M2 Statistics & Data Science

Advanced Statistics & Machine Learning

Faicel Chamroukhi
Professeur
https://chamroukhi.com/

Overview

Models for sequential data

- Markov chains
- Hidden Markov Models (HMMs)
- Types of HMMs
- Parameter estimation for HMMs
- Inference in HMMs
- Viterbi algorithm

Sequential data modeling

- Until now we have considered independence assumption for the observations which were assumed to be independent and identically distributed (i.i.d).
- Now we will relax this assumption by allowing a dependence between the data : the data are supposed to be an observation sequence and therefore ordered in the time.

Markov Chains

- Markov chains are a statistical modeling approach for sequences
- A Markov chain is a sequence of n random variables $(z_1, \ldots, z_n)$, generally referred to as the *states* of the chain, verifying the Markov property that is, the current state given the previous state sequence depends only on the previous state :

$$p(z_t|z_{t-1}, z_{t-2}, \ldots, z_1) = p(z_t|z_{t-1}) \ \forall t > 1.$$

- The probabilities $p(.|.)$ computed from the distribution p are called the *transition probabilities*.
- When the transition probabilities do not depend on t, the chain is called a *homogeneous* or *a stationary* Markov chain.

Markov Chains

- The standard Markov chain can be extended by assuming that the current state depends on a history of the state sequence, in this cas one can speak about high order Markov chains (see for example the thesis of (Muri, 1997)).
- A Markov chain of order p, p being a finite integer, can be defined as

$$p(z_t|z_{t-1}, z_{t-2}, \ldots, z_1) = p(z_t|z_{t-1}, \ldots, z_{t-p}) \; \forall t > p.$$

Hidden Markov Model (HMM)

- Markov chains are often integrated in a statistical latent data model for sequential data where the hidden sequence is assumed to be a Markov chain.
- The resulting model is the so-called hidden Markov model (HMM)
- Hidden Markov Models (HMMs) are a class of latent data models widely used in many application domains, including speech recognition, image analysis, time series prediction, etc Rabiner (1989); Derrode and Pieczynski (2006), etc.
- data are no longer assumed to be independent.
- It can be seen as a generalization of the mixture model by relaxing the independence assumption.
- Let us denote by $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_n)$ the observation sequence where the multidimensional data example $\mathbf{y}_t$ is observed data at time t, and let us denote by $\mathbf{z} = (z_1, \ldots, z_n)$ the hidden state sequence where the discrete random variable z_t which takes its values in the finite set $\mathcal{Z} = \{1, \ldots, K\}$ represents the unobserved state associated with $\mathbf{y}_t$.

Hidden Markov Model (HMM)

- An HMM is fully determined by :
  - the initial distribution $\pi = (\pi_1, \ldots, \pi_K)$ where $\pi_k = p(z_1 = k)$; $k \in \{1, \ldots, K\}$,
  - the matrix of transition probabilities $\mathbf{A}$ where $\mathbf{A}_{\ell k} = p(z_t = k | z_{t-1} = \ell)$ for $t = 2, \ldots, n$, satisfying $\sum_k \mathbf{A}_{\ell k} = 1$,
  - the set of parameters $(\boldsymbol{\Psi}_1, \ldots, \boldsymbol{\Psi}_K)$ of the parametric conditional probability densities of the observed data $p(\mathbf{y}_t | z_t = k; \boldsymbol{\Psi}_k)$ for $t = 1, \ldots, n$ and $k = 1, \ldots, K$. These probabilities are also called the *emission probabilities*.
- e.g., a Gaussian HMM :

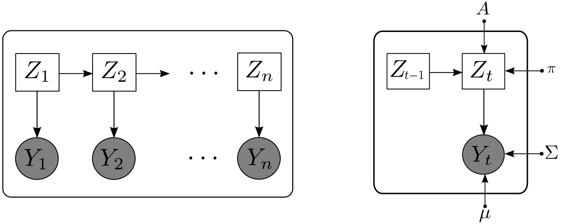

Figure – Graphical model structure for a Gaussian HMM.

