ &= \|x - y\|^2. \end{aligned}$$

Example 2.6. The KL divergence $D_{KL}(x\|y) = \sum_i x_i \log \frac{x_i}{y_i}$, with $x_i, y_i \in \mathbb{R}_+$ and $\sum_i x_i = \sum_i y_i = c$ (where $c = 1$ for probability distributions) is a Bregman divergence, with $\psi(x) =$

$\sum_i x_i \log x_i$. Indeed, we have

$$\begin{aligned} D_\psi(x,y) &= \sum_i x_i \log x_i - \sum_i y_i \log y_i - \sum_i (x_i - y_i)(1 + \log y_i) \\ &= \sum_i x_i \log \frac{x_i}{y_i} - \sum_i x_i + \sum_i y_i \\ &= \sum_i x_i \log \frac{x_i}{y_i} = D_{KL}(x\|y). \end{aligned}$$

Note that if we define ψ as a function of the $p-1$-dimensional vector $x = [x_i]_{i\in\{1,\dots,p-1\}}$ by $\psi(x) = \sum_{i=1}^{p-1} x_i \log x_i + (c - \sum_{i=1}^{p-1} x_i) \log(c - \sum_{i=1}^{p-1} x_i)$, it can be easily shown that the associated Bregman divergence still takes the same form. This formulation has the benefit of having ψ defined on a set with non-empty interior, which will be useful for Legendre duality properties as we will see in §2.2.2 and §2.3.3.

Example 2.7. The Itakura-Saito divergence[1] $D_{IS}(x,y) = \sum_i (\frac{x_i}{y_i} - \log \frac{x_i}{y_i} - 1)$ with $x_i, y_i > 0$ is a Bregman divergence with $\psi(x) = -\sum_i \log x_i$. We have indeed

$$\begin{aligned} D_\psi(x,y) &= -\sum_i \log x_i + \sum_i \log y_i - \sum_i (x_i - y_i) \\ &= \sum_i \left(\frac{x_i}{y_i} - \log \frac{x_i}{y_i} - 1 \right) = D_{IS}(x,y). \end{aligned}$$

2.2.2 Duality

When ψ is differentiable in the interior of Ω and diverges near its boundary, the function ψ and its Fenchel conjugate[2] ψ^* are so-called Legendre duals of each other (Rockafellar, 1997; Banerjee et al., 2005b; Amari and Nagaoka, 2007) and verify $\nabla\psi^* = (\nabla\psi)^{-1}$. If $\mu = \nabla\psi(x)$, or equivalently $x = \nabla\psi^*(\mu)$, we have

$$\psi(x) + \psi^*(\mu) = \langle x, \mu \rangle. \tag{2.5}$$

Then, if we let $\mu = \nabla\psi(x)$ and $\mu = \nabla\psi(y)$, we have the following duality relation between D_ψ and D_{ψ^*}:

$$D_\psi(x,y) = D_{\psi^*}(\nu, \mu). \tag{2.6}$$

2.3 Clustering with Bregman divergences

Now that we have defined a similarity measure between data points, our goal is to use it to cluster data points together. We will focus on centroid-based clustering methods, where each cluster of points is described by its centroid, the K-means algorithm being the simplest example. We will start by defining and characterizing centroids for Bregman divergences (see, e.g. Nielsen and Nock (2009)), then, based on Banerjee et al. (2005b) we will describe a generalization of K-means to Bregman divergences, as well as extensions to probabilistic models thanks to a bijection between Bregman divergences and exponential families. In this section, we will ignore the temporal dependency of data points, and will consider any set of points $\{x_i\}_{i=1..n}$.

[1] In signal processing, the IS divergence is often used with the power spectrum $F(w) = |\hat{x}(w)|^2$ of a signal $x(t)$, which is a function rather than a vector, but we limit ourselves to vectors in the example for simplicity.

[2] The Fenchel conjugate of ψ is defined by $\psi^*(\mu) = \max_{x\in\Omega} \langle \mu, x \rangle - \psi(x)$.

2.3.1 Bregman centroids

A centroid is defined as the point μ minimizing the average distance to a set of other points $\{x_i\}_{i=1..n}$. If the distance we consider is the squared Euclidian distance, it is well known that the centroid is given by the average $\bar{x} = \frac{1}{n}\sum_i x_i$. Since Bregman divergences are not symmetric, one can define three different types of centroids as follows.

Definition 2.8. *The* right-type, left-type *and* symmetrized Bregman centroids *for the divergence D_ψ are defined respectively by*

$$\mu_\psi^R = \arg\min_c \frac{1}{n}\sum_{i=1}^n D_\psi(x_i, c)$$

$$\mu_\psi^L = \arg\min_c \frac{1}{n}\sum_{i=1}^n D_\psi(c, x_i)$$

$$\mu_\psi^S = \arg\min_c \frac{1}{n}\sum_{i=1}^n \frac{D_\psi(x_i, c) + D_\psi(c, x_i)}{2}.$$

The following proposition characterizes right-type Bregman centroids to be point averages and is a key result for extending K-means to Bregman divergences.

Proposition 2.1. *Given a set of points $\{x_i\}_{i=1..n}$ and a Bregman divergence D_ψ, the right-type Bregman centroid is given by*

$$\mu_\psi^R = \bar{x} := \frac{1}{n}\sum_{i=1}^n x_i. \tag{2.7}$$

Proof. We have

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^n D_\psi(x_i, c) &= \frac{1}{n}\sum_{i=1}^n (\psi(x_i) - \psi(c) - \langle x_i - c, \nabla\psi(c)\rangle) \\
&= \left(\frac{1}{n}\sum_{i=1}^n \psi(x_i)\right) - \psi(c) - \langle \bar{x} - c, \nabla\psi(c)\rangle \\
&= \left(\frac{1}{n}\sum_{i=1}^n \psi(x_i)\right) - \psi(\bar{x}) + D_\psi(\bar{x}, c),
\end{aligned}$$

which is minimized for $D_\psi(\bar{x}, c) = 0$, i.e. for $c = \bar{x}$. □

The proof can easily be extended to the weighted case, and we obtain the following result.

Proposition 2.2. *The unique minimizer of the weighted sum $\sum_i w_i D_\psi(x_i, \mu)$, with $w_i \geq 0$ and $\sum_i w_i = 1$, is given by $\mu = \sum_i w_i x_i$.*

Interestingly, Banerjee et al. (2005a) give a reverse characterization of Bregman divergences as the only "loss" functions, under some mild assumptions, for which this proposition is true.

We will limit ourselves to right-type centroids in the following because of their simplicity. That said, it is possible to easily compute left-type centroids thanks to (2.6) by noting that the left-type centroid under D_ψ corresponds to a right-type centroid under D_{ψ^*} in the dual space given by the one-to-one mapping $x \mapsto \nabla\psi(x)$:

$$\mu_\psi^L = (\nabla\psi)^{-1}\left(\frac{1}{n}\sum_{i=1}^n \nabla\psi(x_i)\right). \tag{2.8}$$

By switching to the dual representation and changing the divergence to D_{ψ^*}, one can thus reformulate a problem with left-type centroids as one with right-type centroids. As for symmetrized Bregman centroids, Nielsen and Nock (2009) proposed a dichotomic geodesic-walk algorithm to approximate them, using the differential geometric properties of Bregman divergences.

2.3.2 K-means

The K-means algorithm clusters data points around K centroids by alternating between two steps: (i) assign each point to its closest centroid, (ii) update the centroids from the newly assigned points. Although the classical K-means algorithm uses the squared Euclidian distance $\|x - \mu\|^2$ as a way to measure distance between a point x and its centroid μ, Banerjee et al. (2005b) show that the algorithm can be extended to use any Bregman divergence, with the centroid on the right, $D_\psi(x, \mu)$, to measure the closeness of x to μ. The algorithm is given in Algorithm 2.1, where the names E-step and M-step come from the close resemblance to the EM algorithm, which we will discuss below.

Algorithm 2.1: K-means algorithm.

Data: $x_1, \ldots, x_n$, initial centroids $\mu_1, \ldots, \mu_K$, Bregman divergence D_ψ.
Result: final centroids μ_k, assignments z_i
repeat
  $z_i \leftarrow \arg\min_k D_\psi(x_i, \mu_k) \quad i = 1, \ldots, n$ `/* E-step */`
  $\mu_k \leftarrow \frac{1}{|\{i : z_i = k\}|} \sum_{i:z_i=k} x_i \quad k = 1, \ldots, K$ `/* M-step */`
until *convergence*;

The following proposition justifies the algorithm and explains its convergence.

Proposition 2.3. *The K-means algorithm decreases the following objective (also known as* distortion*) at each step by performing coordinate descent:*

$$\ell(\boldsymbol{\mu}, \boldsymbol{z}) = \sum_{i=1}^{n} D_\psi(x_i, \mu_{z_i}). \tag{2.9}$$

Furthermore, convergence is reached in a finite number of iterations.

Proof. The E-step minimizes the objective w.r.t. $\mathbf{z}$ by minimizing each individual term $D_\psi(x_i, \mu_{z_i})$. That the M-step minimizes the objective w.r.t. $\boldsymbol{\mu}$ follows from Proposition 2.1. Let $\ell_1(\mathbf{z}) = \min_{\boldsymbol{\mu}} \ell(\boldsymbol{\mu}, \mathbf{z})$, that is the value of the objective after the M-step. ℓ_1 is non-negative and decreases with each iteration, and therefore converges. Since there is only a finite number of possible assignments, convergence is reached in a finite number of iterations. □

Initialization One issue with the K-means objective – and with most of the problems we will deal with – is that it is non-convex, and thus relies heavily on a good initialization. A simple approach is to initialize the centroids to random points in the dataset, and keep the result with lowest distortion after multiple such initializations. K-means++ (Arthur and Vassilvitskii, 2007) is an example of another initialization scheme.

Choice of K The number of clusters K needs to be chosen in order to run K-means, and the question comes up of how to pick a good value for it. Of course, if one knows in advance how many clusters are present in the data, then the choice is immediate, however such information isn't always available and it is common to run the algorithm for different values of K and evaluate which is best using some measure. In practice, this measure is often taken to be a penalized distortion measure, where a penalty term λK is added to the objective in (2.9). Kulis and Jordan

(2012) derive an algorithm, called DP-means, which incrementally adds new clusters when all existing clusters are further away than λ from the current point. Their algorithm is justified as an asymptotic Gibbs sampling algorithm in a Dirichlet process mixture of Gaussians (with fixed isotropic covariances), where the variance goes to zero, and is shown to decrease the penalized objective until convergence. Although the link with Bayesian nonparametrics made in (Kulis and Jordan, 2012) relies on the Gaussian distribution and thus the Euclidian distance, the same algorithm with a Bregman divergence still satisfies the property of decreasing the penalized objective.

2.3.3 Bregman divergences and exponential families

Although K-means provides a simple and effective clustering algorithm, it lacks the flexibility of a probabilistic model and the ability to have soft-assignments of a data point to multiple clusters. Mixture models such as Gaussian mixture models are widely used to address these issues, and Banerjee et al. (2005b) show how one can use such models with Bregman divergences by establishing a bijection between Bregman divergences and exponential families (with some restrictions) and using the corresponding exponential family model as the emission distribution.

An *exponential family* distribution with (natural) parameter $\theta \in \Theta \subset \mathbb{R}^p$ and sufficient statistic $\phi(x) \in \mathbb{R}^p$ is given by the density

$$p_\theta(x) = \exp(\langle \phi(x), \theta \rangle - a(\theta)), \tag{2.10}$$

with respect to some measure ν on the input space $\mathcal{X}$ (see, e.g., Wainwright and Jordan (2008); Barndorff-Nielsen (1978)). The log-partition (or cumulant) function a ensures normalization and is given by $a(\theta) = \log \int_{\mathcal{X}} \exp(\langle \phi(x), \theta \rangle) \nu(dx)$. The base measure ν is usually of the form $\nu(dx) = h(x)\lambda(dx)$, where λ is either the Lebesgue measure (for continuous distributions) or the counting measure (for discrete distributions) and $h(x)$ is sometimes called ancillary statistic. Most commonly used distributions, including normal, exponential, gamma, multinomial (with fixed number of trials), and many others, belong to this family. The sufficient statistic $\phi(x)$ is said to be *minimal* if there exists no constant c such that $\langle \phi(x), \theta \rangle = c$ ν-almost everywhere. An important property that we recall is that the derivatives of the log-partition function and the moments of the distribution are linked, and we have in particular

$$\nabla a(\theta) = \mathbb{E}_{X \sim p_\theta}[\phi(X)]. \tag{2.11}$$

In order to establish a bijection with Bregman divergence, we follow Banerjee et al. (2005b) and consider a restricted set of such distributions.

Definition 2.9 (regular exponential family)**.** *A* regular *exponential family distribution is an exponential family distribution for which the parameter space Θ is open and the sufficient statistic $x \in \mathcal{X} = \mathbb{R}^p$ gives a minimal representation. Its density w.r.t. the Lebesgue or counting measure takes the form*

$$p_{\psi,\theta}(x) = h(x) \exp(\langle x, \theta \rangle - \psi(\theta)), \tag{2.12}$$

for $x \in \mathbb{R}^p$, where we denote by ψ the log-partition function.

The log-partition function is always convex, and with the additional properties of this definition, one can show that ψ and its conjugate ψ^* satisfy Legendre-duality properties as in §2.2.2 (see Banerjee et al. (2005b); Barndorff-Nielsen (1978) for details). If we write $\mu = \nabla\psi(\theta)$ – which corresponds to the mean of the distribution from (2.11) –, then $\theta = \nabla\psi^*(\mu)$, and using (2.5), we have

$$\begin{aligned} p_{\psi,\theta}(x) &= h(x) \exp(\langle x, \theta \rangle - \psi(\theta)) \\ &= h(x) \exp(\langle x, \theta \rangle - \langle \mu, \theta \rangle + \psi^*(\mu)) \\ &= h(x) \exp(\psi(x) - \psi(x) + \psi^*(\mu) + \langle x - \mu, \nabla\psi^*(\mu) \rangle) \\ &= h'(x) \exp(-D_{\psi^*}(x, \mu)), \end{aligned}$$

with $h'(x) = h(x)\exp(\psi(x))$. Therefore, a regular exponential familiy can be expressed in terms of the Bregman divergence associated to the conjugate of the log-partition function. Banerjee et al. (2005b) show that the divergences obtained this way belong to a special family which they call *regular Bregman divergences*, and establish a bijection between regular exponential families and regular Bregman divergences.

In practice, this means that when D_ψ is a regular Bregman divergence and we are considering distortions $D_\psi(\cdot, \mu)$ to a centroid μ, we can rely on the corresponding regular exponential family distribution $p_{\psi^*,\theta}$ with $\theta = \nabla\psi(\mu)$ and with log-partition function ψ^*, given by the density

$$p_{\psi^*,\theta}(x) = h(x)\exp(-D_\psi(x, \mu)) \tag{2.13}$$

in a probabilistic context. Note that the mean of this distribution is μ, and the maximum likelihood estimate of μ given observations $x_1, \ldots, x_n$ is given by $\hat{\mu} = n^{-1}\sum_i x_i$, which is the same as computing the right-type Bregman centroid. It is hence a natural choice of distribution to be used as an emission/observation distribution in a mixture model.

We now give two examples of one-to-one correspondences given by this bijection, following Banerjee et al. (2005b).

Example 2.10. If we consider a Gaussian distribution with mean μ and fixed covariance $\sigma^2 I$, taking the conjugate of the log-partition function gives $\psi(x) = \|x\|^2/2\sigma^2$ and thus the corresponding Bregman divergence is $D_\psi(x, \mu) = \|x - \mu\|^2/2\sigma^2$, as we might have expected.

Example 2.11. In the case of a multinomial distribution with fixed number of trials N, we need to make sure to use a minimal representation in order to establish the correspondence. We can use $x = [x_j]_{j\in\{1,\ldots,p-1\}}$ as the minimal sufficient statistic, where $\sum_{j=1}^p x_j = N$, and $\theta = [\log q_j/q_p]_{j\in\{1,\ldots,p-1\}}$ as the natural parameter (which lies in the open set $\mathbb{R}^{p-1}$), where $q_j \geq 0$ are the event probabilities and are such that $\sum_{j=1}^p q_j = 1$. The log-partition function is then $\psi^*(\theta) = N\log(1 + \sum_{j=1}^{p-1} e^{\theta_j})$, and its conjugate is $\psi(\mu) = N\sum_{j=1}^p \frac{x_j}{N}\log\frac{x_j}{N}$, leading to a form of KL-divergence

$$D_\psi(x, \mu) = N\sum_{j=1}^p \frac{x_j}{N}\log\left(\frac{x_j/N}{\mu_j/N}\right).$$

Remark 2.12. Note that in a multinomial distribution, x is almost surely discrete since the base measure is discrete. The bijection implies that if D_ψ is the KL divergence, the corresponding regular exponential family is a multinomial, hence also discrete, even though D_ψ can potentially take any normalized vector $x \in \mathbb{R}_+^p$. In practice, and in our experiments with the audio representation described in § 2.1, one can fix a "number of trials" integer N, normalize $x \in \mathbb{R}_+^p$ such that $\sum_{j=1}^p x_j = N$ and discretize each coordinate x_j to a close integer while keeping the sum equal to N. Larger values of N can give a better, more granular approximation, but it is best to choose N empirically according to experimental results on data. Since our algorithms don't require computing the ancillary statistic $h(x)$ thanks to normalization, the non-integer representation can also be used directly without discretization as an approximation.

