# References

[1] Olivier Cappe, Eric Moulines, and Tobias Ryden. *Inference in Hidden Markov Models*. Springer, 2009.

[2] Arthur Dempster, Nan Laird, and Donald Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977.

[3] Gain Han and Keemin Sohn. Activity imputation for trip-chains elicited from smart-card data using a continuous hidden markov model. *Elsevier Transportation Research*, 83:121–135, 2016.

[4] Z Lin, M Yin, S Feygin, M Sheehan, and JF Paiement. Deep generative models of urban mobility. *UC Berkeley, Department of Transportation*, 2017.

[5] Qiujian Lv, Yuanyuan Qiao, Nirwan Ansari, Jun Liu, and Jie Yang. Big data driven hidden markov model based individual mobility prediction at points of interest. *IEEE Transactions on Vehicular Technology*, PP:1–1, 2016.

[6] Fabio Mazzarella, Virginia Fernandez Arguedas, and Michele Vespe. Knowledge-based vessel position prediction using historical AIS data. *Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, 2015.

[7] Giuliana Pallotta, Michele Vespe, and Karna Bryan. Vessel pattern knowledge discovery from AIS data: a framework for anomaly detection and route prediction. *Entropy*, 15:2218–2245, 2013.

[8] Lawrence Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77:257–286, 1989.

[9] Mogeng Yin, Madeleine Sheehan, and Sidney Feygin. A generative model of urban activities from cellular data. *IEEE Transactions on Intelligent Transportation Systems*, PP:1–15, 2017.

[10] Mohamed Zraiaa. Hidden markov models : A continuous-time version of the Baum-Welch algorithm. *Imperial College London, Department of Computing*, 2010.

# A The Baum-Welch Algorithm

Training our model means finding the parameters that best fit the observations. Hence we want to find $\theta^*$ that maximizes the likelihood of the observations $L_\theta(o_{0:T})$.

The likelihood of the model is:

$$\begin{aligned} L(\theta) &= \mathbb{P}_\theta(o_0, \ldots, o_T) \\ &= \sum_{\text{all possible } x_{0:T}} \mathbb{P}_\lambda(o_{0:T} \mid x_{0:T}) \times \mathbb{P}_\lambda(x_{0:T}) \\ &= \sum_{\text{all possible } x_{0:T}} \left( \prod_{t=0}^{T} p(o_t|x_t, \lambda) \right) \left( \pi_{x_0} \times \prod_{t=1}^{T} a_{x_{t-1}, x_t} \right) \\ &= \sum_{\text{all possible } x_{0:T}} \left( \prod_{t=0}^{T} \sum_{k=1}^{K} g_{x_t,k} p(o_t|\mu_k, \Sigma_k) \right) \left( \pi_{x_0} \times \prod_{t=1}^{T} a_{x_{t-1}, x_t} \right) \end{aligned} \tag{A.0.1}$$

Note that computing this quantity is not an easy task. Fortunately, Dempster *et al.* devised a strategy [2], in the form of the Expectation-Maximization (EM) algorithm.

## A.1 The Expectation-Maximization algorithm

The EM algorithm is an iterative way to find parameters that maximize the likelihood of the model. As we saw, maximizing the likelihood directly is impossible. The prowess behind the EM algorithm is to solve this issue by defining another quantity, $Q$, whose maximization is possible and guarantees the maximization of the likelihood. $Q$ is defined as:

$$Q(\theta, \theta') = \mathbb{E}_{\theta'} \left[ \log L_\theta \mid o_{0:T} \right] \tag{A.1.1}$$

With $L_\theta$ the *complete likelihood* of the model:

$$L_\theta = L(X_{0:t} = x_{0:T},\, O_{0:T} = o_{0:T}\, ;\, \theta) \tag{A.1.2}$$

where $X$ represents the hidden variables, and $O$ the observed variables. Note that $L(\theta) = \sum_{x_{0:T}} L_\theta$.

The EM algorithm looks for $\theta^*$ that maximizes $Q(\theta, \theta')$ over $\theta$ at each step. It can be shown that $L(\theta^*) - L(\theta') \geq Q(\theta^*, \theta') - Q(\theta', \theta')$ (see [2] for a complete proof). Hence, maximizing $Q$ leads to the maximization of $L$.