Types of Hidden Markov Models

- HMMs can be classified according to the properties of their hidden Markov chain and the type of the emission state distribution.
- Homogeneous HMMs : models for which the hidden Markov chain has a stationary transition matrix.
- Non-homogeneous HMMs arise in the case when a temporal dependency is assumed for the HMM transition probabilities. (Diebold et al., 1994; Hughes et al., 1999; Meila and Jordan, 1996)
- Left-right HMMs : the states proceed from left to right according to the state indexes in a successive manner, for example such as in speech signals (Rabiner and Juang, 1993; Rabiner, 1989)
  $\Rightarrow$ imposing some restriction for the model through imposing particular constraints on the transition matrix : e.g.,

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & 0 \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{pmatrix}.$$

Types of Hidden Markov Models

- high order HMMs : when the current state depends on a finite history of the HMM states rather than only on the previous one
- Input Output HMMs (IOHMMs) (Bengio and Frasconi, 1995, 1996)
- Autoregressive HMM further generalize the standard HMMs by allowing the observations to be Autoregressive Markov chains (Muri, 1997; Rabiner, 1989; Juang and Rabiner, 1985; Celeux et al., 2004; Frühwirth-Schnatter, 2006).
- Another HMM extension lies in the Semi-Markov HMM Murphy (2002) which is like an HMM except each state can emit a sequence of observations.

Parameter estimation for a HMM

- $\boldsymbol{\Psi} = (\pi, \mathbf{A}, \boldsymbol{\Psi}_1, \ldots, \boldsymbol{\Psi}_K)$: the model parameter vector to be estimated.

- The parameter estimation is performed by maximum likelihood.
- The observed-data log-likelihood to be maximized is given by :

$$\begin{aligned}\mathcal{L}(\boldsymbol{\Psi}) &= \log p(\mathbf{Y}; \boldsymbol{\Psi}) = \log \sum_{\mathbf{z}} p(\mathbf{Y}, \mathbf{z}; \boldsymbol{\Psi}) \\ &= \log \sum_{z_1,\ldots,z_n} p(z_1; \pi) \prod_{t=2}^{n} p(z_t | z_{t-1}; \mathbf{A}) \prod_{t=1}^{n} p(\mathbf{y}_t | z_t; \boldsymbol{\Psi}).\end{aligned}$$

- this log-likelihood is difficult to maximize directly
- $\Rightarrow$ use the EM algorithm, known as Baum Welch algorithm in the context of HMMs

Hidden Markov Model (HMM)

- the distribution of a particular configuration $\mathbf{z} = (z_1, \ldots, z_n)$ of the latent state sequence is written as

$$p(\mathbf{z}; \pi, \mathbf{A}) = p(z_1; \pi) \prod_{t=2}^{n} p(z_t | z_{t-1}; \mathbf{A}),$$

- conditional independence of the HMM : that is the observation sequence is independent given a particular configuration of the hidden state sequence
- $\Rightarrow$ the conditional distribution of the observed sequence :

$$p(\mathbf{Y} | \mathbf{z}; \mathbf{\Psi}) = \prod_{t=1}^{n} p(\mathbf{y}_t | z_t; \mathbf{\Psi}).$$

$\Rightarrow$ We then get the joint distribution (the complete-data likelihood) :

$$\begin{aligned} p(\mathbf{Y}, \mathbf{z}; \mathbf{\Psi}) &= p(\mathbf{z}; \mathbf{A}, \pi) p(\mathbf{Y} | \mathbf{z}; \theta) \\ &= p(z_1; \pi) \prod_{t=2}^{n} p(z_t | z_{t-1}; \mathbf{A}) \prod_{t=1}^{n} p(\mathbf{y}_t | z_t; \mathbf{\Psi}). \end{aligned}$$

Deriving EM for HMMs

- complete-data likelihood of $\boldsymbol{\Psi}$:

$$p(\mathbf{Y},\mathbf{z};\boldsymbol{\Psi}) = p(z_1;\pi)\prod_{t=2}^{n} p(z_t|z_{t-1};\mathbf{A})\prod_{t=1}^{n} p(\mathbf{y}_t|z_t;\boldsymbol{\Psi})$$

$$= \prod_{k=1}^{K} p(z_1 = k;\pi)^{z_{1k}} \prod_{t=2}^{n}\prod_{k=1}^{K}\prod_{\ell=1}^{K} p(z_t = k|z_{t-1} = \ell;\mathbf{A})^{z_{t-1,\ell}z_{tk}} \prod_{t=1}^{n}\prod_{k=1}^{K} p(\mathbf{y}_t|z_t = k;\boldsymbol{\Psi}_k)^{z_{tk}}$$

$$= \prod_{k=1}^{K} \pi_k^{z_{1k}} \prod_{t=2}^{n}\prod_{k=1}^{K}\prod_{\ell=1}^{K} \mathbf{A}_{\ell k}^{z_{t-1,\ell}z_{tk}} \prod_{t=1}^{n}\prod_{k=1}^{K} p(\mathbf{y}_t|z_t = k;\boldsymbol{\Psi}_k)^{z_{tk}}$$