2.3.4 Mixture models and the EM algorithm

A mixture model with K mixture components is a probabilistic latent variable model which can be described by the following generative process

$$\begin{aligned} z_i &\sim \pi, \quad i = 1, \ldots, n \\ x_i|z_i &\sim p_{\mu_{z_i}}, \quad i = 1, \ldots, n, \end{aligned}$$

where $z_i \in \{1, \ldots, K\}$ is a latent random variable indicating which cluster observation x_i belongs to. $z \sim \pi$ indicates that z follows a categorical distribution (or multinomial with 1 trial) with parameter $\pi \in \mathbb{R}_+^K$, $\sum_k \pi_k = 1$. p_{μ_k} denotes the observation/emission distribution of mixture

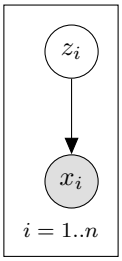


Figure 2.1: Graphical model representation of a mixture model.

component k, with parameter μ_k. In the case of a Gaussian mixture model, the emission parameters are the means and covariance matrices, $\mu_k = (m_k, \Sigma_k)$. If we take our emission distributions to be in the regular exponential family corresponding to a Bregman divergence D_ψ given by the density $p_{(\psi^*,\theta)}$, we can take a mean parameterization $p_{\mu_k} = p_{(\psi^*,\nabla\psi(\mu_k))}$ which takes the form

$$p_{\mu_k}(x) = h(x)\exp(-D_\psi(x,\mu_k)). \tag{2.14}$$

We will call this a Bregman distribution associated with ψ.

A graphical model representation of the mixture model is given in figure 2.1. The joint distribution of the observed variables $\mathbf{x} = (x_i)_{i=1..n}$ and hidden variables $\mathbf{z} = (z_i)_{i=1..n}$ factorizes as follows

$$\begin{aligned} p(\mathbf{x},\mathbf{z};\pi,\mu) &= \prod_{i=1}^{n} p(z_i;\pi)p(x_i|z_i;\mu) \\ &= \prod_{i=1}^{n}\left(\prod_{k=1}^{K}\pi_k^{\mathbb{1}(z_i=k)}\right)\left(\prod_{k=1}^{K}p(x_i|z_i=k;\mu_k)^{\mathbb{1}(z_i=k)}\right), \end{aligned} \tag{2.15}$$

where $p(x_i|z_i = k;\mu_k) = p_{\mu_k}(x_i)$.

In order to estimate parameters π and $\mu = (\mu_k)_{k=1..K}$ from the observed data, we can use a maximum likelihood estimator and try to maximize the (log-)likelihood of the observed data, given by

$$\ell(\pi,\mu) = \log p(\mathbf{x};\pi,\mu) = \sum_{i=1}^{n}\log p(x_i;\pi,\mu) = \sum_{i=1}^{n}\log\sum_{k=1}^{K}p(x_i,z_i=k;\pi,\mu). \tag{2.16}$$

Note that like minimizing the K-means objective, maximizing this log-likelihood is a non-convex problem and thus makes it hard to find a global maximum. It is possible nonetheless to find local maxima using ascent methods, which try increase the value of the log-likelihood after each iteration. A standard technique used for maximum likelihood estimation in latent variable models is the Expectation-Maximization (EM) algorithm (Dempster et al., 1977), which iteratively maximizes lower bounds on the likelihood.

Derivation of EM If we denote by $\mathbf{x}$ the set of observed variables, by $\mathbf{z}$ the hidden variables and by θ the set of parameters, we can use Jensen's inequality to find a lower bound on the

log-likelihood in terms of any probability distribution q on $\mathbf{z}$

$$\begin{aligned}\ell(\theta) &= \log \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}; \theta) \\ &= \log \sum_{\mathbf{z}} q(\mathbf{z}) \frac{p(\mathbf{x}, \mathbf{z}; \theta)}{q(\mathbf{z})} \\ &\geq \sum_{\mathbf{z}} q(\mathbf{z}) \log \frac{p(\mathbf{x}, \mathbf{z}; \theta)}{q(\mathbf{z})}. \end{aligned} \tag{2.17}$$

The EM algorithm proceeds by maximizing this lower bound with respect to q and θ, in a coordinate-ascent fasion. Maximizing it with respect to the distribution q can be done by having an equality in Jensen's inequality, thus having a tight bound. The equality case corresponds to having a constant in the expectation, i.e. $q(\mathbf{z}) \propto p(\mathbf{x}, \mathbf{z}; \theta)$, that is

$$q(\mathbf{z}) = p(\mathbf{z}|\mathbf{x}; \theta), \tag{2.18}$$

which is known as the E-step, since it usually corresponds to computing expected sufficient statistics. If we now fix q in (2.17), we can now maximize this new lower bound as a function of θ, which, up to a constant term (the entropy of q), is equal to the expected complete-data likelihood under q

$$\mathbb{E}_{\mathbf{z} \sim q}[\log p(\mathbf{x}, \mathbf{z}; \theta)]. \tag{2.19}$$

This is known as the M-step. This optimization problem can often be solved in closed form, as it resembles a complete-data maximum-likelihood problem. Given an initial parameter $\theta^{(0)}$, the EM algorithm thus repeats the following two steps until convergence

- (E-step) Compute q using current parameter θ

$$q(\mathbf{z}) = p(\mathbf{z}|\mathbf{x}; \theta^{(t)})$$

- (M-step) Update parameter θ

$$\theta^{(t+1)} = \arg\max_{\theta} \mathbb{E}_{\mathbf{z} \sim q}[\log p(\mathbf{x}, \mathbf{z}; \theta)].$$

Since the lower bound 2.17 increases after each step and is tight after each E-step, the log-likelihood increases after each iteration, that is, $\ell(\theta^{(t+1)}) \geq \ell(\theta^{(t)})$, and thus converges, assuming it is bounded above[3]. A good stopping criterion is then to see if the increase in likelihood is small. The rate of convergence of EM is linear[4], but is often preffered in practice to second-order methods because of its simplicity and because it generally suffices to quickly get a good fit to the sample data in typical machine learning applications (Xu and Jordan, 1996).

EM for mixture models In order to derive an EM algorithm for our mixture model, we can start by computing the expected complete-data log-likelihood used in the M-step, following Eq. (2.15)

$$\mathbb{E}_{\mathbf{z} \sim q}[\log p(\mathbf{x}, \mathbf{z}; \pi, \mu)] = \sum_i \sum_k \mathbb{E}_q[\mathbb{1}\{z_i = k\}] \log \pi_k + \sum_i \sum_k \mathbb{E}_q[\mathbb{1}\{z_i = k\}] \log p(x_i|z_i = k; \mu_k), \tag{2.20}$$

[3]Note that this doesn't necessarily imply convergence to a stationary point of the likelihood, nor does it imply the convergence of the iterates. For more details the convergence properties of EM, see, e.g., Dempster et al. (1977); Wu (1983); Xu and Jordan (1996).

[4]This follows from the fact that EM can be seen as a fixed point problem, under a mapping M such that $\theta^{(t+1)} = M(\theta^{(t)})$.

where the $\tau_{ik} := \mathbb{E}_q[\mathbb{1}\{z_i = k\}] = p(z_i = k|x_i)$ are computed using the previous parameter estimates in the E-step as follows

$$\begin{aligned}\tau_{ik} &= p(z_i = k|x_i; \pi, \mu) \\ &= \frac{1}{Z} p(z_i = k; \pi) p(x_i|z_i = k; \mu) \\ &= \frac{1}{Z} \pi_k e^{-D_\psi(x_i, \mu_k)},\end{aligned}$$

where Z is a normalizing constant which ensures $\sum_k \tau_{ik} = 1$ and the last line holds in the specific case of Bregman emission distributions (2.14).

Maximizing (2.20) with respect to π is straightforward, while maximizing with respect to μ_k in the case of a Bregman distribution is equivalent to minimizing

$$\frac{\sum_i \tau_{ik} D_\psi(x_i, \mu_k)}{\sum_i \tau_{ik}},$$

which, by Proposition 2.2, gives

$$\mu_k = \frac{\sum_i \tau_{ik} x_i}{\sum_i \tau_{ik}}.$$

Note that this can also be seen as computing expected sufficient statistics and doing moment-matching (see, e.g., Wainwright and Jordan (2008)). The entire algorithm for mixtures of Bregman distributions, i.e. Bregman soft-clustering, is given in Algorithm 2.2.

Algorithm 2.2: EM algorithm for soft clustering with Bregman divergences.

Data: $x_1, \ldots, x_n$, initial parameters π, μ, Bregman divergence D_ψ.
Result: final parameters π, μ, soft-assignments τ_{ik}
repeat
  `// E-step`
  $\tau_{ik} \leftarrow p(z_i = k|x_i; \pi, \mu) = \frac{1}{Z} \pi_k e^{-D_\psi(x_i, \mu_k)}$
  `// M-step`
  $\pi_k \leftarrow \frac{1}{n} \sum_i \tau_{ik}$
  $\mu_k \leftarrow \frac{\sum_i \tau_{ik} x_i}{\sum_i \tau_{ik}}$
until *convergence*;

2.4 Hidden Markov Models (HMMs)

Mixture models assume that the observations are i.i.d., and therefore using such a model in the case of our audio representation presented in Section 2.1 would ignore the temporal dependencies in the data. Hidden Markov models (HMMs) (Cappé et al., 2005; Rabiner, 1989) are latent variable models in which the latent state variables evolve with Markovian dynamics, and thus are well-suited for modeling data with sequential structure, as is the case for audio.

2.4.1 Model

Let $(x_t)_{t=1..T}$ be our sequence of observations, with $x_t \in \mathbb{R}^p$, and $(z_t)_{t=1..T}$ be our hidden state sequence, where each state is one of K states, that is $z_t \in \{1, \ldots, K\}$. An HMM is a generative latent variable model which can be described by the following generative process, for which a

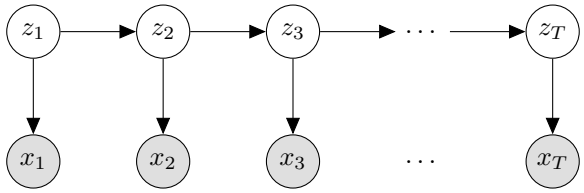


Figure 2.2: Graphical model representation of an HMM.

graphical model representation is given in Figure 2.2

$$\begin{aligned} z_1 &\sim \pi \\ z_t|z_{t-1} = i &\sim A_i, \quad t = 2, \dots, T \\ x_t|z_t = i &\sim p_{\mu_i}, \quad t = 1, \dots, T \end{aligned}$$

where π gives distribution of z_1, $A \in \mathbb{R}_+^{K \times K}$ is a transition matrix such that $A_{ij} = p(z_t = j|z_{t-1} = i)$, $A\mathbf{1} = \mathbf{1}$ and $A_i = (A_{ij})_j$, where $\mathbf{1} = (1, \dots, 1)^\top$. μ_k is the parameter of the k-th emission distribution, which we will consider to be the distribution associated to a Bregman divergence D_ψ, although one can easily extend what follows to other emission distributions such as Gaussians. The joint probability of a sequence $z_{1:T} = (z_1, \dots, z_T)$ and observations $x_{1:T} = (x_1, \dots, x_T)$ is

$$p(x_{1:T}, z_{1:T}; \pi, A, \mu) = p(z_1; \pi) \prod_{t=2}^{T} p(z_t|z_{t-1}; A) \prod_{t=1}^{T} p(x_t|z_t; \mu). \tag{2.21}$$

2.4.2 Inference

The goal of probabilistic inference is to infer hidden variables from observed variables, when the parameters θ of the model are fixed. This generally means computing the posterior distribution $p(\mathbf{z}_Q|\mathbf{x}; \theta)$ over a set of query variables $\mathbf{z}_Q$ among hidden variables. A different form of inference is *maximum a posteriori* (MAP) inference, which aims at finding an assignment over hidden variables with maximum posterior probability, $\mathbf{z}^{MAP} = \arg\max_{\mathbf{z}} p(\mathbf{z}|\mathbf{x}; \theta) = \arg\max_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}; \theta)$.

In HMMs, there are a few inference tasks of interest:

- *Smoothing*: compute $p(z_t|x_{1:T})$ for $t < T$, uses all available observations to compute the marginal of a particular hidden state.
- *Filtering*: compute $p(z_t|x_{1:t})$, which is particularly useful when inference is done online, as is the case for the score-following system *Antescofo*.
- *Prediction*: compute $p(z_t|x_{1:T})$ for $t > T$, useful for predicting future states.
- *MAP inference*: compute the most likely sequence $z_{1:T}^{MAP} = \arg\max_{z_{1:T}} p(z_{1:T}|x_{1:T})$.

We now describe the *forward-backward* and *Viterbi* algorithms, which are the standard algorithms for posterior and MAP inference, respectively.

Forward-Backward algorithm The main difficulty of posterior inference comes from the complicated summations over many latent variables that appear for marginalization and for computing the normalization constants in the target posterior distribution. However, the factorized form of the joint probability in Eq. (2.21) makes it possible to propagate these sums in smaller factors and then propagate back, as is common in standard belief propation or message

passing algorithms (also known as sum-product algorithms) in tree-structured graphical models, making the algorithm much more efficient. In the specific case of HMMs, it is usual to define

$$\alpha_t(i) = p(z_t = i, x_1, \dots, x_t)$$
$$\beta_t(i) = p(x_{t+1}, \dots, x_T | z_t = i).$$

The forward-backward algorithm computes these quantities recursively as follows. If we set $\alpha_1(i) = \pi_i p(x_1|z_1 = i; \mu_i)$, the other α_t can be computed using the forward recursion

$$\begin{aligned}
\alpha_{t+1}(j) &= p(z_{t+1} = j, x_1, \dots, x_{t+1}) \\
&= \sum_j p(z_t = i, z_{t+1} = j, x_1, \dots, x_{t+1}) \\
&= \sum_j p(z_t = i, x_1, \dots, x_t) p(z_{t+1} = j | z_t = i) p(x_{t+1} | z_{t+1} = j) \\
&= \sum_i \alpha_t(i) A_{ij} p(x_{t+1} | z_{t+1} = j; \mu_j).
\end{aligned}$$

Similarly, if we let $\beta_T(i) = 1$, the β_t are computed using the backward recursion

$$\begin{aligned}
\beta_t(i) &= \sum_j p(x_{t+1}, \dots, x_T | z_t = i, z_{t+1} = j) p(z_{t+1} = j | z_t = i) \\
&= \sum_j A_{ij} p(x_{t+1} | z_{t+1} = j; \mu_j) \beta_{t+1}(j).
\end{aligned}$$

The time complexity of the algorithm is $O(TK^2)$. Note that these quantities can be very small, leading to numerical problems in the implementation. The usual way to deal with this problem is to use the logarithms of all the quantities involved, and use the log-sum-exp operation when a summation is involved.

Once the α and β quantities are computed, one can easily obtain the following useful quantities

$$\begin{aligned}
p(z_t = i | x_{1:T}) &= \frac{p(z_t = i, x_{1:t}) p(x_{t+1:T} | z_t = i)}{p(x_{1:T})} = \frac{1}{Z} \alpha_t(i) \beta_t(i) \\
p(z_t = i, z_{t+1} = j | x_{1:T}) &= \frac{1}{Z} \alpha_t(i) A_{ij} p(x_{t+1} | z_{t+1} = j; \mu_j) \beta_{t+1}(j) \\
p(z_t = i | x_{1:t}) &= \frac{1}{Z} \alpha_t(i) \\
p(x_{1:T}) &= \sum_i \alpha_T(i),
\end{aligned}$$

where Z is the appropriate normalization constant in each equation.

Viterbi algorithm In order to compute the MAP estimate $z_{1:T}^{MAP} = \arg\max_{z_{1:T}} p(z_{1:T}|x_{1:T})$, we use a similar recursion to the forward recursion, but where we replace $\sum_i$ by $\max_i$. This is known as the max-product or Viterbi algorithm. Let's define

$$\gamma_t(i) = \max_{z_1, \dots, z_{t-1}} p(z_1, \dots, z_{t-1}, z_t = i, x_1, \dots, x_t).$$

If we let $\gamma_1(i) = \pi_i p(x_1|z_1 = i; \mu_i)$, we have the recursion

$$\gamma_{t+1}(j) = \max_i \gamma_t(i) A_{ij} p(x_{t+1} | z_{t+1} = j; \mu_j). \tag{2.22}$$

By storing backpointers $b_{t+1}(j) = \arg\max_i \gamma_t(i) A_{ij} p(x_{t+1}|z_{t+1} = j; \mu_j)$, we can then recover the MAP sequence $z_{1:T}$ by taking $z_T = \arg\max_i \gamma_T(i)$ and $z_t = b_{t+1}(z_{t+1})$ for $t = T-1, \dots, 1$.

2.4.3 EM algorithm

We now derive an EM algorithm for HMMs, allowing us to estimate good parameters $\theta = (\pi, A, \mu)$ from data. We start by computing the complete log-likelihood from Eq. (2.21)

$$\begin{aligned}\ell(\theta) &= \log p(x_{1:T}, z_{1:T}; \theta) \\ &= \sum_i \mathbb{1}\{z_1 = i\} \log \pi_i + \sum_{t\geq 2} \sum_i \sum_j \mathbb{1}\{z_{t-1} = i, z_t = j\} \log A_{ij} + \sum_{t\geq 1} \sum_i \mathbb{1}\{z_t = i\} \log p(x_t | z_t = i; \mu_i).\end{aligned}$$

The E-step thus consists in computing the expected sufficient statistics $p(z_t = i | x_{1:T}; \theta^{(k)})$ and $p(z_{t-1} = i, z_t = j | x_{1:T}; \theta^{(k)})$ given current parameter $\theta^{(k)}$, which is an inference task and can be solved with the forward-backward algorithm. The M-step then uses these expected sufficient statistics to maximize the expected complete log-likelihood and obtain a new parameter $\theta^{(k+1)} = \arg\max_{\theta'} \mathbb{E}_z[\ell(\theta') | x_{1:T}; \theta^{(k)}]$. The resulting algorithm is outlined in Algorithm 2.3. Note that because the likelihood is non-convex, initialization plays a big role, and emissions are often initialized using the results of K-means or a mixture model.