- $z_{tk} = 1$ if $z_t = k$ (i.e $\mathbf{y}_t$ originates from the kth state at time t) and $z_{tk} = 0$ otherwise.
- complete-data log-likelihood of $\boldsymbol{\Psi}$:

$$\mathcal{L}_c(\boldsymbol{\Psi}) = \log p(\boldsymbol{Y},\mathbf{z};\boldsymbol{\Psi})$$

$$= \sum_{k=1}^{K} z_{1k}\log\pi_k + \sum_{t=2}^{n}\sum_{k=1}^{K}\sum_{\ell=1}^{K} z_{tk}z_{t-1,\ell}\log\mathbf{A}_{\ell k} + \sum_{t=1}^{n}\sum_{k=1}^{K} z_{tk}\log p(\mathbf{y}_t|z_t = k;\boldsymbol{\Psi}_k).$$

The EM (Baum-Welch) algorithm

Start with an initial parameter $\mathbf{\Psi}^{(0)}$ and repeat the E- and M- steps until convergence :

E-step : compute the expectation of the complete-data log-likelihood :

$$
\begin{aligned}
Q(\mathbf{\Psi}, \mathbf{\Psi}^{(q)}) &= \mathbb{E}\Big[\mathcal{L}_c(\mathbf{\Psi})|\mathbf{Y}; \mathbf{\Psi}^{(q)}\Big] = \sum_{k=1}^{K} \mathbb{E}\Big[z_{1k}|\mathbf{Y}; \mathbf{\Psi}^{(q)}\Big] \log \pi_k + \\
& \sum_{t=2}^{n}\sum_{k=1}^{K}\sum_{\ell=1}^{K} \mathbb{E}\Big[z_{tk}z_{t-1,\ell}|\mathbf{Y}; \mathbf{\Psi}^{(q)}\Big] \log \mathbf{A}_{\ell k} + \sum_{t=1}^{n}\sum_{k=1}^{K} \mathbb{E}\Big[z_{tk}|\mathbf{Y}; \mathbf{\Psi}^{(q)}\Big] \log p(\mathbf{y}_t|z_t = k; \\
&= \sum_{k=1}^{K} p(z_1 = k|\mathbf{Y}; \mathbf{\Psi}^{(q)}) \log \pi_k + \sum_{t=2}^{n}\sum_{k=1}^{K}\sum_{\ell=1}^{K} p(z_t = k, z_{t-1} = \ell|\mathbf{Y}; \mathbf{\Psi}^{(q)}) \log \mathbf{A}_{\ell k} \\
& + \sum_{t=1}^{n}\sum_{k=1}^{K} p(z_t = k|\mathbf{Y}; \mathbf{\Psi}^{(q)}) \log p(\mathbf{y}_t|z_t = k; \mathbf{\Psi}_k) \\
&= \sum_{k=1}^{K} \tau_{1k}^{(q)} \log \pi_k + \sum_{t=2}^{n}\sum_{k=1}^{K}\sum_{\ell=1}^{K} \xi_{t\ell k}^{(q)} \log \mathbf{A}_{\ell k} + \sum_{t=1}^{n}\sum_{k=1}^{K} \tau_{tk}^{(q)} \log p(\mathbf{y}_t|z_t = k; \mathbf{\Psi}_k),
\end{aligned}
$$

The EM (Baum-Welch) algorithm

where

- $\tau_{tk}^{(q)} = p(z_t = k|\mathbf{Y};\boldsymbol{\Psi}^{(q)})\ \forall t = 1,\ldots,n$ and $k = 1,\ldots,K$ is the posterior probability of the state k at time t given the whole observation sequence and the current parameter estimation $\boldsymbol{\Psi}^{(q)}$. The τ_{tk}'s are also referred to as the *smoothing probabilities*,

- $\xi_{t\ell k}^{(q)} = p(z_t = k, z_{t-1} = \ell|\mathbf{Y};\boldsymbol{\Psi}^{(q)})\ \forall t = 2,\ldots,n$ and $k,\ell = 1,\ldots,K$ is the joint posterior probability of the state k at time t and the state ℓ at time $t-1$ given the whole observation sequence and the current parameter estimation $\boldsymbol{\Psi}^{(q)}$.

- As shown in the expression of the Q-function, this step requires the computation of the posterior probabilities $\tau_{tk}^{(q)}$ and $\xi_{t\ell k}^{(q)}$.
- $\Rightarrow$ These posterior probabilities are computed by the forward-backward recursions.

Forward-Backward

- The forward procedure computes recursively the probabilities

$$\alpha_{tk} = p(\mathbf{y}_1, \ldots, \mathbf{y}_t, z_t = k; \boldsymbol{\Psi}),$$

  $\Rightarrow$ the probability of observing the partial sequence $(\mathbf{y}_1, \ldots, \mathbf{y}_t)$ and ending with the state k at time t.

- $\Rightarrow$ the log-likelihood $\mathcal{L}$ can be computed after the forward pass as :

$$\log p(\mathbf{Y}; \boldsymbol{\Psi}) = \log \sum_{k=1}^{K} \alpha_{nk}.$$

Forward-Backward

- The backward procedure computes the probabilities

$$\beta_{tk} = p(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_n | z_t = k; \boldsymbol{\Psi})$$

$\Rightarrow$ the probability of observing the rest of the sequence $(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_1)$ knowing that we start with the k at time t.

- The forward and backward probabilities are computed recursively by the so-called Forward-Backward algorithm

- Notice that in practice, since the recursive computation of the α's and the β's involve repeated multiplications of small numbers which causes underflow problems, their computation is performed using a scaling technique in order to avoid underflow problems.

Posterior probabilities for an HMM

The posterior probability of the state k at time t given the whole sequence of observations $\mathbf{Y}$ and a model parameters $\mathbf{\Psi}$ is computed from the Forward-Backward and is given by

$$\begin{aligned}
\tau_{tk} &= p(z_t = k|\mathbf{Y}) \\
&= \frac{p(\mathbf{Y}, z_t = k)}{p(\mathbf{Y})} \\
&= \frac{p(\mathbf{Y}|z_t = k)p(z_t = k)}{\sum_{l=1}^{K} p(\mathbf{Y}|z_t = l)p(z_t = l)} \\
&= \frac{p(\mathbf{y}_1, \ldots, \mathbf{y}_t|z_t = k)p(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_n|z_t = k)p(z_t = k)}{\sum_{l=1}^{K} p(\mathbf{y}_1, \ldots, \mathbf{y}_t|z_t = l)p(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_n|z_t = l)p(z_t = l)} \\
&= \frac{p(\mathbf{y}_1, \ldots, \mathbf{y}_t, z_t = k)p(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_n|z_t = k)}{\sum_{l=1}^{K} p(\mathbf{y}_1, \ldots, \mathbf{y}_t, z_t = l)p(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_n|z_t = l)} \\
&= \frac{\alpha_{tk}\beta_{tk}}{\sum_{l=1}^{K} \alpha_{tl}\beta_{tl}} \cdot \qquad (1)
\end{aligned}$$

Joint posterior probabilities for an HMM

The joint posterior probabilities of the state k at time t and the state ℓ at time $t-1$ given the whole sequence of observations are therefore given by

$$\begin{aligned}
\xi_{t\ell k} &= p(z_t = k, z_{t-1} = \ell|\mathbf{Y}) \\
&= \frac{p(z_t = k, z_{t-1} = \ell, \mathbf{Y})}{p(\mathbf{Y})} \\
&= \frac{p(z_t = k, z_{t-1} = \ell, \mathbf{Y})}{\sum_{\ell=1}^{K}\sum_{k=1}^{K} p(z_t = k, z_{t-1} = \ell, \mathbf{Y})} \\
&= \frac{p(\mathbf{Y}|z_t = k, z_{t-1} = \ell)p(z_t = k, z_{t-1} = \ell)}{\sum_{\ell=1}^{K}\sum_{k=1}^{K} p(\mathbf{Y}|z_t = k, z_{t-1} = \ell)p(z_t = k, z_{t-1} = \ell)} \\
&= \frac{p(\mathbf{y}_1, \ldots, \mathbf{y}_{t-1}, \mathbf{y}_t, \mathbf{y}_{t+1}, \ldots, \mathbf{y}_1|z_t = k, z_{t-1} = \ell)p(z_t = k, z_{t-1} = \ell)}{\sum_{\ell=1}^{K}\sum_{k=1}^{K} p(\mathbf{Y}|z_t = k, z_{t-1} = \ell)p(z_t = k, z_{t-1} = \ell)} \\
&= \frac{\alpha_{(t-1)\ell}\, p(\mathbf{y}_t|z_t = k)\beta tk A_{\ell k}}{\sum_{\ell=1}^{K}\sum_{k=1}^{K} \alpha_{(t-1)\ell}\, p(\mathbf{y}_t|z_t = k)\beta tk A_{\ell k}}\,. \qquad (2)
\end{aligned}$$