Algorithm 2.3: EM algorithm for HMMs.

Data: $x_{1:T}$, initial parameters $\theta = (\pi, A, \mu)$, Bregman divergence D_ψ.
Result: final parameters $\theta = (\pi, A, \mu)$.
repeat
  `// E-step`
  $\alpha, \beta \leftarrow$ `ForwardBackward(`$x_{1:T}, \theta$`)`
  $\tau_t(i) \leftarrow p(z_t = i | x_{1:T}; \theta) = \frac{1}{Z} \alpha_t(i) \beta_t(i)$
  $\tau_t(i,j) \leftarrow p(z_{t-1} = i, z_t = j | x_{1:T}; \theta) = \frac{1}{Z} \alpha_{t-1}(i) A_{ij} p(x_t | z_t = j; \mu_j) \beta_t(j)$
  `// M-step`
  $\pi_i \leftarrow \tau_1(i)$
  $A_{ij} \leftarrow \frac{\sum_{t\geq 2} \tau_t(i,j)}{\sum_{j'} \sum_{t\geq 2} \tau_t(i,j')}$
  $\mu_i \leftarrow \frac{\sum_{t\geq 1} \tau_t(i) x_i}{\sum_{t\geq 1} \tau_t(i)}$
until *convergence*;

2.5 Hidden Semi-Markov Models (HSMMs)

One drawback of HMMs is their limited capability for expressing segment durations, that is the number of time steps during which the hidden state stays the same. In fact, if we consider the prior probability – ignoring observations – of staying in a given state i for exactly d time steps, it is equal to $A_{ii}^{d-1}(1 - A_{ii})$, that is, the duration distribution for each state is geometric. Although the parameters A_{ii} can be estimated to give a good fit to the data in many applications, the choice of a geometric distribution can be quite restrictive in some cases.

Hidden Semi-Markov Models (HSMMs) allow us to overcome this problem by explicitely modeling the segment durations with any distribution (Murphy, 2002; Guédon, 2003).

2.5.1 Model

An HSMM[5] can be thought of as a generalization of an HMM where each hidden state corresponds to a segment, with both a state variable z and a segment length l sampled from the

[5]Several different types of HSMMs exist, see (Yu, 2010) for details. We will focus on the "explicit duration HMM", as described in (Murphy, 2002; Guédon, 2003)

Distribution	**pmf**	**parameter**	**mean**
Poisson	$\frac{\lambda^k e^{-\lambda}}{k!}$	λ	λ
Negative Binomial	$\binom{k+r-1}{k} p^r (1-p)^k$	r, p	$\frac{(1-p)r}{p}$

Table 2.1: Duration distributions

duration distribution $p_z(l)$, and where l observations are drawn independently from the emission distribution in state z. The transition matrix A now represents transitions between segments, and we will consider it to only depend on state variables. This HMM representation isn't tractable, since it would require having KD possible hidden states, where D is the maximum segment length, and using segments as states stops us from processing observations one by one. Instead, we follow Murphy (2002) by introducing deterministic transitions and "finish nodes": z_t^D will denote a counter of the number of transitions left until the end of the segment (for a segment of length d, z_t^D will go from d to 1, deterministically), and the finish variable f_t will stay equal to 0 inside a segment, and turn to 1 when the end of the segment is reached, i.e. $f_t = 1$ if and only if $z_t^D = 1$. We then have

$$\begin{aligned} p(z_t = j | z_{t-1} = i, f_{t-1} = f) &= \begin{cases} \delta(i,j), & \text{if } f = 0 \\ A_{ij}, & \text{if } f = 1 \text{ (transition)} \end{cases} \\ p(z_t^D = d | z_t = i, f_{t-1} = 1) &= p_i(d) \\ p(z_t^D = d | z_t = i, z_{t-1}^D = d' \geq 2) &= \delta(d, d' - 1), \end{aligned}$$

where δ is the Dirac delta ($\delta(i,j) = 1$ if $i = j$ and 0 otherwise). Note that having $d' = 1$ in the last identity would give the previous identity, since the events $\{z_{t-1}^D = 1\}$ and $\{f_{t-1} = 1\}$ are the same. The rest of the model is similar to the HMM:

$$\begin{aligned} z_1 &\sim \pi \\ x_t | z_t = i &\sim p_{\mu_i}. \end{aligned}$$

Duration distributions The distributions $p_i(d)$ can either be nonparametric tabular distributions described by each entry, or parametric distributions such as Poisson or negative binomial. Table 2.1 shows the probability mass function, parameters and means for these parametric distributions.

Complete likelihood If we consider a hidden sequence $z_{1:T}$ given by a segment description $(t_i, l_i, q_i)_{i=1..\tau}$ where τ is the number of segments, t_i is the start of each segment, l_i its length and q_i its state, with $t_1 = 1$, $t_{i+1} = t_i + l_i$, $\sum_{i=1}^{\tau} l_i = T$, the joint probability of $z_{1:T}$ and $x_{1:T}$ is

$$p(x_{1:T}, z_{1:T}; \theta) = \prod_{i=1}^{\tau} \left(p_{q_i}(l_i) p(q_i | q_{i-1}; \pi, A) \prod_{t=t_i}^{t_i + l_i - 1} p(x_t | z_t = q_i; \mu) \right), \tag{2.23}$$

where $\theta = (\pi, A, \mu, (p_j)_j)$ is the set of parameters.

2.5.2 Inference

Forward-backward algorithm As in the HMM case, we can derive a forward-backward algorithm using the following quantities:

$$\begin{aligned}\alpha_t(j) &= p(z_t = j, f_t = 1, x_{1:t}) \\ \alpha_t^*(j) &= p(z_{t+1} = j, f_t = 1, x_{1:t}) \\ \beta_t(i) &= p(x_{t+1:T}|z_t = i, f_t = 1) \\ \beta_t^*(i) &= p(x_{t+1:T}|z_{t+1} = i, f_t = 1).\end{aligned}$$

If we define $\alpha_0^*(j) = \pi_j$, the following forward recursions are obtained by marginalizing respectively on the length of the current segment and the state of the previous segment:

$$\begin{aligned}\alpha_t(j) &= \sum_d p(x_{t-d+1:t}|j)p_j(d)\alpha_{t-d}^*(j) \\ \alpha_t^*(j) &= \sum_i \alpha_t(i)A_{ij},\end{aligned}$$

where we have, in our case, $p(x_{t-d+1:t}|j) = \prod_{u=t-d+1}^{t} p(x_u|z_u = j; \mu_j)$. The backward recursions are similarly obtained by marginalizing respectively on the state of the next segment and the length of the current segment:

$$\begin{aligned}\beta_t(i) &= \sum_j \beta_t^*(j)A_{ij} \\ \beta_t^*(i) &= \sum_d \beta_{t+d}(i)p_i(d)p(x_{t+1:t+d}|i),\end{aligned}$$

with $\beta_T(i) = 1$. The time complexity of the algorithm is $O(TK^2D)$ when we force segments to be of length at most D. Note that in these recursions we assumed that the first segment starts exactly at time $t = 1$ and the last segment ends at time $t = T$. This is known as an *uncensored* formulation. In contrast, if we allow the last segment to end after $t = T$, this is known as *right-censoring* (Guédon, 2003), and the backward recursion on β^* becomes

$$\beta_t^*(i) = \sum_{d=1}^{T-t} \beta_{t+d}(i)p_i(d)p(x_{t+1:t+d}|i) + D_i(T - t + 1)p(x_{t+1:T}|i),$$

where $D_i(d) := \sum_{d' \geq d} p_i(d')$ is the survivor function of the segment length, that is $D_i(d) = p(l \geq d|z = i)$. The last term is thus a right-censoring term which accounts for the case of an unfinished segment at the end of the sequence. A left-censored formulation can be obtained by adding a similar term to the forward recursion, allowing initial segments to start before time $t = 1$.

Expected sufficient statistics We can now compute the expected sufficient statistics for transitions across segment boundaries and durations

$$\begin{aligned}p(z_t = i, z_{t+1} = j, f_t = 1|x_{1:T}) &= \frac{1}{Z}\alpha_t(i)A_{ij}\beta_t^*(j) \\ p(l_t = d, z_t = i, f_t = 1|x_{1:T}) &= \frac{1}{Z}\alpha_{t-d}^*(i)p_i(d)p(x_{t-d+1:t}|i)\beta_t(i)\end{aligned}$$

In order to compute $p(z_t = i|x_{1:T})$, we follow Murphy (2002) and define

$$\begin{aligned}\gamma_t(i) &= p(z_t = i, f_t = 1|x_{1:T}) = \frac{1}{Z}\alpha_t(i)\beta_t(i) \\ \gamma_t^*(i) &= p(z_{t+1} = i, f_t = 1|x_{1:T}) = \frac{1}{Z}\alpha_t^*(i)\beta_t^*(i).\end{aligned}$$

We then have

$$p(z_t = i|x_{1:T}) = \sum_{\tau<t}(\gamma^*_\tau(i) - \gamma_\tau(i)), \tag{2.24}$$

which follows from the identity $p(z_{t+1} = i|x_{1:T}) = p(z_t = i|x_{1:T}) + p(z_{t+1} = i, f_t = 1|x_{1:T}) - p(z_t = i, f_t = 1|x_{1:T})$. See (Murphy, 2002; Guédon, 2003; Rabiner, 1989) for details.

If we are in an online filtering task where the quantities $p(z_t = i|x_{1:t})$ are needed, we can use the following:

$$p(z_t = i, x_{1:t}) = \sum_d \alpha^*_{t-d}(i) D_i(d) p(x_{t-d+1:t}|i),$$

which is obtained by marginalizing over the start of the segment, and where $D_i(d)$ appears because the segment is possibly unfinished (Guédon, 2003). The $p(z_t = i|x_{1:t})$ are then obtained by normalization.

Viterbi algorithm If we define

$$\begin{aligned}
\delta_t(j) &= \max_{z_{1:t-1}} p(z_{1:t-1}, z_t = j, f_t = 1, x_{1:t}) \\
\delta^*_t(j) &= \max_{z_{1:t}} p(z_{1:t}, z_{t+1} = j, f_t = 1, x_{1:t}),
\end{aligned}$$

then if $\delta^*_0(j) = \pi_j$, we have the Viterbi recursion

$$\begin{aligned}
\delta_t(j) &= \max_d p(x_{t-d+1:t}|j) p_j(d) \delta^*_{t-d}(j) \\
b_t(j) &= \arg\max_d p(x_{t-d+1:t}|j) p_j(d) \delta^*_{t-d}(j) \\
\delta^*_t(j) &= \max_i \delta_t(i) A_{ij} \\
b^*_t(j) &= \arg\max_i \delta_t(i) A_{ij},
\end{aligned}$$

where b_t, b^*_t are backpointers which can be used to recover the MAP sequence $z_{1:T}$ segment by segment as follows: initialize $t \leftarrow T$, $i \leftarrow \arg\max_j \delta_t(j)$, then repeat the steps $d \leftarrow b_t(i)$, $z_{t-d+1:t} \leftarrow i$, $t \leftarrow t - d$, $i \leftarrow b^*_t(i)$ until the beginning of the sequence is reached.

2.5.3 EM algorithm

The E-step of the EM algorithm computes the expected sufficient statistics using the forward-backward algorithm as described above: $p(z_t = i|x_{1:T})$ for emissions, $p(z_t = i, z_{t+1} = j, f_t = 1|x_{1:T})$ for transitions, and $p(l_t = d, z_t = i, f_t = 1|x_{1:T})$ for duration distributions if they need to be learned.

The M-step updates are as follows:

$$\begin{aligned}
\pi_i &= p(z_1 = i|x_{1:T}) \\
A_{ij} &= \frac{\sum_t p(z_t = i, z_{t+1} = j, f_t = 1|x_{1:T})}{\sum_{j'} \sum_t p(z_t = i, z_{t+1} = j', f_t = 1|x_{1:T})} \\
\mu_i &= \frac{\sum_t p(z_t = i|x_{1:T}) x_t}{\sum_t p(z_t = i|x_{1:T})}
\end{aligned}$$

A tabular duration distribution can be estimated easily from the sufficient statistics:

$$\hat{p}_i(d) = \frac{\sum_t p(l_t = d, z_t = i, f_t = 1|x_{1:T})}{\sum_{d'} \sum_t p(l_t = d', z_t = i, f_t = 1|x_{1:T})}.$$

However having such a nonparametric distribution can be impractical because of the large number of parameters that need to be learned. The parameter λ of a Poisson duration distribution can be estimated from the mean of the estimated distribution

$$\hat{\lambda}_i = \sum_d d\hat{p}_i(d).$$

In the case of a negative binomial distribution with fixed r, the parameter p can be estimated as follows:

$$\hat{p}_i = \frac{r}{r + \sum_d d\hat{p}_i(d)}.$$

2.6 Summary and discussion

In this chapter, we introduced a framework for the representation of audio signals through short-time Fourier coefficients, and their modeling with Bregman divergences, or the corresponding regular exponential family distributions in a probabilistic setting. We then developed algorithms for clustering these signals into homogeneous groups, starting with time-independent algorithms such as K-means and EM for mixture models, followed by sequential models such as HMMs and HSMMs, which explicitly model temporal dependencies and are thus more adapted to segmentation tasks.

One issue with these models is that the number of clusters K needs to be fixed in advance, which isn't always an obvious task, e.g. in an musical structure segmentation task where the number of structures present (like notes and chords) isn't known in advance. We briefly discussed a way to add new states in the K-means algorithm. In the probabilistic setting, this can be achieved by placing Bayesian nonparametric priors on the emission distributions, such as the Dirichlet process or the hierarchical Dirichlet process. These methods have been applied to HMMs (Teh et al., 2006; Fox et al., 2008) and HSMMs (Johnson and Willsky, 2013), but usually require expensive sampling procedures for Bayesian inference, for which convergence is hard to assess.

In the next chapter, we will develop online algorithms for these models, both as a way to accelerate learning and as a way to run the clustering and segmentation algorithms in a real-time context.

Chapter 3

Online algorithms

In this chapter, we will cover various algorithms for online learning in sequential models such as HMMs and HSMMs. Online learning has had many recent successes, and has been shown to often outperform batch learning, especially in large-scale settings (see, e.g., (Bottou and Bousquet, 2008)). In the context of latent variable models, online EM algorithms have been shown to improve performance, both in terms of speed and accuracy, compared to batch EM (Liang and Klein, 2009). These algorithms have mainly been used for i.i.d. data. An extension of the online EM of Cappé and Moulines (2009) to HMMs has been given in (Cappé, 2011), and we give a further extension to HSMMs in Section 3.1. We derive alternative online algorithms based on incremental minorization-maximization schemes (similar to, e.g., (Mairal et al., 2010; Mairal, 2014)), both in a non-probabilistic setting in Section 3.2 and a probabilistic one in Section 3.3, giving a new incremental EM similar to that of Neal and Hinton (1998) for sequential models.

3.1 Online EM

In this section, we present the online EM algorithm of Cappé (2011) for HMMs, which is based on similar ideas to the approach of Cappé and Moulines (2009) for independent observations, which we briefly present. Let's consider the case of a batch EM algorithm on a latent variable model with independent observations $(x_i)_{i=1..n}$ where the complete-data model $p(x, z; \theta)$ lies in the exponential family and takes the form

$$p(x, z; \theta) = h(x, z) \exp(\langle s(x, z), \eta(\theta) \rangle - a(\theta)). \tag{3.1}$$

The batch EM algorithm at iteration t then takes the form

$$\begin{aligned} S_t &= \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_z[s(x_i, z_i) | x_i; \theta_{t-1}] \\ \theta_t &= \bar{\theta}(S_t), \end{aligned}$$

where $\bar{\theta}(s) := \arg\max_\theta \langle s, \eta(\theta) \rangle - a(\theta)$ gives the complete-data maximum likelihood estimate from sufficient statistics s, which is assumed to be unique. In the case of Bregman emissions, the complete-data sufficient statistics $s(x, z)$ are given by the vectors $s^0_k(x, z) = \mathbb{1}\{z = k\}$ and $s^1_k(x, z) = \mathbb{1}\{z = k\}x$ for every mixture component k, so that the parameter update $\bar{\theta}$ is given by $\pi_k = S^0_{t,k}$ and $\mu_k = S^1_{t,k}/S^0_{t,k}$, where $S^{0/1}_{t,k}$ are the sums in the E-step corresponding to $s^{0/1}_k$.

By taking the limit $n \to \infty$, the empirical average in the E-step converges to the expectation with respect to the true distribution P of the observations (assuming $x_1, \ldots, x_n \overset{iid}{\sim} P$), and we

obtain the following *limiting EM* recursion

$$S_t = \mathbb{E}_{x\sim P}[\mathbb{E}_z[s(x,z)|x;\theta_{t-1}]]$$
$$\theta_t = \bar{\theta}(S_t).$$

The online EM algorithm of Cappé and Moulines (2009) then tries to approach this limiting recursion by using a stochastic approximation (a.k.a. Robbins-Monro) procedure on the space of sufficient statistics, with an appropriate sequence of step sizes $(\gamma_t)_t$, giving rise to the recursion

$$\hat{s}_t = (1-\gamma_t)\hat{s}_{t-1} + \gamma_t \,\mathbb{E}_z[s(x_t,z)|x_t;\hat{\theta}_{t-1}]$$
$$\hat{\theta}_t = \bar{\theta}(\hat{s}_t).$$

We now describe the online EM algorithm of Cappé (2011) for HMMs, which relies on a special form of smoothing called forward smoothing in order to compute sums of expected sufficient statistics in HMMs, and we will then adapt this algorithm to HSMMs by formulating them as a dynamic bayesian network with two hidden variables, which allows us to maintain a markovian assumption.

3.1.1 Online EM for HMMs

We consider an HMM, defined as in Section 2.4, where the final time T may be unbounded in the online case. The initial distribution π of z_0 is consider fixed as it cannot be estimated accurately from a single observation sequence (as explained in (Cappé, 2011; Cappé et al., 2005)). We can then fully describe the model from the complete-data distribution at time t conditioned on the hidden state at time $t-1$, $p(x_t, z_t|z_{t-1};\theta)$, which we assume to be in the exponential family, just as in Eq. (3.1):

$$p(x_t,z_t|z_{t-1};\theta) = h(z_t,x_t)\exp(\langle s(z_{t-1},z_t,x_t),\eta(\theta)\rangle - a(\theta)). \tag{3.2}$$

The batch EM algorithm at iteration k then takes the form

$$S_k = \frac{1}{T}\,\mathbb{E}_z\left[\sum_{t=1}^{T} s(z_{t-1},z_t,x_t) \;\middle|\; x_{0:T};\theta_{k-1}\right]$$
$$\theta_k = \bar{\theta}(S_k),$$

where $\bar{\theta}(s) := \arg\max_\theta \langle s,\eta(\theta)\rangle - a(\theta)$ is assumed unique. In the HMM with Bregman emissions, we can consider the vector $s(z_{t-1},z_t,x_t)$ to be given by $s_{ij}(z_{t-1},z_t,x_t) = \mathbb{1}\{z_{t-1}=i, z_t=j\}$, $s_i^0(z_{t-1},z_t,x_t) = \mathbb{1}\{z_t=i\}$ and $s_i^1(z_{t-1},z_t,x_t) = \mathbb{1}\{z_t=i\}x_t$ for $i,j\in\{1,\dots,K\}$ and the parameter update then becomes:

$$A_{ij} = \frac{S_k(i,j)}{\sum_{j'} S_k(i,j')} \tag{3.3}$$
$$\mu_i = \frac{S_k^1(i)}{S_k^0(i)}, \tag{3.4}$$

where $S_k(i,j), S_k^{0/1}(i)$ correspond to the E-step expected sums using s_{ij} and $s_i^{0/1}$, respectively.

Cappé (2011) shows that under certain stationarity assumptions on the observed sequence and (strong) forgetting assumptions on the model, the sum in the E-step converges to an expectation, and we obtain a *limiting EM* recursion

$$S_k = \mathbb{E}_{x\sim P}[\mathbb{E}_z[s(z_{-1},z_0,x_0)|x_{-\infty:\infty};\theta_{k-1}]]$$
$$\theta_k = \bar{\theta}(S_k),$$

which can be approximated by the online EM algorithm we describe below, although a full proof of convergence is still lacking.

Forward smoothing recursion

The sums of expected sufficient statistics from the E-step are usually computed using a forward-backward algorithm, which isn't suitable in an online setting. In contrast, the online EM approach of (Cappé, 2011) is inspired by a forward-only smoothing recursion, which allows to recursively update the sum of interest as new observations come in. Let's define

$$\begin{aligned} S_t &= \frac{1}{t}\,\mathbb{E}_z\left[\sum_{t'=1}^{t} s(z_{t'-1}, z_{t'}, x_{t'}) \;\middle|\; x_{0:t}; \theta\right] \\ \phi_t(i) &= p(z_t = i | x_{0:t}) \\ \rho_t(i) &= \frac{1}{t}\,\mathbb{E}_z\left[\sum_{t'=1}^{t} s(z_{t'-1}, z_{t'}, x_{t'}) \;\middle|\; x_{0:t}, z_t = i; \theta\right]. \end{aligned}$$

We then have $S_t = \sum_i \rho_t(i)\phi_t(i)$. If we initialize $\phi_0(i) = \frac{1}{Z}\pi_i p(x_0|z_0 = i)$ and $\rho_0(i) = 0$, the forward smoothing recursion is given by the following:

$$\begin{aligned} \phi_{t+1}(j) &= \frac{1}{Z}\sum_i \phi_t(i) A_{ij} p(x_{t+1}|z_{t+1} = j) \\ \rho_{t+1}(j) &= \frac{1}{t+1}\,\mathbb{E}\left[\sum_{t'=1}^{t+1} s(z_{t'-1}, z_{t'}, x_{t'}) \;\middle|\; x_{0:t+1}, z_{t+1} = j; \theta\right] \\ &= \frac{1}{t+1}\sum_i \mathbb{E}\left[\sum_{t'=1}^{t+1} s(z_{t'-1}, z_{t'}, x_{t'}) \;\middle|\; x_{0:t+1}, z_{t+1} = j, z_t = i; \theta\right] p(z_t = i | z_{t+1} = j, x_{0:t}) \\ &= \sum_i \left(\frac{1}{t+1} s(i, j, x_{t+1}) + \left(1 - \frac{1}{t+1}\right)\rho_t(i)\right) p(z_t = i | z_{t+1} = j, x_{0:t}), \end{aligned} \tag{3.5}$$

where we used the conditional independence properties of HMMs: $p(z_{t'-1}, z_{t'}|z_t = i, z_{t+1} = j, x_{0:t+1}) = p(z_{t'-1}, z_{t'}|z_t = i, x_{0:t})$ for $t' \leq t$ and $p(z_t = i|z_{t+1} = j, x_{0:t+1}) = p(z_t = i|z_{t+1} = j, x_{0:t})$. This *backward retrospective probability* can be expressed in terms of ϕ as

$$p(z_t = i | z_{t+1} = j, x_{0:t}) = \frac{1}{Z}\phi_t(i) A_{ij}.$$

Online EM algorithm

The idea of Cappé (2011) is to perform the M-step for updating the parameters after each iteration in the forward recursion (starting after a minimal number $t_{\min}$ of iterations), leading to an online algorithm which isn't restricted to finite-length sequences. The recursion update (3.5) becomes a stochastic approximation update of sufficient statistics similar to the update in Cappé and Moulines (2009), and the step size $1/t$ can be replaced by different decreasing sequences $(\gamma_t)_t$ satisfying the usual requirements $\sum_t \gamma_t = \infty$ and $\sum_t \gamma_t^2 < \infty$.

The online EM algorithm is as follows, given an initial parameter $\hat{\theta}_0$.

Initialization

$$\begin{aligned} \hat{\phi}_0(i) &= \frac{1}{Z}\pi_i p(x_0|z_0 = i; \hat{\theta}_0) \\ \hat{\rho}_0(i) &= 0, \end{aligned}$$

The initialization of $\hat{\rho}$ should be replaced by the emission sufficient statistics of x_0 for the corresponding elements in $\hat{\rho}_0$.

Online EM recursion For $t = 1, 2, \ldots$

- Stochastic Approximation E-step

$$\hat{\phi}_{t+1}(j) = \frac{1}{Z} \sum_i \hat{\phi}_t(i) A_{ij}^{(t)} p(x_{t+1}|z_{t+1} = j; \hat{\theta}_t)$$
$$\hat{\rho}_{t+1}(j) = \sum_i \left(\gamma_{t+1} s(i, j, x_{t+1}) + (1 - \gamma_{t+1})\hat{\rho}_t(i)\right) \hat{r}_{t+1}(i|j), \tag{3.6}$$

  where $\hat{r}_{t+1}(i|j) = \frac{1}{Z}\hat{\phi}_t(i)A_{ij}^{(t)}$ approximates the backward retrospective probability.

- M-step

$$\hat{\theta}_{t+1} = \begin{cases} \bar{\theta}\left(\sum_i \hat{\rho}_{t+1}(i)\hat{\phi}_{t+1}(i)\right), & \text{if } t \geq t_{\min} \\ \hat{\theta}_t, & \text{otherwise.} \end{cases}$$

Example 3.1. We saw in Eq. (3.3, 3.4) how parameters $\theta = (A, \mu)$ can be updated in a batch EM algorithm (at iteration k) for HMMs with Bregman emission distributions using the quantities

$$S_T^A(i, j) = \frac{1}{T}\mathbb{E}_z\left[\sum_{t=1}^T \mathbb{1}\{z_{t-1} = i, z_t = j\} \;\middle|\; x_{0:T}; \hat{A}^{(k)}, \hat{\mu}^{(k)}\right]$$
$$S_T^{\mu,\nu}(i) = \frac{1}{T}\mathbb{E}_z\left[\sum_{t=1}^T \mathbb{1}\{z_t = i\} s^\nu(x_t) \;\middle|\; x_{0:T}; \hat{A}^{(k)}, \hat{\mu}^{(k)}\right],$$

with $s^0(x) = 1$ and $s^1(x) = x$, and we will write $s(x) = (s^0(x), s^1(x)^\top)^\top$. S_T^A and $S_T^\mu := (S_T^{\mu,0}, S_T^{\mu,1\top})^\top$ can be obtained using the forward smoothing recursion by introducing the ρ quantities

$$\rho_T^A(i, j, k) = \frac{1}{T}\mathbb{E}_z\left[\sum_{t=1}^T \mathbb{1}\{z_{t-1} = i, z_t = j\} \;\middle|\; x_{0:T}, z_T = k; \hat{A}^{(k)}, \hat{\mu}^{(k)}\right]$$
$$\rho_T^\mu(i, k) = \frac{1}{T}\mathbb{E}_z\left[\sum_{t=1}^T \mathbb{1}\{z_t = i\} s(x_t) \;\middle|\; x_{0:T}, z_T = k; \hat{A}^{(k)}, \hat{\mu}^{(k)}\right].$$

The sums of interest are then given by

$$S_T^A(i, j) = \sum_k \rho_T^A(i, j, k)\phi_T(k)$$
$$S_T^\mu(i, k) = \sum_k \rho_T^\mu(i, k)\phi_T(k),$$

where $\phi_T(k) = p(z_T = k|x_{0:T})$. In the online EM algorithm, the ρ quantities at time $t + 1$ are estimated recursively from those at time t:

$$\begin{aligned}\hat{\rho}_{t+1}^A(i, j, k') &= \gamma_{t+1} \sum_k \mathbb{1}\{k = i, k' = j\}\hat{r}_{t+1}(k|k') + (1 - \gamma_{t+1}) \sum_k \hat{\rho}_t(i, j, k)\hat{r}_{t+1}(k|k') \\ &= \gamma_{t+1}\delta(j, k')\hat{r}_{t+1}(i|j) + (1 - \gamma_{t+1}) \sum_k \hat{\rho}_t^A(i, j, k)\hat{r}_{t+1}(k|k') \\ \hat{\rho}_{t+1}^\mu(i, k') &= \gamma_{t+1}\delta(i, k')s(x_{t+1}) + (1 - \gamma_{t+1}) \sum_k \hat{\rho}_t^\mu(i, k)\hat{r}_{t+1}(k|k'),\end{aligned}$$

where δ denotes the Kronecker delta and $\hat{r}_{t+1}$ the approximate backward retrospective probability. These updates require a time complexity of $O(K^4 + K^3 p)$ for each observation. Estimates

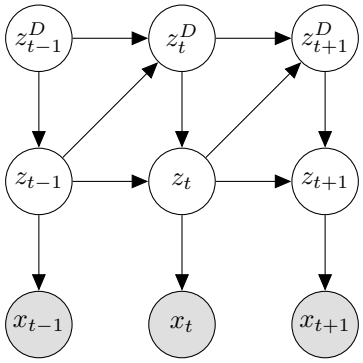


Figure 3.1: Graphical model representation of the HSMM model used for online EM.

of the desired sums $\hat{S}^A$ and $\hat{S}^\mu$ can then be computed as before from $\hat{\rho}$ and $\hat{\phi}$, and can be used for online parameter updates (M-step)

$$\hat{A}_{ij}^{(t+1)} = \frac{\hat{S}_t^A(i,j)}{\sum_{j'} \hat{S}_t^A(i,j')}$$
$$\hat{\mu}_i^{(t+1)} = \frac{\hat{S}_t^{\mu,1}(i)}{\hat{S}_t^{\mu,0}(i)}.$$

3.1.2 Online EM for HSMMs

Model

In order to derive an online EM algorithm for the HSMM, we introduce a new parameterization, different from the one presented in Section 2.5. We consider the HSMM as a dynamic Bayesian network with two hidden variables, z_t and z_t^D, which correspond, respectively, to the current state and the length of the current segment up until time t (as opposed to the time left to the end of the segment in §2.5.1). That is, the event $\{z_t = i, z_t^D = d\}$ means that the chain was in state i for the past d states, and we must have $z_{t-d+1:t} = i$ and $z_{t-u}^D = d - u$ for $0 \leq u \leq d-1$. The transitions between segments are thus given by $A_{ij} = p(z_t = j | z_{t-1} = i, z_t^D = 1)$. We recall the definition of the survivor function, $D_i(d) := \sum_{d' \geq d} p_i(d')$, which is the probability of having a segment duration of at least d in state i.

The transition between time $t-1$ and time t can be encoded in the following conditional probability distributions:

$$p(z_t = j | z_{t-1} = i, z_t^D = d) = \begin{cases} A_{ij}, & \text{if } d = 1 \\ \delta(i,j), & \text{otherwise} \end{cases}$$
$$p(z_t^D = d' | z_{t-1} = i, z_{t-1}^D = d) = \begin{cases} \frac{D_i(d+1)}{D_i(d)}, & \text{if } d' = d+1 \\ 1 - \frac{D_i(d+1)}{D_i(d)}, & \text{if } d' = 1 \\ 0, & \text{otherwise.} \end{cases}$$

A graphical model representation is given in Figure 3.1. The probability of having a segment in state i of length d is then:

$$\frac{D_i(2)}{D_i(1)} \frac{D_i(3)}{D_i(2)} \cdots \frac{D_i(d)}{D_i(d-1)} \left(1 - \frac{D_i(d+1)}{D_i(d)}\right) = D_i(d) - D_i(d+1) = p_i(d), \tag{3.7}$$

since $D_i(1) = 1$. Note that this parameterization of the HSMM inherently includes the right-censoring formulation of Guédon (2003). Indeed, the probability that the last segment ends with

$z_t^D = d$ in state i is

$$\frac{D_i(2)}{D_i(1)}\frac{D_i(3)}{D_i(2)}\cdots\frac{D_i(d)}{D_i(d-1)} = D_i(d).$$

The emission densities are assumed to only depend on the current state z_t and are given by $p(x_t|z_t = k)$. The distribution of the initial state z_0 is π and $z_0^D = 1$ with probability one. The complete likelihood of the model with finite horizon T is thus

$$p(z_{0:T}, z_{0:T}^D, x_{0:T}) = p(z_0)p(z_0^D)\prod_{t\geq 1} p(z_t^D|z_{t-1}, z_{t-1}^D)p(z_t|z_{t-1}, z_t^D)p(x_t|z_t).$$

The exponential family assumption for online EM is now on the model $p(z_t, z_t^D, x_t|z_{t-1}, z_{t-1}^D)$, and we denote its sufficient statistics vector by $s(z_{t-1}, z_{t-1}^D, z_t, z_t^D, x_t)$.

Forward smoothing recursion

Since we consider the joint state (z_t, z_t^D), our model satisfies the same conditional independence properties required for the forward smoothing recursion in the HMM. We have the following forward recursion:

$$\begin{aligned}
\alpha_t(i,d) &:= p(z_t = i, z_t^D = d, x_{0:t}) \\
\alpha_{t+1}(j,1) &= \sum_d \sum_i \alpha_t(i,d)\left(1 - \frac{D_i(d+1)}{D_i(d)}\right) A_{ij} p(x_{t+1}|z_{t+1} = j) \\
\alpha_{t+1}(j,d) &= \alpha_t(j,d-1)\frac{D_j(d)}{D_j(d-1)} p(x_{t+1}|z_{t+1} = j), \qquad \text{if } d \geq 2.
\end{aligned}$$

A similar recursion can be obtained for the forward filter

$$\phi_t(i,d) := p(z_t = i, z_t^D = d|x_{0:t}) = \frac{\alpha_t(i,d)}{\sum_j \sum_{d'} \alpha_t(j,d')}$$

by normalizing after each update, so that $\sum_i \sum_d \phi_t(i,d) = 1$.

The goal is now to compute the sum

$$S_t = \frac{1}{t}\,\mathbb{E}_z\left[\sum_{t'=1}^{t} s(z_{t'-1}, z_{t'-1}^D, z_{t'}, z_{t'}^D, x_{t'}) \;\middle|\; x_{0:t}; \theta\right] \tag{3.8}$$

recursively. Let's define

$$\rho_t(i,d) = \frac{1}{t}\,\mathbb{E}_z\left[\sum_{t'=1}^{t} s(z_{t'-1}, z_{t'-1}^D, z_{t'}, z_{t'}^D, x_{t'}) \;\middle|\; x_{0:t}, z_t = i, z_t^D = d; \theta\right].$$

We then have $S_t = \sum_i \sum_d \rho_t(i,d)\phi_t(i,d)$. The forward smoothing recursion is as follows.