Forward-Backward

- The posterior probabilities are then expressed in function of the forward backward probabilities as follows :

$$\tau_{tk}^{(q)} = \frac{\alpha_{tk}^{(q)} \beta_{tk}^{(q)}}{\sum_{k=1}^{K} \alpha_{tk}^{(q)} \beta_{tk}^{(q)}}$$

and

$$\xi_{t\ell k}^{(q)} = \frac{\alpha_{t-1,\ell}^{(q)} p(\mathbf{y}_t | z_t = k; \boldsymbol{\theta}^{(q)}) \beta_{tk}^{(q)} \mathbf{A}_{\ell k}^{(q)}}{\sum_{\ell=1}^{K} \sum_{k=1}^{K} \alpha_{t-1,\ell}^{(q)} p(\mathbf{y}_t^{(q)} | z_t = k; \boldsymbol{\Psi}) \beta_{tk}^{(q)} \mathbf{A}_{\ell k}^{(q)}}.$$

Forward Recursion

$$
\begin{aligned}
\alpha_{tk} &= p(\mathbf{y}_1, \ldots, \mathbf{y}_t, z_t = k) \\
&= p(\mathbf{y}_1, \ldots, \mathbf{y}_t | z_t = k) p(z_t = k) \\
&= p(\mathbf{y}_1, \ldots, \mathbf{y}_{t-1} | z_t = k) p(\mathbf{y}_t | z_t = k) p(z_t = k) \\
&= p(\mathbf{y}_1, \ldots, \mathbf{y}_{t-1}, z_t = k) p(\mathbf{y}_t | z_t = k) \\
&= \sum_{\ell=1}^{K} p(\mathbf{y}_1, \ldots, \mathbf{y}_{t-1}, z_{t-1} = \ell, z_t = k) p(\mathbf{y}_t | z_t = k) \\
&= \sum_{\ell=1}^{K} p(\mathbf{y}_1, \ldots, \mathbf{y}_{t-1} | z_{t-1} = \ell) p(z_t = k, z_{t-1} = \ell) p(\mathbf{y}_t | z_t = k) \\
&= \sum_{\ell=1}^{K} p(\mathbf{y}_1, \ldots, \mathbf{y}_{t-1}, z_t = k | z_{t-1} = \ell) p(z_t = k | z_{t-1} = \ell) p(z_{t-1} = \ell) p(\mathbf{y}_t | z_t = \\
&= \sum_{\ell=1}^{K} p(\mathbf{y}_1, \ldots, \mathbf{y}_{t-1}, z_{t-1} = \ell) p(z_t = k | z_{t-1} = \ell) p(\mathbf{y}_t | z_t = k) \\
&= \Big[\sum_{\ell=1}^{K} \alpha_{(t-1)\ell} A_{\ell k} \Big] p(\mathbf{y}_t | z_t = k) \qquad (
\end{aligned}
$$

Backward Recursion

$$
\begin{aligned}
\beta_{t\ell} &= p(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_n | z_t = \ell) \\
&= \sum_{k=1}^{K} p(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_n, z_{t+1} = k | z_t = \ell) \\
&= \sum_{k=1}^{K} p(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_n | z_{t+1} = k, z_t = \ell) p(z_{t+1} = k | z_t = \ell) \\
&= \sum_{k=1}^{K} p(\mathbf{y}_{t+2}, \ldots, \mathbf{y}_n | z_{t+1} = k, z_t = \ell) p(z_{t+1} = k | z_t = \ell) p(\mathbf{y}_{t+1} | z_{t+1} = k) \\
&= \sum_{k=1}^{K} p(\mathbf{y}_{t+2}, \ldots, \mathbf{y}_n | z_{t+1} = k) p(z_{t+1} = k | z_t = \ell) p(\mathbf{y}_{t+1} | z_{t+1} = k) \\
&= \sum_{k=1}^{K} \beta_{(t+1)k} A_{\ell k} p(\mathbf{y}_{t+1} | z_{t+1} = k). \qquad (4)
\end{aligned}
$$

Forward-Backward

The computation of these quantities is therefore performed by the Forward Backward procedure. For all $\ell, k = 1, \ldots, K$:

For all $\ell, k = 1, \ldots, K$:

1. **Forward procedure**
   - $\alpha_{1k} = p(\mathbf{y}_1, z_1 = 1; \mathbf{\Psi}) = p(z_1 = 1)p(\mathbf{y}_1|z_1 = 1; \boldsymbol{\theta}) = \pi_k p(\mathbf{y}_1|z_1 = k; \boldsymbol{\theta})$ **for** $t = 1$**,**
   - $\alpha_{tk} = [\sum_{\ell=1}^{K} \alpha_{(t-1)\ell} A_{\ell k}] p(\mathbf{y}_t|z_t = k; \mathbf{\Psi}) \quad \forall\ t = 2, \ldots, n$**.**
2. **Backward procedure**
   - $\beta_{nk} = 1$ **for** $t = n$**,**
   - $\beta_{t\ell} = \sum_{k=1}^{K} \beta_{(t+1)k} A_{\ell k} p(\mathbf{y}_{t+1}|z_{t+1} = k; \mathbf{\Psi}) \quad \forall\ t = n-1, \ldots, 1$**.**

The EM (Baum-Welch) algorithm

M-step : update the value of $\mathbf{\Psi}$ by computing the parameter $\mathbf{\Psi}^{(q+1)}$ maximizing the expectation Q-function with respect to $\mathbf{\Psi}$. The Q-function can be decomposed as

$$Q(\mathbf{\Psi}, \mathbf{\Psi}^{(q)}) = Q_\pi(\pi, \mathbf{\Psi}^{(q)}) + Q_{\mathbf{A}}(\mathbf{A}, \mathbf{\Psi}^{(q)}) + \sum_{k=1}^{K} Q(\mathbf{\Psi}_k, \mathbf{\Psi}^{(q)})$$

with

$$\begin{aligned}
Q_\pi(\pi, \mathbf{\Psi}^{(q)}) &= \sum_{k=1}^{K} \tau_{1k}^{(q)} \log \pi_k, \\
Q_{\mathbf{A}}(\mathbf{A}, \mathbf{\Psi}^{(q)}) &= \sum_{t=2}^{n} \sum_{k=1}^{K} \sum_{\ell=1}^{K} \xi_{t\ell k}^{(q)} \log \mathbf{A}_{\ell k}, \\
Q_{\mathbf{\Psi}_k}(\mathbf{\Psi}, \mathbf{\Psi}^{(q)}) &= \sum_{t=1}^{n} \tau_{tk}^{(q)} \log p(\mathbf{y}_t | z_t = k; \bar{\ }_k).
\end{aligned}$$

- The maximization of $Q(\mathbf{\Psi}, \mathbf{\Psi}^{(q)})$ with respect to $\mathbf{\Psi}$ is then performed by separately maximizing $Q_\pi(\pi, \mathbf{\Psi}^{(q)})$, $Q_{\mathbf{A}}(\mathbf{A}, \mathbf{\Psi}^{(q)})$ and $Q_{\mathbf{\Psi}_k}(\mathbf{\Psi}, \mathbf{\Psi}^{(q)})$ $(k = 1, \ldots, K)$.

- The updating formulas for the Markov chain parameters are given by :

$$\begin{aligned}
\pi_k^{(q+1)} &= \arg\max_{\pi_k} Q_\pi(\pi, \mathbf{\Psi}^{(q)}) \text{ subject to} \sum_k \pi_k = 1 \\
&= \tau_{1k}^{(q)} \\
\mathbf{A}_{\ell k}^{(q+1)} &= \arg\max_{\boldsymbol{A}_{\ell k}} Q_{\boldsymbol{A}}(\mathrm{s1}, \mathbf{\Psi}^{(q)}) \text{ subject to} \sum_k \boldsymbol{A}_{\ell k} = 1 \\
&= \frac{\sum_{t=2}^n \xi_{tk\ell}^{(q)}}{\sum_{t=2}^n \sum_k \xi_{t\ell k}^{(q)}} = \frac{\sum_{t=2}^n \xi_{tk\ell}^{(q)}}{\sum_{t=2}^n \tau_{t\ell}^{(q)}}
\end{aligned}$$

These constrained maximizations are solved using Lagrange multipliers.