Initialization

$$\begin{aligned}
\phi_0(i,1) &= \frac{\pi_i p(x_0|z_0 = i)}{\sum_i \pi_i p(x_0|z_0 = i)} \\
\phi_0(i,d) &= 0, \qquad \text{if } d \geq 2 \\
\rho_0(i,d) &= 0.
\end{aligned}$$

Recursion

$$\tilde{\phi}_{t+1}(j,1) = \sum_d \sum_i \phi_t(i,d) \left(1 - \frac{D_i(d+1)}{D_i(d)}\right) A_{ij} p(x_{t+1}|z_{t+1}=j)$$
$$\tilde{\phi}_{t+1}(j,d) = \phi_t(j,d-1)\frac{D_j(d)}{D_j(d-1)} p(x_{t+1}|z_{t+1}=j), \quad \text{if } d \geq 2$$
$$\phi_{t+1}(i,d) = \frac{\tilde{\phi}(i,d)}{\sum_j \sum_{d'} \tilde{\phi}(j,d')}$$
$$\rho_{t+1}(j,1) = \sum_i \sum_d \left(\frac{1}{t+1} s(i,d,j,1,x_{t+1}) + \left(1 - \frac{1}{t+1}\right)\rho_t(i,d)\right) r_{t+1}(i,d|j,1)$$
$$\rho_{t+1}(j,d) = \frac{1}{t+1} s(j,d-1,j,d,x_{t+1}) + \left(1 - \frac{1}{t+1}\right)\rho_t(j,d-1), \quad \text{if } d \geq 2,$$

where the backward retrospective probabilities are defined by

$$r_{t+1}(i,d|j,d') := p(z_t = i, z_t^D = d | z_{t+1} = j, z_{t+1} = d', x_{0:t}).$$

Note that the summation in the ρ update disappears for $d \geq 2$ since $r_{t+1}(i,d'|j,d) = 1$ if $i = j$ and $d' = d-1$ and $r_{t+1}(i,d'|j,d) = 0$ otherwise. For $d = 1$, we have

$$\begin{aligned} r_{t+1}(i,d|j,1) &= \frac{1}{Z} p(z_{t+1}=j, z_{t+1}=1 | z_t = i, z_t^D = d) p(z_t = i, z_t^D = d | x_{0:t}) \\ &= \frac{1}{Z}\left(1 - \frac{D_i(d+1)}{D_i(d)}\right) A_{ij}\phi_t(i,d), \end{aligned}$$

with Z such that $\sum_i \sum_d r_{t+1}(i,d|j,1) = 1$.

Online EM algorithm

Given a decreasing sequence $(\gamma_t)_t$ satisfying $\sum_t \gamma_t = \infty$ and $\sum_t \gamma_t^2 < \infty$, the minimum time before the first parameter update $t_{\min}$, and an initial parameter vector $\hat{\theta}_0$, the online EM algorithm for the previously described HSMM model is as follows.

Initialization

$$\hat{\phi}_0(i,1) = \frac{\pi_i p(x_0|z_0 = i; \hat{\theta}_0)}{\sum_i \pi_i p(x_0|z_0 = i; \hat{\theta}_0)}$$
$$\hat{\phi}_0(i,d) = 0, \quad \text{if } d \geq 2$$
$$\hat{\rho}_0(i,d) = 0.$$

$\hat{\rho}_0$ can be initialized differently for emission sufficient statistics.

Online EM recursion For $t = 1, 2, \ldots$

- SA E-step

$$\tilde{\phi}_{t+1}(j,1) = \sum_d \sum_i \hat{\phi}_t(i,d) \left(1 - \frac{D_i(d+1)}{D_i(d)}\right) A_{ij} p(x_{t+1}|z_{t+1}=j)$$

$$\tilde{\phi}_{t+1}(j,d) = \hat{\phi}_t(j,d-1)\frac{D_j(d)}{D_j(d-1)} p(x_{t+1}|z_{t+1}=j), \quad \text{if } d \geq 2$$

$$\hat{\phi}_{t+1}(i,d) = \frac{\tilde{\phi}(i,d)}{\sum_j \sum_{d'} \tilde{\phi}(j,d')}$$

$$\hat{\rho}_{t+1}(j,1) = \sum_i \sum_d \left(\gamma_{t+1} s(i,d,j,1,x_{t+1}) + (1-\gamma_{t+1})\hat{\rho}_t(i,d)\right) \hat{r}_{t+1}(i,d|j,1)$$

$$\hat{\rho}_{t+1}(j,d) = \gamma_{t+1} s(j,d-1,j,d,x_{t+1}) + (1-\gamma_{t+1})\hat{\rho}_t(j,d-1), \quad \text{if } d \geq 2,$$

  where A, $p(\cdot|z_{t+1}=i)$ and possibly D_i implicitly depend on the current parameter estimate $\hat{\theta}_t$, and

$$\hat{r}_{t+1}(i,d|j,1) = \frac{1}{Z}\left(1 - \frac{D_i(d+1)}{D_i(d)}\right) A_{ij}\hat{\phi}_t(i,d).$$

- M-step

$$\hat{\theta}_{t+1} = \begin{cases} \bar{\theta}\left(\sum_i \sum_d \hat{\rho}_{t+1}(i,d)\hat{\phi}_{t+1}(i,d)\right), & \text{if } t \geq t_{\min} \\ \hat{\theta}_t, & \text{otherwise.} \end{cases}$$

Parameter estimation

Transition matrix The parameter update for the transition matrix A is similar to the case of regular HMMs. The expected sum of interest is

$$S_T^A(i,j) = \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{z_{t-1}=i, z_t=j, z_t^D=1\} \;\middle|\; x_{0:T}\right],$$

and we define the corresponding forward smoothing ρ quantity

$$\rho_T^A(i,j,k,u) = \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{z_{t-1}=i, z_t=j, z_t^D=1\} \;\middle|\; x_{0:T}, z_T=k, z_T^D=u\right].$$

The online EM recursion for estimating ρ^A is as follows:

$$\hat{\rho}_{t+1}^A(i,j,k',1) = \gamma_{t+1}\delta(j,k')\hat{r}_{t+1}(i|j,1) + (1-\gamma_{t+1})\sum_k \sum_u \hat{\rho}_t^A(i,j,k,u)\hat{r}_{t+1}(k,u|k',1)$$

$$\hat{\rho}_{t+1}^A(i,j,k',u) = (1-\gamma_{t+1})\hat{\rho}_t^A(i,j,k',u-1), \quad \text{if } u \geq 2,$$

where $\hat{r}_{t+1}(\cdot,\cdot|\cdot,\cdot)$ is the approximate backward retrospective probability, and $\hat{r}_{t+1}(i|j,1) = \sum_d \hat{r}_{t+1}(i,d|j,1)$. We then have an online EM approximation of the statistics S_t^A:

$$\hat{S}_t^A(i,j) = \sum_k \sum_u \hat{\rho}_t^A(i,j,k,u)\phi_t(k,u),$$

and the parameter update on the transition matrix becomes

$$\hat{A}_{ij}^{(t+1)} = \frac{\hat{S}_t^A(i,j)}{\sum_{j'} \hat{S}_t^A(i,j')}.$$

The updates for emission parameters are the same as the HMM case (with the additional duration state variable in the online EM recursion). The time complexity of the A and μ updates is now $O(K^4 D + K^3 Dp)$, which only grows linearly with the maximum duration D thanks to the deterministic transitions in our model.

Duration distributions In order to estimate duration distributions, we can consider the quantities $\lambda_i(d) = D_i(d+1)/D_i(d)$ as our duration parameters. These parameters can then be estimated using the following expected sums of sufficient statistics:

$$S_T(i,d) = \frac{1}{T}\,\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{z_{t-1} = i, z_{t-1}^D = d, z_t^D = d+1\} \;\middle|\; x_{0:T}\right]$$
$$S'_T(i,d) = \frac{1}{T}\,\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{z_{t-1} = i, z_{t-1}^D = d, z_t^D = 1\} \;\middle|\; x_{0:T}\right],$$

or their online EM estimates. The maximum likelihood parameter is then given by

$$\hat{\lambda}_i(d) = \frac{S_T(i,d)}{S_T(i,d) + S'_T(i,d)}.$$

The survivor function $D_i(d)$ can then be obtained by setting $D_i(1) = 1$ and $D_i(d+1) = \lambda_i(d)D_i(d)$ for $d \geq 1$. One can also estimate a parametric duration distribution from the obtained tabular distribution. Note that it is hard to estimate duration distributions in an online manner at the beginning of the sequence since only few segments can be observed, thus it can help to add priors, either on the λ_i or on the parameters of the parametric duration distribution, and perform MAP estimation rather than maximum likelihood (see §3.4).

3.2 Non-probabilistic models

If we consider a mixture model with emission distributions associated to the Bregman divergence D_ψ, and with $\pi_k = 1/K$ fixed, then we have:

$$\begin{aligned} -\log p(\mathbf{x}, \mathbf{z}; \mu) &= -\sum_i \log p(z_i) - \sum_i \log p(x_i | z_i; \mu_{z_i}) \\ &= C + \sum_i D_\psi(x_i, \mu_{z_i}). \end{aligned}$$

Thus, the negative comoplete-data log-likelihood is equal to the K-means objective (2.9), up to a constant term. The K-means algorithm can thus be seen as alternatively maximizing the complete-data log-likelihood with respect to the hidden variables z and the parameters μ. Or, conversely, if we have a probabilistic latent variable model, we can consider the complete-data likelihood as an objective function, and optimize it directly with respect to the latent variables and the parameters. More generally, we are replacing the the sum-product operation used to obtain expected sufficient statistics in the E-step of EM with a max-product operation, which finds a set of variables $\mathbf{z}$ with maximum joint probability (which is equivalent to MAP estimation). The two steps in the main loop of the algorithm are therefore:

$$\begin{aligned} \mathbf{z}^{(k)} &= \arg\max_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}; \theta^{(k-1)}) \\ \theta^{(k)} &= \arg\max_{\theta} p(\mathbf{x}, \mathbf{z}^{(k)}; \theta). \end{aligned}$$

In the case of HMMs, the first step is given by the Viterbi algorithm, and the second steps computes the centroids of the points in each cluster. This has been called the segmental K-means algorithm (Juang and Rabiner, 1990), and a similar algorithm can be derived for HSMMs.

3.2.1 Online algorithm

If we consider an HMM with parameters π and A fixed and look at the form of the joint likelihood $p(x_{1:T}, z_{1:T}; \mu)$ given in Eq. (2.21), we can derive a corresponding objective function to be

minimized:

$$\ell(z_{1:T}, \mu) = \frac{1}{T} \sum_{t \geq 1} D_\psi(x_t, \mu_{z_t}) + \frac{\lambda_1}{T} \sum_{t \geq 2} d(z_{t-1}, z_t), \tag{3.9}$$

where $d(z_{t-1}, z_t)$ is the cost of a transition from z_{t-1} to z_t and λ_1 a parameter that we fix. An offline algorithm could minimize this objective with respect to the sequence $z_{1:T}$ using a Viterbi algorithm, then with respect to μ by computing centroids. This can be seen as a majorization-minimization algorithm (Mairal, 2014) on the objective

$$f_T(\mu) := \min_{z_{1:T}} \ell(z_{1:T}, \mu), \tag{3.10}$$

where the majorizing surrogates (upper bounds on f_T) take the form $\mu \mapsto \ell(z_{1:T}, \mu)$ for a fixed sequence $z_{1:T}$, and are tight at the current parameter μ after the "E-step" (Viterbi step). In an online setting, we can follow (Mairal et al., 2010) and at each time step t, minimize a new surrogate

$$\hat{f}_t(\mu) := \frac{1}{t} \sum_{i=1}^{t} D_\psi(x_i, \mu_{z_i}) + \frac{\lambda_1}{t} \sum_{i=2}^{t} d(z_{i-1}, z_i), \tag{3.11}$$

where $z_{1:t-1}$ have been previously computed and z_t is chosen greedily to minimize this surrogate with the current parameter μ. μ is then updated by minimizing $\hat{f}_t$, which corresponds to computing new centroids and can be done as an online update rule similar to the online K-means algorithm described in Bottou and Bengio (1995); Bottou (1998). The function $\hat{f}_t$ is still a majorizing surrogate, but given that the previous sequence elements are not updated, the bound on f_t isn't tight, as in (Mairal et al., 2010). The algorithm is given in Algorithm 3.1.

Algorithm 3.1: Online optimization algorithm for the non-probabilistic HMM.

Data: Input stream $x_t, t = 1, \ldots$, initial centroids $\mu_1, \ldots, \mu_K$, Bregman divergence D_ψ.
Result: final centroids $\mu_1, \ldots, \mu_K$
$n_1 = \ldots = n_K = 0$; `// cluster counts`
$z_1 \leftarrow \arg\max_j D_\psi(x_1, \mu_j)$;
$\mu_{z_1} \leftarrow x_1$;
for $t = 2, \ldots$ **do**
  $j \leftarrow \arg\min_k D_\psi(x_t, \mu_k) + \lambda_1 d(z_{t-1}, k)$
  $z_t \leftarrow j$;
  $n_j \leftarrow n_j + 1$;
  $\mu_j \leftarrow \mu_j + \frac{1}{n_j}(x_t - \mu_j)$
end

Adding new states The algorithm can easily be modified to add new states when needed, by adding a penalty term on the number of clusters K to the objective, as we did with K-means:

$$\ell(z_{1:T}, \mu, K) = \frac{1}{T} \sum_{t \geq 1} D_\psi(x_t, \mu_{z_t}) + \frac{\lambda_1}{T} \sum_{t \geq 2} d(z_{t-1}, z_t) + \frac{\lambda_2}{T} K. \tag{3.12}$$

This gives us Algorithm 3.2, where we consider the transition cost to be $d(j, k) = 0$ if $j = k$ and 1 otherwise.

Semi-Markov extension Note that the algorithm can be extended to a semi-Markov model, by considering the parameterization described in §3.1.2, and taking a transition cost of the form

$$d(z_{t-1}, z_{t-1}^D, z_t, z_t^D) = -\log p(z_t, z_t^D | z_{t-1}, z_{t-1}^D).$$

Algorithm 3.2: Online optimization algorithm for the non-probabilistic HMM where new states can be added.

Data: Input stream $x_t, t = 1, \ldots$, Bregman divergence D_ψ.
Result: final centroids $\mu_1, \ldots, \mu_K$
$K \leftarrow 1$;
$z_1 \leftarrow 1$;
$\mu_1 \leftarrow x_1$;
$n_1 \leftarrow 1$;
for $t = 2, \ldots$ **do**

$$j \leftarrow \arg\min_{k \in \{1,\ldots,K+1\}} \begin{cases} D_\psi(x_t, \mu_k), & \text{if } k = z_{t-1} \\ \lambda_1 + \lambda_2, & \text{if } k = K+1 \\ D_\psi(x_t, \mu_k) + \lambda_1, & \text{otherwise.} \end{cases}$$

  if $j = K + 1$ **then**
    $K \leftarrow K + 1$;
    $z_t \leftarrow K$;
    $\mu_K \leftarrow x_t$;
    $n_K \leftarrow 1$;
  else
    $z_t \leftarrow j$;
    $n_j \leftarrow n_j + 1$;
    $\mu_j \leftarrow \mu_j + \frac{1}{n_j}(x_t - \mu_j)$
  end
end

Because of the deterministic transitions in the model, most transitions have infinite cost, and the only possible choices are staying in the same state with $z_{t-1}^D = d$ and $z_t^D = d + 1$, or changing state and having $z_t^D = 1$.

3.3 Incremental EM

An alternative algorithm to perform online updates in EM, in the case of independent observations, is the incremental approach of Neal and Hinton (1998). Rather than relying on a stochastic approximation procedure as in the online EM of Section 3.1, the algorithm performs an incremental minorization-maximization of the log-likelihood just like batch EM, but where the E-step is only taken on a single observation. As we saw in Section 2.3.4, the E-step maximizes a lower bound on the log-likelihood with respect to a distribution q obtained with Jensen's inequality:

$$f(\theta) = p(\mathbf{x}; \theta) \geq \sum_{\mathbf{z}} q(\mathbf{z}) \log \frac{p(\mathbf{x}, \mathbf{z}; \theta)}{q(\mathbf{z})}.$$

We saw that this lower bound is maximized (and made tight) for $q(\mathbf{z}) = p(\mathbf{z}|\mathbf{x}; \theta) = \prod_i p(z_i|x_i; \theta)$. We can thus limit ourselves to distributions of the form $q(\mathbf{z}) = \prod_i q_i(z_i)$ where $\sum_k q_i(k) = 1$ for

all i since the maximizer takes this form. The lower bound then takes the form

$$\begin{aligned}\hat{f}_q(\theta) &= \sum_{z_1,\dots,z_n} (\prod_i q_i(z_i)) \log \frac{\prod_i p(x_i, z_i; \theta)}{\prod_i q_i(z_i)} \\ &= \sum_{i=1}^{n} \sum_{z_1,\dots,z_n} (\prod_i q_i(z_i)) \log \frac{p(x_i, z_i; \theta)}{q_i(z_i)} \\ &= \sum_{i=1}^{n} \sum_{z_i} q_i(z_i) \log \frac{p(x_i, z_i; \theta)}{q_i(z_i)}.\end{aligned}$$

The incremental EM algorithm proceeds by selecting a data point i, maximizing the lower bound with respect to q_i, i.e. taking $q_i(z_i) = p(z_i|x_i;\theta)$, then maximizing the updated minorizing surrogate $\hat{f}_q$. In the online streaming case (where new observations come in), one can consider surrogates of the form

$$\hat{f}_n(\theta) = \frac{1}{n} \sum_{i=1}^{n} \sum_{z_i} q_i(z_i) \log \frac{p(x_i, z_i; \theta)}{q_i(z_i)}, \tag{3.13}$$

where $q_1, \dots, q_{n-1}$ are fixed from the past. $q_n(z_n) = p(z_n|x_n;\theta)$ is computed in the E-step using the current parameter estimate θ, and the M-step then maximizes this new surrogate $\hat{f}_n$, which is equal to $(1/n)\,\mathbb{E}_q[\log p(\mathbf{x}, \mathbf{z}; \theta)]$ up to a constant entropy term on q independent of θ. When the complete-data model lies in the exponential family as in Eq. (3.1), we have

$$\begin{aligned}\hat{f}_n(\theta) &= C + \frac{1}{n} \sum_{i=1}^{n} \sum_{z_i} q_i(z_i)(\langle s(x_i, z_i), \eta(\theta)\rangle - a(\theta)) - \frac{1}{n} \sum_{i=1}^{n} \sum_{z_i} q_i(z_i) \log q_i(z_i) \\ &= C + \langle \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{z_i \sim q_i}[s(x_i, z_i)], \eta(\theta)\rangle - a(\theta) + \frac{1}{n} H(q),\end{aligned}$$

where $H(q)$ is the entropy of q and C a constant which only depends on observations. This leads to an online algorithm which is similar to the online EM of Cappé and Moulines (2009) described in Section 3.1, where sufficient statistics are updated with step sizes of the form $\gamma_n = 1/n$ and the parameter updates are obtained in closed form from sufficient statistics.

3.3.1 Incremental EM for HMMs

In the case of HMMs, the factorization $q(z_{1:T}) = \prod_t q_t(z_t)$ doesn't hold anymore since observations aren't i.i.d., but using the chain rule and conditional independences in HMMs, we can factor the posterior distribution as follows:

$$p(z_{1:T}|x_{1:T};\theta) = p(z_1|x_{1:T}) \prod_{t \geq 1} p(z_t|z_{t-1}, x_{1:T}).$$

Hence, we can limit ourselves to distributions q of the form $q(z_{1:T}) = q_1(z_1) \prod_{t \geq 2} q_t(z_t|z_{t-1})$ (with $\sum_j q_t(j|i) = 1$ for all $i, t \geq 2$ and q_1 fixed to π) in the lower bound on the likelihood, since

the maximizer takes this form. We then have (with the abuse of notation $q_1(z_1|z_0) = q_1(z_1)$)

$$\begin{aligned} f_T(\theta) = \frac{1}{T}\log p(x_{1:T};\theta) &\geq \frac{1}{T}\sum_{z_{1:T}} q(z_{1:T}) \log \frac{p(x_{1:T}, z_{1:T};\theta)}{q(z_{1:T})} \\ &= \frac{1}{T}\sum_{z_{1:T}} q(z_{1:T}) \log \frac{\prod_t p(x_t, z_t|z_{t-1};\theta)}{\prod_t q_t(z_t|z_{t-1})} \\ &= \frac{1}{T}\sum_{t=1}^{T}\left[\sum_{z_{1:T}} q(z_{1:T}) \log \frac{p(x_t, z_t|z_{t-1};\theta)}{q_t(z_t|z_{t-1})}\right] \\ &= \frac{1}{T}\sum_{t=1}^{T}\left[\sum_{z_{t-1},z_t} \phi_{t-1}(z_{t-1}) q_t(z_t|z_{t-1}) \log \frac{p(x_t, z_t|z_{t-1};\theta)}{q_t(z_t|z_{t-1})}\right] =: \hat{f}_T(\theta), \end{aligned}$$

where we defined $\phi_t(z_t) := \sum_{z_{1:t-1}} q_1(z_1)q_2(z_2|z_1)\dots q_t(z_t|z_{t-1}) = \sum_{z_{t-1}} \phi_{t-1}(z_{t-1})q(z_t|z_{t-1})$ for $t \geq 2$ and $\phi_1(z_1) = q_1(z_1)$. The quantity $\phi_t(z_t)$ corresponds to the marginal of q on z_t, while $\phi_{t-1}(z_{t-1})q_t(z_t|z_{t-1})$ is the pairwise marginal on z_{t-1} and z_t, which appears naturally in our lower bound. Thus, in the online streaming setting, we can consider at time T the surrogate $\hat{f}_T$, where q_t and ϕ_t are fixed from the past for $t < T$, and q_T is obtained in the E-step to maximize the surrogate at the current parameter estimate, giving $q_T(z_T|z_{T-1}) = p(z_T|z_{T-1}, x_T;\theta)$. The M-step then maximizes this new surrogate. If we assume $p(x_t, z_t|z_{t-1};\theta)$ is in the exponential family as in Eq. 3.2, the surrogate takes the form

$$\begin{aligned} \hat{f}_T(\theta) &= C + \frac{1}{T}\sum_{t=1}^{T}\left[\sum_{z_{t-1},z_t} \phi_{t-1}(z_{t-1})q_t(z_t|z_{t-1})(\langle s(z_{t-1}, z_t, x_t), \eta(\theta)\rangle - a(\theta))\right] + \frac{1}{T}H(q) \\ &= C + \langle S_T, \eta(\theta)\rangle - a(\theta) + \frac{1}{T}H(q), \end{aligned}$$

where $S_T = \frac{1}{T}\sum_{t=1}^{T}\sum_{z_{t-1},z_t} \phi_{t-1}(z_{t-1})q_t(z_t|z_{t-1})s(z_{t-1}, z_t, x_t)$ and C is independent of θ. If we define $\bar{\theta}(s) := \arg\max_\theta \langle s, \eta(\theta)\rangle - a(\theta)$ the maximizer of the complete-data likelihood (assumed unique as in §3.1.1), the surrogate $\hat{f}_T$ is maximized by taking $\theta = \bar{\theta}(S_T)$. We obtain Algorithm 3.3. In order to get our definition of S_T, the step sizes are taken to be $\gamma_t = 1/t$, although in practice we often obtained improvements for slower step sizes such as $\gamma_t = t^{-0.6}$.

Algorithm 3.3: Incremental EM algorithm for HMMs.

Data: Input stream $x_t, t = 1, \dots$, initial parameter θ, fixed π, first M-step time t_{min}.
Result: final parameter θ
$\phi_1(i) \leftarrow p(z_1 = i|x_1;\theta) = \frac{1}{Z}\pi_i p(x_1|z_1 = i;\theta)$;
$S \leftarrow \sum_i \phi_1(i)s(i, x_1)$;
for $t = 2, \dots$ **do**
  `// E-step`
  $q_t(j|i) \leftarrow \frac{p(x_t, z_t = j|z_{t-1} = i;\theta)}{\sum_{j'} p(x_t, z_t = j'|z_{t-1} = i;\theta)}$;
  $\phi_t(j) \leftarrow \sum_i \phi_{t-1}(i)q_t(j|i)$;
  $S \leftarrow (1 - \gamma_t)S + \gamma_t \sum_i \sum_j \phi_{t-1}(i)q_t(j|i)s(i, j, x_t)$;
  `// M-step`
  if $t \geq t_{min}$ **then**
    $\theta \leftarrow \bar{\theta}(S)$;
  end
end

Example 3.2. If we now consider the HMM with transition matrix A and emissions associated to Bregman divergence D_ψ with centroids μ_k, we have $p(x_t, z_t = j|z_{t-1} = i; \theta) = A_{ij}p(x_t|z_t = j; \mu_j)$. The sufficient statistics $s(z_{t-1}, z_t, x_t)$ are given as in §3.1.1 by $s_{ij}(z_{t-1}, z_t, x_t) = \mathbb{1}\{z_{t-1} = i, z_t = j\}$, $s_i^0(z_{t-1}, z_t, x_t) = \mathbb{1}\{z_t = i\}$, $s_i^1(z_{t-1}, z_t, x_t) = \mathbb{1}\{z_t = i\}x_t$. It follows that the updates of sufficient statistics in the E-step are simply

$$\begin{aligned} S_t^A(i,j) &= (1-\gamma_t)S_{t-1}^A(i,j) + \gamma_t\phi_{t-1}(i)q_t(j|i) \\ S_t^{\mu,0}(i) &= (1-\gamma_t)S_{t-1}^{\mu,0}(i) + \gamma_t\phi_t(i) \\ S_t^{\mu,1}(i) &= (1-\gamma_t)S_{t-1}^{\mu,2}(i) + \gamma_t\phi_t(i)x_t. \end{aligned}$$

Note that these updates do not require summing over the previous state as in the ρ updates of the forward smoothing recursion in §3.1.1, and are thus less costly, with a time complexity of $O(K^2 + Kp)$. The M-step is similar to the online EM algorithm in §3.1.1:

$$\begin{aligned} A_{ij}^{(t)} &= \frac{S_t^A(i,j)}{\sum_{j'} S_t^A(i,j')} \\ \mu_i^{(t)} &= \frac{S_t^{\mu,1}(i)}{S_t^{\mu,0}(i)}. \end{aligned}$$

3.3.2 Semi-Markov extension

Our incremental EM algorithm can be extended to HSMMs using the two-variable hidden chain described in Section 3.1.2. The quantities of interest are now $q_t(z_t, z_t^D|z_{t-1}, z_{t-1}^D)$ and $\phi_t(z_t, z_t^D)$. Thanks to the deterministic transitions, the quantities $q_t(j, d'|i, d)$ will be zero both if $d' \notin \{1, d+1\}$ or if $i \neq j$ and $d = d+1$, thus the E-step reduces to the following:

$$\begin{aligned} \tilde{q}_t(j,1|i,d) &= (1-\lambda_i(d))A_{ij}p(x_t|z_t=j) \\ \tilde{q}_t(i,d+1|i,d) &= \lambda_i(d)p(x_t|z_t=i) \\ q_t(j,d'|i,d) &= \begin{cases} \frac{1}{Z}\tilde{q}_t(j,d'|i,d), & \text{if } d'=1 \text{ or } d'=d+1 \text{ and } i=j \\ 0, & \text{otherwise.} \end{cases} \\ \phi_t(j,1) &= \sum_{i,d} \phi_{t-1}(i,d)q_t(j,1|i,d) \\ \phi_t(j,d) &= \phi_{t-1}(j,d-1)q_t(j,d|j,d-1), \quad \text{if } d \geq 2, \end{aligned}$$

where $\lambda_i(d) = D_i(d+1)/D_i(d)$ and the normalizing constant Z is given by $Z = \tilde{q}_t(i,d+1|i,d) + \sum_j \tilde{q}_t(j,1|i,d)$. The complexity of these updates is thus $O(K^2D)$.

Example 3.3. If we consider the HSMM version of the model in Example 3.2, the relevant sufficient statistics for estimating the transition matrix and the emission parameters are $\mathbb{1}\{z_{t-1} = i, z_t = j, z_t^D = 1\}$, $\mathbb{1}\{z_t = i\}$ and $\mathbb{1}\{z_t = i\}x_t$. The updates are then the following:

$$\begin{aligned} S_t^A(i,j) &= (1-\gamma_t)S_{t-1}^A(i,j) + \gamma_t \sum_d \phi_{t-1}(i,d)q_t(j,1|i,d) \\ S_t^{\mu,0}(i) &= (1-\gamma_t)S_{t-1}^{\mu,0}(i) + \gamma_t \sum_d \phi_t(i,d) \\ S_t^{\mu,1}(i) &= (1-\gamma_t)S_{t-1}^{\mu,2}(i) + \gamma_t \left(\sum_d \phi_t(i,d)\right) x_t. \end{aligned}$$

The M-step update are similar to the HMM (see Example 3.2). This gives us an overall time complexity per observation of $O(K^2D + KDp)$, where D is the maximal duration of a segment. Note that one can also estimate the duration parameters λ_i using the statistics $\mathbb{1}\{z_{t-1} = i, z_{t-1}^D = d, z_t^D = d+1\}$ and $\mathbb{1}\{z_{t-1} = i, z_{t-1}^D = d, z_t^D = 1\}$, as in §3.1.2.

3.4 Including prior knowledge with Bayesian priors

The online algorithms we presented so far can be noisy in the beginning and take some time before giving good results. In addition, one often needs to wait a given number t_{min} of iterations before carrying M-step updates, so that the sufficient statistics are somewhat stabilized and the parameter update is numerically stable.

Often, we might have some reliable prior information about what the different clusters should look like. For example, if we are segmenting musical structures, such as notes and chords, we might know in advance the harmonic structure of these elements. In this case, it is desirable to encode this information into the model in order to improve the accuracy. A simple way to take this into account is to have good initial parameters which represent each cluster, but this initialization will be ignored after the first parameter update. A more suitable and natural way to encode prior information is to add custom Bayesian priors on the model parameters, and replace maximum likelihood updates of the parameters with *maximum a posteriori* (MAP) updates.

We will limit ourselves to conjugate priors, which will lead to a very similar form of the updates, by simply having an additional number of "virtual" observations that trade off against actual observations, and are forgotten as the number of observations increases. If we consider an exponential family form for $p(x_t, z_t|z_{t-1}, \theta)$:

$$p(x_t, z_t|z_{t-1}, \theta) = h(z_t, x_t) \exp(\langle s(z_{t-1}, z_t, x_t), \eta(\theta)\rangle - a(\theta)),$$

the corresponding conjugate prior on θ takes the form

$$p(\theta; \kappa, \tau) = \exp(\langle \tau\kappa, \eta(\theta)\rangle - \tau a(\theta) - b(\tau, \kappa)), \tag{3.14}$$

where b is the log-partition function. The product $\tau\kappa$ is often considered itself as a parameter, but our parameterization leads to a natural interpretation, where τ corresponds to an equivalent sample size, and κ a vector of sufficient statistics that represents what we consider to be a good observed sufficient statistic $s(z_{t-1}, z_t, x_t)$. To see why this is true, let's consider a completely observed HMM with state sequence $z_{1:T}$ and observations $x_{1:T}$. Maximizing the posterior $p(\theta|x_{1:T}, z_{1:T})$ is equivalent to maximizing the product of the likelihood and the prior since by Bayes rule we have $p(\theta|x_{1:T}, z_{1:T}) \propto p(x_{1:T}, z_{1:T}|\theta)p(\theta)$. Taking the logarithm of this quantity gives us

$$\begin{aligned}\log p(\theta; \kappa, \tau)p(x_{1:T}, z_{1:T}|\theta) &= C + \langle \tau\kappa, \eta(\theta)\rangle - \tau a(\theta) + \langle \sum_{t=1}^{T} s(z_{t-1}, z_t, x_t), \eta(\theta)\rangle - Ta(\theta) \\ &= C + \langle \tau\kappa + TS_T, \eta(\theta)\rangle - (\tau + T)a(\theta),\end{aligned}$$

where $S_T = \frac{1}{T}\sum_{t=1}^{T} s(z_{t-1}, z_t, x_t)$ and C is a constant which doesn't depend on θ. If $\bar{\theta}(s) := \arg\max_\theta \langle s, \eta(\theta)\rangle - a(\theta)$, the maximum a posteriori estimate of θ is then given by

$$\hat{\theta}^{MAP} = \bar{\theta}\left(\frac{\tau\kappa + TS_T}{\tau + T}\right). \tag{3.15}$$

We can see the desired trade-off between the statistics of the prior κ and of the observations S_T, which leans towards the prior when T is small and towards observed data when T becomes large. When the sequence $z_{1:T}$ is unobserved, the same parameter update is obtained in the M-step of a (batch or online) EM algorithm, by considering S_T to be the expected average sum of sufficient statistics, as we defined for the online EM algorithm of Section 3.1.1 and the incremental EM algorithm of Section 3.3.1.

Example 3.4. In our HMM with Bregman emissions, using the notations of Example 3.2, we can define priors on A using an equivalent number of transitions κ_{ij}^A that trade-off against the observed transitions, and for μ, an equivalent number of observations $\kappa_i^{\mu,0}$ and a vector $\kappa_i^{\mu,1} \in \mathbb{R}^p$,

which might correspond to an ideal short-time Fourier spectrum vector for a given note or chord in the context of musical note segmentation. For simplicity, we can take $\tau = 1$ in this case, since the vector κ already encodes equivalent sample sizes. The M-step updates are then

$$A_{ij}^{(t)} = \frac{\kappa_{ij}^A + TS_t^A(i,j)}{\sum_{j'}(\kappa_{ij'}^A + TS_t^A(i,j'))}$$
$$\mu_i^{(t)} = \frac{\kappa_i^{\mu,0}\kappa_i^{\mu,1} + TS_t^{\mu,1}(i)}{\kappa_i^{\mu,0} + TS_t^{\mu,0}(i)}.$$

Note that the transitions are given by categorical distributions (multinomial with a single trial), hence the corresponding conjugate priors are Dirichlet distributions. Similarly, if the emissions are multinomial, that is, D_ψ is the KL divergence, the conjugate prior is also Dirichlet.

In the case of HSMMs, similar priors can be defined, and additional priors can be defined on duration distributions, the conjugate priors being Gamma distributions for Poisson durations, and Beta distributions for the parameter p of Negative Binomial durations.

3.5 Experiments on synthetic data

In this section, we compare the behavior of online EM (§3.1.1) and incremental EM (§3.3.1) for HMMs on synthetic sequences generated from some fixed parameters, with either Gaussian or KL (multinomial) emissions. See (Cappé, 2011) for a description of the sufficient statistics updates used for Gaussians. We used a training sequence of length 4000 and a test sequence of length 1000 to monitor how the test likelihood evolves. The step sizes were fixed to $\gamma_t = t^{-0.6}$, and the time of the first M-step update to $t_{min} = 80$.

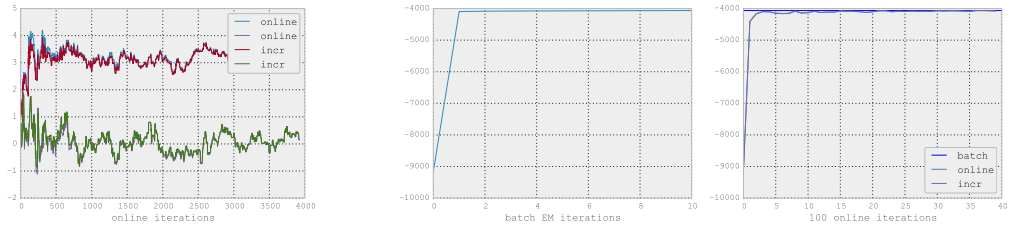
Gaussian toy example We first compared the two algorithms on two-dimensional Gaussians with $K = 2$:

$$\mu_1 = \begin{pmatrix} 3 \\ 0 \end{pmatrix} \quad \Sigma_1 = \begin{pmatrix} 2 & 1 \\ 1 & 4 \end{pmatrix}$$
$$\mu_2 = \begin{pmatrix} -2 \\ 3 \end{pmatrix} \quad \Sigma_2 = \begin{pmatrix} 3 & -1 \\ -1 & 2 \end{pmatrix}.$$

Initial parameters were set to $\mu_1^0 = (1,0)^\top, \mu_2^0 = (-1,0)^\top$ and covariances equal to identity. Figure 3.2 shows the evolution of parameter estimates and log-likelihood during the algorithm. We can see that both algorithms give very similar parameter estimates in this low-dimensional case, and that the test log-likelihood comes close to the batch value after a just a few hundred observations.