- The maximization of $Q(\boldsymbol{\Psi}, \boldsymbol{\Psi}^{(q)})$ with respect to $Q_{\boldsymbol{\Psi}_k}(\boldsymbol{\Psi}, \boldsymbol{\Psi}^{(q)})$ $(k = 1, \ldots, K)$ depends on the form of emission probability function. Foa example, for the Gaussian case where $p(\boldsymbol{y}_t | z_t = k; \boldsymbol{\Psi}_k = \mathcal{N}(\mathbf{y}_t; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, we have the following updating formulas :

- The updating formulas are given by :

$$
\begin{aligned}
\boldsymbol{\mu}_k^{(q+1)} &= \frac{1}{\sum_{t=1}^{n} \tau_{tk}^{(q)}} \sum_{t=1}^{n} \tau_{tk}^{(q)} \mathbf{y}_t \\
\boldsymbol{\Sigma}_k^{(q+1)} &= \frac{1}{\sum_{t=1}^{n} \tau_{tk}^{(q)}} \sum_{t=1}^{n} \tau_{tk}^{(q)} (\mathbf{y}_t - \boldsymbol{\mu}_k^{(q+1)})(\mathbf{y}_t - \boldsymbol{\mu}_k^{(q+1)})^T.
\end{aligned}
$$

Gaussian HMM

- an HMM with Gaussian emission probabilities :

$$\mathbf{y}_t = \boldsymbol{\mu}_{z_t} + \boldsymbol{\epsilon}_t \quad ; \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{z_t}),$$

- the latent sequence $\mathbf{z} = (z_1, \ldots, z_n)$ is drawn from a first-order homogeneous Markov chain
- the ϵ_t are independent random variables distributed according to a Gaussian distribution with zero mean and covariance matrix $\boldsymbol{\Sigma}_{z_t}$.
- the state conditional density $p(\mathbf{y}_t|z_t = k; \boldsymbol{\Psi}_k)$ is Gaussian :

$$p(\mathbf{y}_t|z_t = k; \boldsymbol{\Psi}_k) = \mathcal{N}(\mathbf{y}_t; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

where $\boldsymbol{\Psi}_k = (\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$.

Gaussian HMM

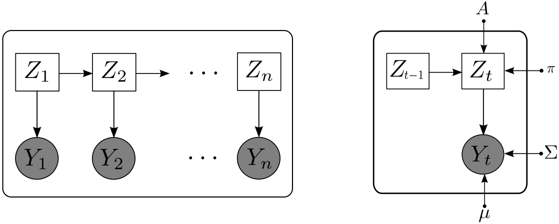

Figure – Graphical model structure for a Gaussian HMM.

- The model parameters are learned in a maximum likelihood framework by the EM algorithm.
- EM (Baum-Welch in this context of HMMs) includes forward-backward recursions to compute the E-Step
- the M-step is performed in a similar way as for a Gaussian mixture

Viterbi decoding algorithm I

Recall that we have three basic problems associated with HMMs :

1. Find $p(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n; \boldsymbol{\Psi})$, that is the likelhiood for an observation sequence $\boldsymbol{Y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n)$ given an HMM ($\boldsymbol{\Psi}$) : **an evaluation problem**.
   $\Rightarrow$ As seen previously, we use the forward (or the backward) procedure for this since it is much more efficient than direct evaluation.
2. Find an HMM ($\boldsymbol{\Psi}$) given an observation sequence $(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n)$: **a Learning problem**
   $\Rightarrow$ As seen before, the Baum-Welch (EM) algorithm solves this problem,
3. Given an observation sequence $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$ and a HMM ($\boldsymbol{\Psi}$), find the most likely state sequence $\mathbf{z} = (z_1, \ldots, z_n)$ that have generated $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$ under $\boldsymbol{\Psi}$: **an Inference problem**.
   $\Rightarrow$ As we can see it now, the Viterbi algorithm solves this problem

Viterbi decoding algorithm II

The Viterbi algorithm (Viterbi, 1967; Forney, 1973) provides an efficient dynamic programming approach to computing the most likely state sequence $(\hat{z}_1, \ldots, \hat{z}_n)$ that have generated an observation sequence $(\mathbf{y}_1, \ldots, \mathbf{y}_n)$, given a set of HMM parameters $(\mathbf{\Psi})$.

It estimates the following MAP state sequence :

$$
\begin{aligned}
\hat{\mathbf{z}} &= \arg\max_{z_1, \ldots, z_n} p(\mathbf{y}_1, \ldots, \mathbf{y}_n, z_1, \ldots, z_n; \mathbf{\Psi}) \\
&= \arg\max_{z_1, \ldots, z_n} p(z_1) p(\mathbf{y}_1 | z_1) \prod_{t=2}^{n} p(z_t | z_{t-1}) p(\mathbf{y}_t | z_t) \\
&= \arg\min_{z_1, \ldots, z_n} \Big[-\log \pi - \log p(\mathbf{y}_1 | z_1) + \sum_{t=2}^{n} -\log p(z_t | z_{t-1}) - \log p(\mathbf{y}_t | z_t) \Big].
\end{aligned}
$$

The Viterbi algorithm works on the dynamic programming principle that is :

Viterbi decoding algorithm III

The minimum cost path to $z_t = k$ is equivalent to the minimum cost path to node z_{t-1} plus the cost of a transition from z_{t-1} to $z_t = k$ (and the cost incurred by observation $\mathbf{y}_t$ given $z_t = k$).