We noticed that the online approaches are not effective when considering Gaussians with full covariance matrices in higher dimensions (see Figure 3.3). This could be due to the fact that many observations are needed to correctly estimate covariance matrices, and it might be a bad idea to update them in an online manner without a strong prior.

Squared euclidian distance and KL divergence We then compared batch and online algorithms for Bregman divergence-based emissions, in particular the squared Euclidian distance (corresponding to Gaussians with covariance matrix equal to identity) and the KL divergence (multinomial, for which we fixed the number of trials to $N = 100$), for different values of the number of clusters K and the dimensionality of the data p. Our Gaussian centroids were sampled according to $\mathcal{N}(0, I)$, and the means of our multinomials were sampled from a $Dir(1, \ldots, 1)$. Initial parameters were sampled from $\mathcal{N}(0, I)$ for Gaussians, and for the multinomial, they were taken to be slightly perturbed uniform distributions.



(a) (b)

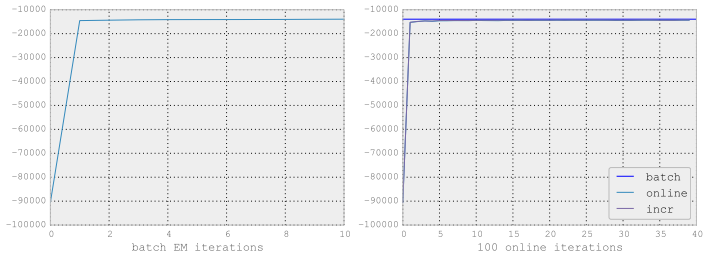
Figure 3.2: Gaussian HMM toy example, online EM vs incremental EM. (a) Evolution of parameters $\mu_{1,1}$ and $\mu_{1,2}$ for both algorithms. (b) Test log-likelihood across EM iterations for the batch EM algorithm and online observations for the online algorithms (the x axis denotes every 100th iteration after the first M-step update).

Figures 3.4, 3.5, 3.6 show results of the different algorithms for different values of K and p. The plots on the top show the evolution of the log-likelihood, both for the batch algorithm across each EM iteration, and for the online algorithms over time (plotted against the final value of batch EM). The bottom plots show the training sequence of observations, projected onto their first two principal components, where the color of each point gives the true or predicted state (for the batch algorithm, we show smoothing estimates, i.e. maximum posterior marginals, while for online algorithms we show the predicted state given by filtering during the algorithm).

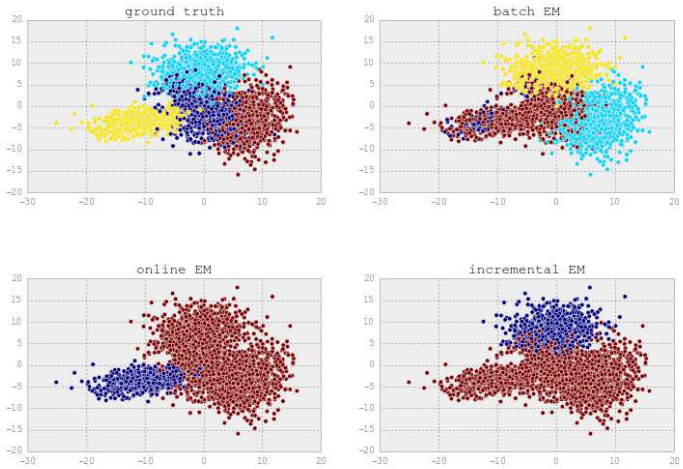
We can see that the online algorithms give good predictions, and give a good value for the test log-likelihood quite quickly, especially when the dimensionality is relatively low. The convergence is usually faster in the KL divergence case, and the incremental EM approach generally performs better than the online EM algorithm of (Cappé, 2011) when looking at the test likelihood, despite its lower computational complexity. For instance, in Figure 3.5b, we can see that after a few hundred observations incremental EM performs just as well (in terms of test likelihood) as batch EM after convergence, which required 5 iterations on the entire training sequence to converge. Of course, the non-convexity of the problem makes it highly sensitive to the random initializations, but these remarks remain consistent with the multiple simulations we ran.

3.6 Summary and discussion

In this chapter, we covered various algorithms for online learning in HMMs and HSMMs, mainly based on the EM algorithm. We explored algorithms based on stochastic approximations, following (Cappé and Moulines, 2009; Cappé, 2011), as well as incremental majorization-minimization (or minorization-maximization) algorithms, with both probabilistic and non-probabilistic versions. We discussed how to add new states on the fly in some models, and how to encode prior information about the data in the model through the use of Bayesian conjugate priors. Some questions that can be explored in future work include the use of mini-batches in online algorithms for HMMs (see the use of blocks of increasing size in the Block Online EM algorithm in (Le Corff and Fort, 2013), or the frequent use of mini-batches in i.i.d. data, e.g., (Liang and Klein, 2009; Mairal et al., 2010)), and an analysis of convergence of our incremental algorithms.



(a)



(b)

Figure 3.3: Results for Gaussians with full covariance matrices, $K = 4$, $p = 5$. (a) Test log-likelihood. (b) Projection of the observation sequence on the first two principal components. Color indicates predicted state.

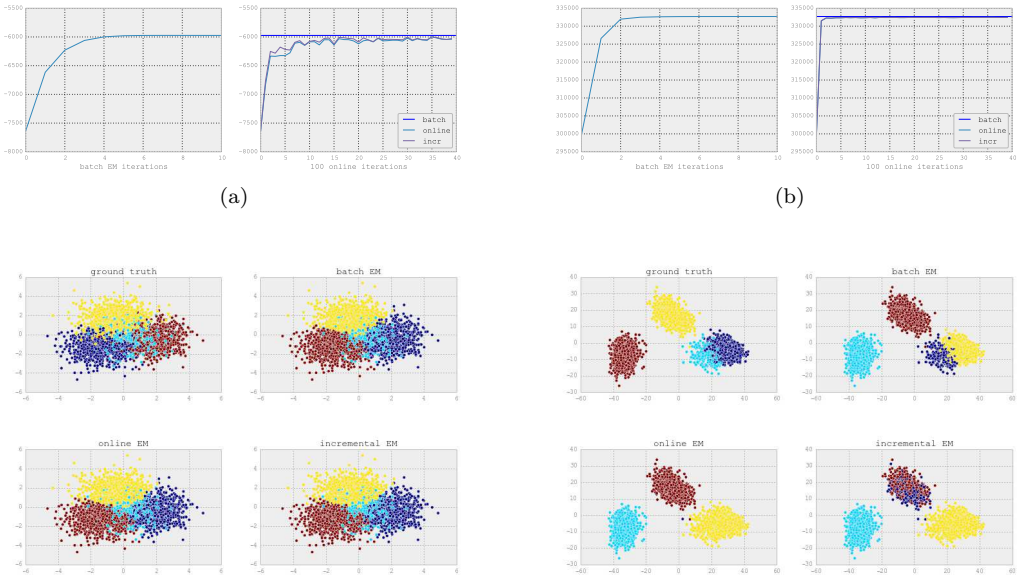


Figure 3.4: Results for squared Euclidian distance (a,c) and KL divergence (b,d), with $K = 4$, $p = 5$.

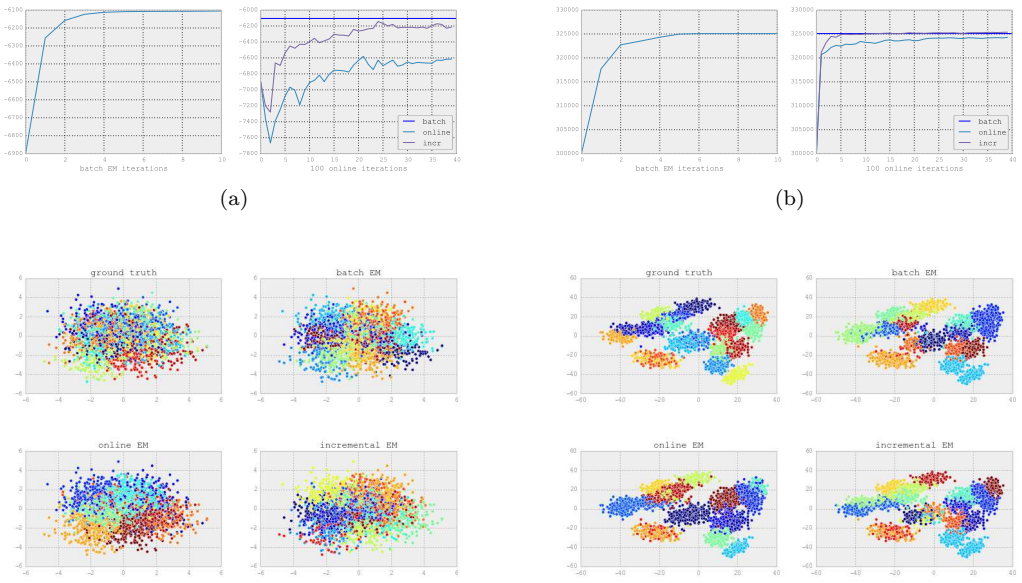


Figure 3.5: Results for squared Euclidian distance (a,c) and KL divergence (b,d), with $K = 20$, $p = 5$.

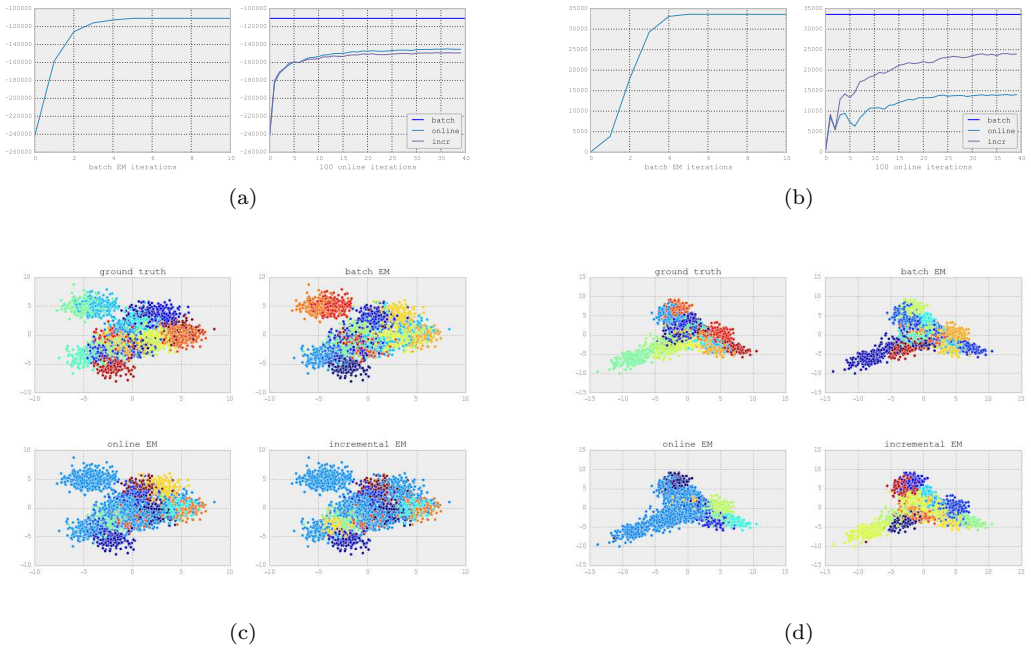


Figure 3.6: Results for squared Euclidian distance (a,c) and KL divergence (b,d), with $K = 20$, $p = 100$.

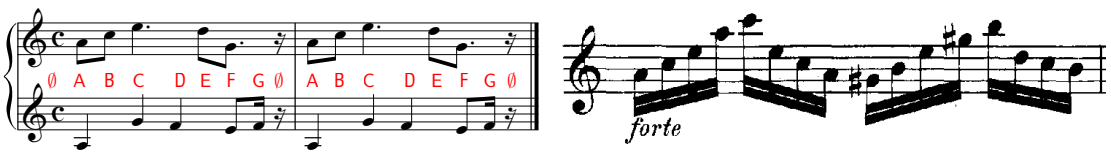
Chapter 4

Audio segmentation experiments

In this chapter, we describe and discuss some of our experiments on the audio segmentation task using the algorithms presented in the previous chapters, both in the context of musical note/structure segmentation, and acoustic scene segmentation. For musical structure segmentation, we will use two main examples, one by M. Ravel for piano and one by J.S. Bach for violin. Their music score is shown in Figure 4.1. We use the short-time spectral representation described in Section 2.1, computed using Hamming windows of size 4096 with an offset of size 512 between each window (the sample rate of our audio files is 44.1KHz). Our emission distributions are multinomial distributions, that is, the regular exponential family associated to the KL divergence. The normalization constant of our short-time vectors (which corresponds to a number of trials N in the multinomial) is chosen depending on the example: we used $N = 5$ for the Ravel example and $N = 20$ for the Bach sonata. For the acoustic scenes, we used sounds from the Office Live Dataset (Giannoulis et al., 2013), with a similar representation, and we fixed $N = 150$.

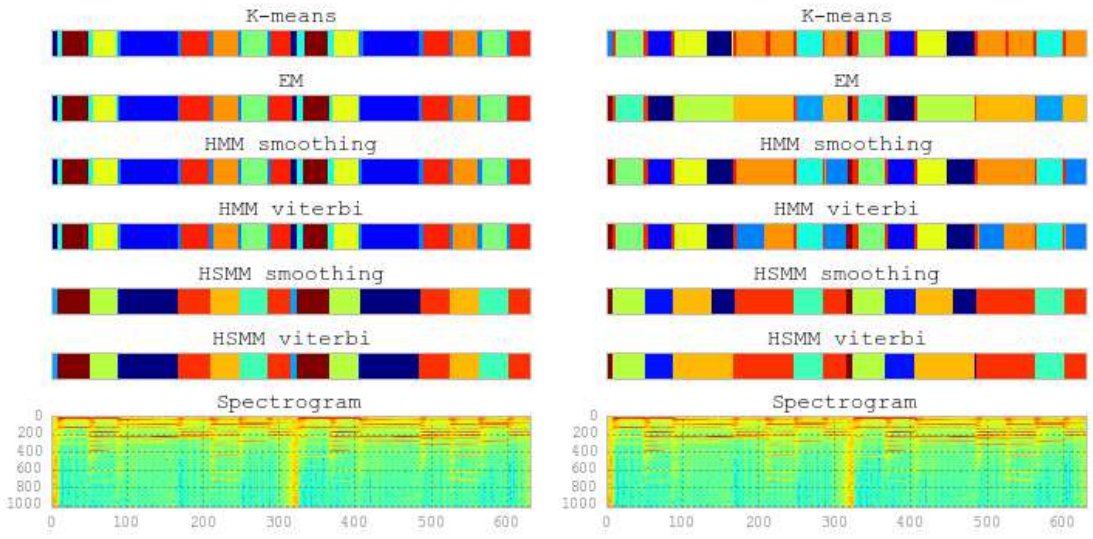
4.1 Offline segmentation results

When running the segmentation algorithms offline, we have the entire audio signal at hand, and it is common in this case to initialize the emission parameters of the HMM and HSMM with centroids obtained using the K-means algorithm (itself run multiple times with different random initializations). Figure 4.2 shows results on the Ravel excerpt, with two different K-means initializations. In the case of HSMMs, we fixed the duration distributions to a Negative Binomial distribution with parameters $r = 5, p = 0.95$, up to a maximal duration $D = 200$. We can see that the initialization has a strong effect on the resulting sequences: the results on the left are very accurate for the HSMM, while the ones on the right show some segments which include multiple different notes. We can also see that the explicit duration distributions given



(a) (b)

Figure 4.1: (a) Music score of the excerpt from M. Ravel's *Ma Mère l'Oye*. Source: (Lostanlen, 2013). (b) Score of one bar of Bach's violin sonata n. 2 (Allegro).



(a) (b)

Figure 4.2: Results on Ravel for offline algorithms with different K-means initializations. $K = 9$. HSMM duration distributions are fixed to $NegBin(5, 0.95)$.

by the HSMM help the model avoid short segments, such as attacks, which are obtained for all other models. Note that the $NB(5, 0.95)$ distribution is actually quite similar to a geometric distribution, but we are forcing a slower decay, while the HMM learns a very steep duration distribution for the attack states through the transition matrix.

Figure 4.3 shows results on the Bach example. In this case, the HMM and HSMM emissions were initialized with slightly perturbed uniform multinomial distributions ($\mu_j \propto 1 + \epsilon_j$ for $j = 1, \dots, p$, where ϵ_j is a small noise term). We can see that non-sequential methods (K-means is initialized multiple times and EM uses K-means as initialization) perform well here, since the notes are quite distinctive in this example, and the onsets are not as disruptive on the violin as they are on the piano. We can see however that the orange segment in K-means and EM is split into two by the HSMM, thanks to the duration distribution which encourages segments of length close to 20. Note that despite the random initialization, the HSMM still manages to obtain accurate segments.

4.2 Online EM for HMMs and HSMMs

Figure 4.4 compares the results of the online EM algorithms for HMMs and HSMMs described in Section 3.1 on the example by Bach. The duration distributions were chosen to be Negative Binomial with mean 20 and parameter $r = 30$. The first M-step time was $t_{min} = 80$. We show both the filtered state given in real-time during online estimation, and the full viterbi sequence obtained with the final parameters, and in Figure 4.4b we repeat the sequence of notes twice to see how the results improve. We can see that the resulting segmentation is indeed improved after seeing the sequence twice, and we can see that the HSMM manages to discover more notes (clusters) than the HMM. The execution time can be quite slow due to the expensive forward smoothing recursions, especially in the HSMM case. In the HMM case, the segmentation took about 3s on a laptop (which is close to real-time) for the single sequence, while in the HSMM case it took about 40s, with a truncation of the duration distribution to $D = 70$, which is definitely

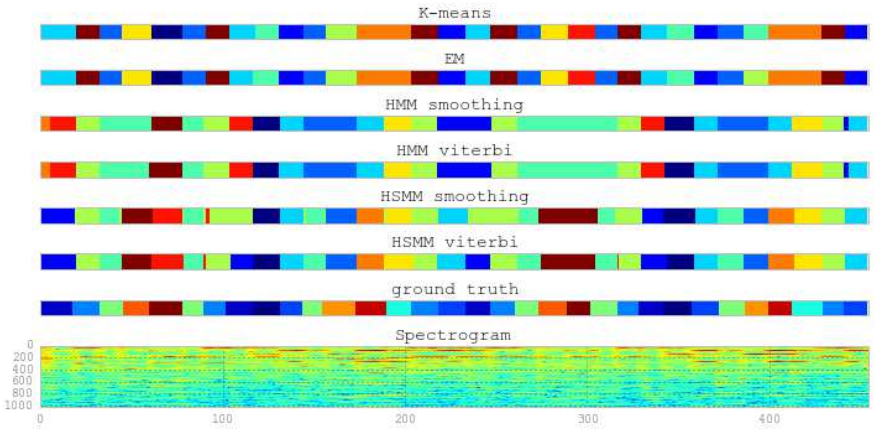


Figure 4.3: Results on the Bach example for offline algorithms. $K = 10$. HSMM durations: $NB(5, 0.2)$ (mean 20). The last segmentation shown is the ground truth.

too long for real-time applications, although it could be improved by using a compiled language rather than Python.

4.3 Online vs incremental EM for HMMs

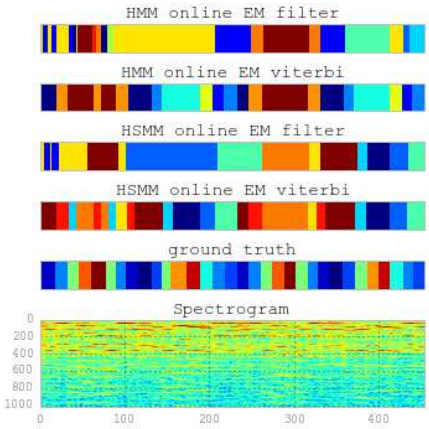
Figure 4.5 compares the online and incremental EM algorithms for HMMs, described in §3.1.1 and §3.3.1, respectively. We show results for a single sequence or the same sequence repeated twice, with $K = 10$. We can see that both methods give similar results. The online EM approach seems to model transitions better than incremental EM, which gives many short segments, especially in the shorter sequence, while incremental EM seems to discover more states compared to online EM. Another important difference is in the execution time: incremental EM is faster by almost an order of magnitude compared to online EM (0.4s vs 1.5s), and the difference gets bigger as K increases, since the complexities per observation differ by a factor $O(K^2)$.

4.4 Segmentation of acoustic scenes

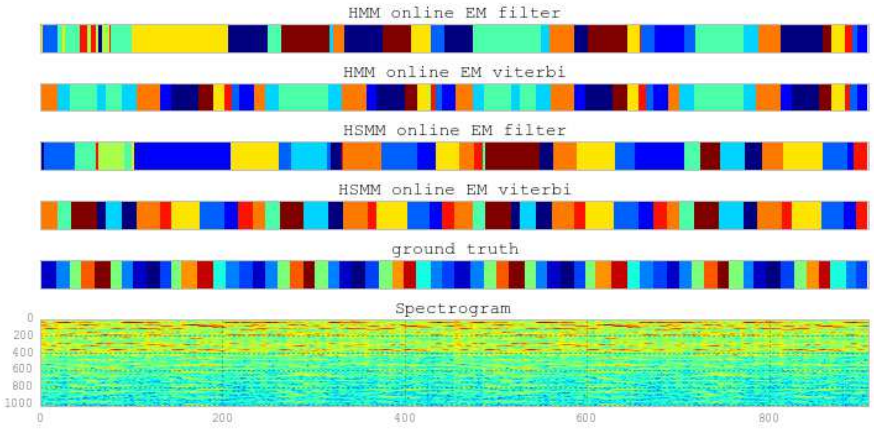
The previous examples we've shown considered the task of musical note segmentation, in which we wish to obtain a single segment for each note. In this situation, a single note or chord is mostly homogeneous and usually corresponds to a single segment (although the onset can be quite different). If we instead consider the task of detecting acoustic scenes or events – such as a door slam, a cough or the sound of keys being dropped –, the audio content isn't always homogeneous anymore, but might be described by a sequence of segments rather than a single segment. If we manage to get a similar sequence of segments for different instances of the acoustic scene, it is then easy to detect the scene with a higher-level algorithm on top of this segmentation.

In Figure 4.6, we consider two audio signals, each of which contains three different instances of two acoustic scenes, taken from the Office Live Dataset (Giannoulis et al., 2013). Emissions are initialized to slightly perturbed uniform distributions on spectra, and HSMM durations are fixed to $NB(5, 0.2)$ (mean 20). For the incremental EM, we also show the real-time estimate ("filter"), in addition to the viterbi sequence given by the final parameters.

The top example alternates between a telephone ringing sound and a coughing sound. As we can see in the spectrogram, the telephone ringing instances look quite similar, while the middle coughing is quite different from the other two. We can see that all three algorithms give consistent



(a)

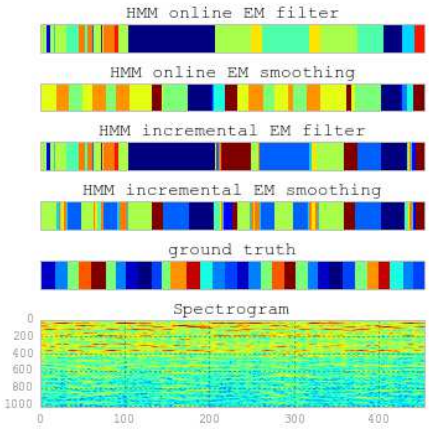


(b)

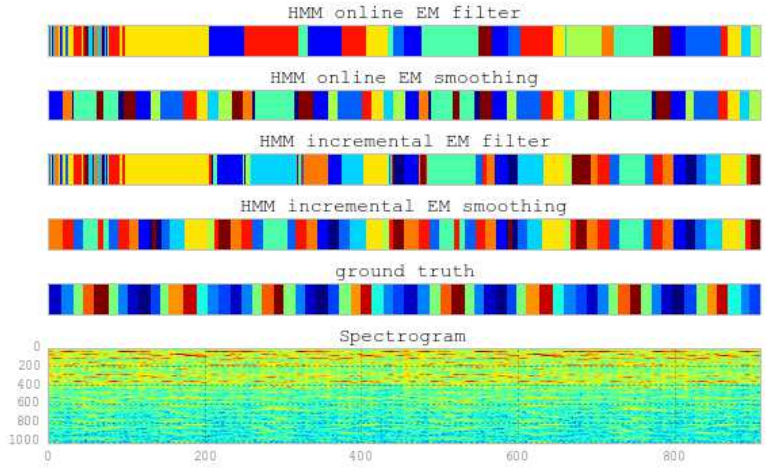
Figure 4.4: Online EM for HMM/HSMM on Bach. $K = 10$, $NB(30, 0.6)$ (mean 20). (a) Single sequence. (b) Sequence repeated twice.

segment sequences across the different instances. By encouraging longer segments, the HSMM has the added benefit of having fewer small segments for each scene, for instance the coughing sound is made of a single cluster (light blue), while the HMM breaks it into several clusters, which are different in the middle coughing. Finally, the segmentation given by the incremental EM algorithm for HMMs is quite satisfactory, given that it only observed the sequence once. Note that the background sound (silence) was detected accurately here in a single cluster, but in general one could also use a separate algorithm – based, e.g., on spectral flatness – for distinguishing it from actual content before performing the segmentation.

The bottom example alternates between sounds of keys dropping on a table and door slams. The second key drop is preceded by a sound of shaking keys that is clearly visible in the spectrogram. The background here is quite noisy and thus gives us several different clusters, but these are different from the ones obtained during the acoustic scenes, and it can be helpful in this case to rely on a separate silence detection algorithm for ignoring the background sound. If we focus on the final viterbi sequences around the acoustic events, we can see that all three algorithms give consistent segmentations across the different instances, and even manage to capture the shaking key sound (between $t = 210$ and $t = 280$) as a separate cluster. Once again, we can see



(a)



(b)

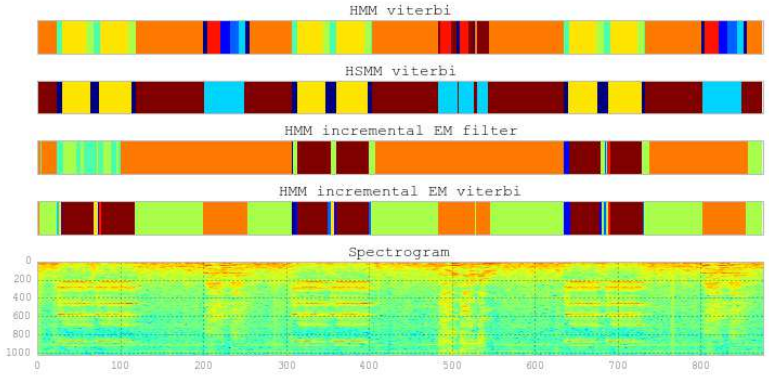
Figure 4.5: Online vs incremental EM for HMMs. The Bach sequence is repeated twice in (b)

that the HSMM uses a single segment to describe some scenes while the HMM might split it in two: for instance, the door slam sound uses a single cluster in the HSMM (dark blue), versus two clusters in the HMM (brown and red in the offline HMM).

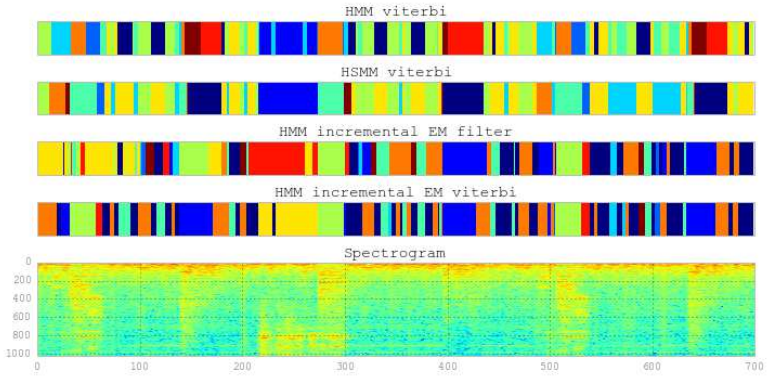
These segmentation results on acoustic scenes are quite promising, and could eventually be used in systems such as *Antescofo* for detecting other sounds than musical notes, including some percussion instruments, or other complex acoustic events that are increasingly common in contemporary classical music, and which cannot be described by hand-crafted emission templates, as is currently done for musical notes in *Antescofo*.

4.5 Summary and discussion

In this chapter, we applied the learning algorithms presented in previous chapters to the task of audio segmentation and clustering. We've seen that offline algorithms can give very accurate segmentations of musical notes, and that the explicit modeling of durations in HSMMs can help in avoiding short segments or encouraging segments of specific lengths. We then experimented with



(a)



(b)

Figure 4.6: Scenes segmentation results. In both examples, two different events are alternated and repeated 3 times. (a) phone ringing and coughing. (b) keys dropping and door slam.

online algorithms and their use for real-time audio segmentation. Although results in general are not as good as in the offline case, we saw that these algorithms can still give good segmentation results despite the initial lack of data, and how they can improve when more data is observed over time. Note that if prior information about the different clusters is at hand, encoding it into the model – as explained in Section 3.4 – can help getting better segmentation results from the start, for instance by having hand-designed prior templates for each different musical note. We also explored the use of our segmentation algorithms for acoustic scenes, and saw how different examples of a similar acoustic event produce similar sequences of segments, which can then be used at a higher level for detecting and recognizing these events. This could be a promising start towards the goal of having a real-time artificial listening machine, which might hear a few examples of a sound sequence and then recognize it later when it hears it again.

Bibliography

S.-I. Amari and H. Nagaoka. *Methods of Information Geometry*. American Mathematical Society, Providence, RI, Apr. 2007.

J. Andén and S. Mallat. Deep scattering spectrum. *IEEE Transactions on Signal Processing*, 2011.

D. Arthur and S. Vassilvitskii. K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pages 1027–1035, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.

A. Banerjee, X. Guo, and H. Wang. On the optimality of conditional expectation as a bregman predictor. *IEEE Transactions on Information Theory*, 51(7):2664–2669, July 2005a.

A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. Clustering with bregman divergences. *Journal of Machine Learning Research*, 6:1705–1749, Dec. 2005b.

O. Barndorff-Nielsen. *Information and exponential families in statistical theory*. John Wiley & Sons, 1978.

M. Basseville, I. V. Nikiforov, and others. *Detection of abrupt changes: theory and application*, volume 104. Prentice Hall Englewood Cliffs, 1993.

L. Bottou. Online learning and stochastic approximations. *On-line learning in neural networks*, 17:9, 1998.

L. Bottou and Y. Bengio. Convergence properties of the k-means algorithms. In *Advances in Neural Information Processing Systems 7*. Citeseer, 1995.

L. Bottou and O. Bousquet. The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems 20*, pages 161–168, 2008.

O. Cappé. Online sequential monte carlo EM algorithm. In *Statistical Signal Processing, 2009. SSP'09. IEEE/SP 15th Workshop on*, pages 37–40. IEEE, 2009.

O. Cappé. Online EM algorithm for hidden markov models. *Journal of Computational and Graphical Statistics*, 20(3):728–749, Jan. 2011.

O. Cappé and E. Moulines. Online expectation–maximization algorithm for latent data models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(3):593–613, June 2009.

O. Cappé, E. Moulines, and T. Ryden. *Inference in Hidden Markov Models (Springer Series in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

A. Cont. A coupled duration-focused architecture for real-time music-to-score alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):974–987, June 2010.

A. Cont, S. Dubnov, and G. Assayag. On the information geometry of audio streams with applications to similarity computing. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):837–846, May 2011.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

A. Dessein. *Computational Methods of Information Geometry with Real-Time Applications in Audio Signal Processing*. PhD thesis, Université Pierre et Marie Curie, Dec. 2012.

A. Dessein and A. Cont. An information-geometric approach to real-time audio segmentation. *Signal Processing Letters, IEEE*, 20(4):331–334, 2013.

E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky. An HDP-HMM for systems with state persistence. In *Proceedings of the 25th international conference on Machine learning*, pages 312–319. ACM, 2008.

D. Giannoulis, E. Benetos, D. Stowell, M. Rossignol, M. Lagrange, and M. D. Plumbley. Detection and classification of acoustic scenes and events: An ieee aasp challenge. In *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2013 IEEE Workshop on*, pages 1–4. IEEE, 2013.

Y. Guédon. Estimating hidden semi-markov chains from discrete sequences. *Journal of Computational and Graphical Statistics*, 12(3):604–639, 2003.

M. J. Johnson and A. S. Willsky. Bayesian nonparametric hidden semi-markov models. *The Journal of Machine Learning Research*, 14(1):673–701, 2013.

B.-H. Juang and L. Rabiner. The segmental k-means algorithm for estimating parameters of hidden markov models. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38(9): 1639–1641, Sept. 1990.

B. Kulis and M. I. Jordan. Revisiting k-means: New algorithms via bayesian nonparametrics. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 513–520, 2012.

S. Le Corff and G. Fort. Online expectation maximization based algorithms for inference in hidden markov models. *Electronic Journal of Statistics*, 7:763–792, 2013.

P. Liang and D. Klein. Online EM for unsupervised models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 611–619, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

V. Lostanlen. Découverte automatique de structures musicales en temps réel par la géométrie de l'information. 2013.

J. Mairal. Incremental majorization-minimization optimization with application to large-scale machine learning. *arXiv:1402.4419 [cs, math, stat]*, Feb. 2014. arXiv: 1402.4419.

J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, Mar. 2010.

K. P. Murphy. Hidden semi-markov models (hsmms). *unpublished notes*, 2002.

R. Neal and G. E. Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*, pages 355–368. Kluwer Academic Publishers, 1998.

F. Nielsen and R. Nock. Sided and symmetrized bregman centroids. *IEEE Transactions on Information Theory*, 55(6):2882–2904, June 2009.

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

R. T. Rockafellar. *Convex analysis*. Number 28. Princeton university press, 1997.

Y. Stylianou and A. Syrdal. Perceptual and objective detection of discontinuities in concatenative speech synthesis. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01)*, volume 2, pages 837–840 vol.2, 2001.

A. Tartakovsky, I. Nikiforov, and M. Basseville. *Sequential Analysis: Hypothesis Testing and Changepoint Detection*. Monographs on Statistics & Applied Probability (Vol 136). Chapman and Hall/CRC, Boca Raton, 1 edition edition, Aug. 2014.

Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical dirichlet processes. *Journal of the american statistical association*, 101(476), 2006.

G. Tzanetakis and P. Cook. Multifeature audio segmentation for browsing and annotation. In *Applications of Signal Processing to Audio and Acoustics, 1999 IEEE Workshop on*, pages 103–106. IEEE, 1999.

M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.

C. F. J. Wu. On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11 (1):95–103, Mar. 1983.

L. Xu and M. I. Jordan. On convergence properties of the EM algorithm for gaussian mixtures. *Neural computation*, 8(1):129–151, 1996.

S. Yildirim, S. S. Singh, and A. Doucet. An online expectation-maximization algorithm for changepoint models. *Journal of Computational and Graphical Statistics*, (doi:), Apr. 2012.

S.-Z. Yu. Hidden semi-markov models. *Artificial Intelligence*, 174(2):215–243, 2010.