The MAP state sequence is then determined by starting at node z_t and reconstructing the optimal path backwards based on the stored calculations.

Viterbi decoding reduces the computation cost to $\mathcal{O}(K^2 n)$ operations instead of the brute force $O(K^n)$ operations. The Viterbi algorithm steps are outlined in Algorithm 1.

Viterbi decoding algorithm IV

Algorithm 1 Pseudo code of the Viterbi algorithm for an HMM.

Inputs : Observations $(\mathbf{y}_1, \ldots, \mathbf{y}_n)$ and HMM params $\boldsymbol{\Psi}$

1: Initialization : initialize minimum path sum to state $z_1 = k$ for $k = 1, \ldots, K$:

$$S_1(z_1 = k) = -\log \pi_k - \log p(\mathbf{y}_1 | z_1 = k)$$

2: Recursion : for $t = 2, \ldots, n$ and for $k = 1, \ldots, K$, calculate the minimum path sum to state $z_t = k$:

$$S_t(z_t = k) = -\log p(\mathbf{y}_t | z_t = k) + \min_{z_{t-\mathbf{1}}} \left[S_{t-1}(z_{t-1}) - \log p(z_t = k | z_{t-1}) \right]$$

and let
$$z^*_{t-1}(z_t) = \arg\min_{z_{t-\mathbf{1}}} \left[S_{t-1}(z_{t-1}) - \log p(z_t = k | z_{t-1}) \right]$$

3: Termination : compute $\min_{z_n} S_n(z_n)$ and set $\hat{z}_n = \arg\min_{z_n} S_n(z_n)$

4: State sequence backtracking : iteratively set, for $t = n-1, \ldots, 1$

$$\hat{z}_t = z^*_t(\hat{z}_{t+1})$$

Outputs : The most likely state sequence $(\hat{z}_1, \ldots, \hat{z}_n)$.

Bibliography I

Bengio, Y. and Frasconi, P. (1995). An input output hmm architecture. In Advances in Neural Information Processing Systems, volume 7, pages 427–434.

Bengio, Y. and Frasconi, P. (1996). Input Output HMM's for sequences processing. IEEE Transactions on Neural Networks, 7(5).

Celeux, G., Nascimento, J., and Marques, J. (2004). Learning switching dynamic models for objects tracking. PR, 37(9) :1841–1853.

Derrode, S. Benyoussef, L. and Pieczynski, W. (2006). Contextual estimation of hidden markov chains with application to image segmentation. In ICASSP, pages 15–19, Toulouse.

Diebold, F., Lee, J.-H., and Weinbach, G. (1994). Regime switching with time-varying transition probabilities. Nonstationary Time Series Analysis and Cointegration. (Advanced Texts in Econometrics), pages 283–302.

Forney, G. D. (1973). The viterbi algorithm. Proceedings of the IEEE, 61(3) :268–278.

Frühwirth-Schnatter, S. (2006). Finite Mixture and Markov Switching Models (Springer Series in Statistics). Springer Verlag, New York.

Hughes, J. P., Guttorp, P., and Charles, S. P. (1999). A non-homogeneous hidden markov model for precipitation occurrence. Applied Statistics, 48 :15–30.

Juang, B.-H. and Rabiner, L. R. (1985). Mixture autoregressive hidden markov models for speech signals. IEEE Transactions on Acoustics, Speech and Signal Processing, 33(6) :1404–1413.

Meila, M. and Jordan, M. I. (1996). Learning fine motion by markov mixtures of experts. In Advances in Neural Information Processing Systems 8, pages 1003–1009. MIT Press.

Muri, F. (1997). Comparaison d'algorithmes d'identification de chaînes de Markov cachées et application à la détection de régions homogènes dans les séquences ADN. PhD thesis, Université Paris Descartes, Paris V.

Murphy, K. P. (2002). Dynamic Bayesian Networks : Representation, Inference and Learning. PhD thesis, UC Berkeley, Computer Science Division.

Rabiner, L. and Juang, B.-H. (1993). Fundamentals of speech recognition. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

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

Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Transactions on Information Theory, 13(2) :260–269.