The layout of the algorithm is thus:

1. Initialize $\theta$ with random values.

2. Do until convergence:

   **Estimation** Compute $Q(\theta, \theta^s) = \sum_{x_{1:T}} \log[\mathbb{P}_\theta(o_{1:T})] \times \mathbb{P}_{\theta^s}(x_{1:T}, o_{1:T})$

   **Maximization** Set $\theta^{s+1} \leftarrow \arg\max_\theta Q(\theta, \theta^s)$

The algorithm is converged when the difference between one step and the following one falls below a given threshold.

## A.2 Application to the HMM : the Baum-Welch Algorithm

In this part, we will describe the complete derivation of the algorithm for the case of the simple HMM, whose framework is reminded figure 1. The derivation for more complex models does not present any specific difficulty and will thus be omitted. However, the reader can refer to [10, 1] for a more comprehensive explanation.

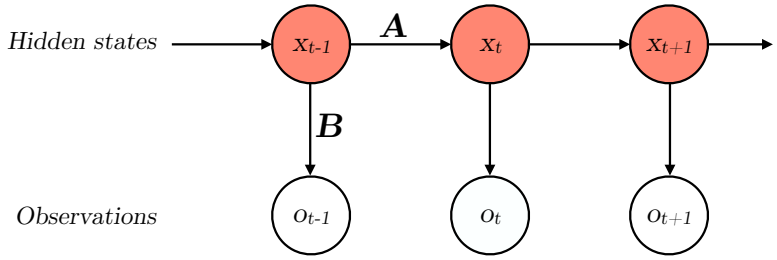


Figure 16: Schematics of an HMM

The complete likelihood of the model is:

$$\begin{aligned} L_\theta(x_{0:T},\, o_{0:T}) &= L_\theta(o_{0:T} \mid x_{0:T}) \times L_\theta(x_{0:T}) \\ &= \prod_{t=0}^{T} \mathbb{P}_\theta(O_t = o_t \mid X_t = x_t) \times \pi_{x_0} \cdot \prod_{t=1}^{T} \mathbb{P}_\theta(X_t = x_t \mid X_{t-1} = x_{t-1}) \\ &= \prod_{t=0}^{T} b_{x_t o_t} \times \pi_{x_0} \cdot \prod_{t=1}^{T} a_{x_{t-1} x_t} \end{aligned} \tag{A.2.1}$$

Hence the complete log-likelihood:

$$\log L_\theta = \sum_{t=0}^{T} \log b_{x_t o_t} + \log \pi_{x_0} + \sum_{t=1}^{T} \log a_{x_{t-1} x_t} \tag{A.2.2}$$

Equations (A.1.1) and (A.2.2) yield:

$$\begin{aligned} Q(\theta, \theta') = &\underbrace{\sum_{x_{0:T}} \sum_{t=0}^{T} \log b_{x_t o_t} \cdot \mathbb{P}_{\theta'}(X_{0:T} = x_{0:T} \mid O_{0:T} = o_{0:T})}_{T_B} \\ &+ \underbrace{\sum_{x_{0:T}} \log \pi_{x_0} \cdot \mathbb{P}_{\theta'}(X_{0:T} = x_{0:T} \mid O_{0:T} = o_{0:T})}_{T_\pi} \\ &+ \underbrace{\sum_{x_{0:T}} \sum_{t=1}^{T} \log a_{x_{t-1} x_t} \cdot \mathbb{P}_{\theta'}(X_{0:T} = x_{0:T} \mid O_{0:T} = o_{0:T})}_{T_A} \end{aligned} \tag{A.2.3}$$

We can further simplify the equation:

$$\begin{aligned} T_B &= \sum_{t=0}^{T} \sum_{x_t} \log b_{x_t o_t} \underbrace{\sum_{x_{0:t-1},\, x_{t+1:T}} \mathbb{P}_{\theta'}(X_{0:T} \mid O_{0:T})}_{\mathbb{P}_{\theta'}(X_t \mid O_t)} \\ &= \sum_{t=0}^{T} \sum_{x_t} \log b_{x_t o_t} \cdot \mathbb{P}_{\theta'}(X_t \mid O_{0:T}) \end{aligned} \tag{A.2.4}$$

Doing a similar simplification on $T_\pi$ and $T_A$ yields:

$$T_\pi = \sum_{x_0} \log \pi_{x_0} \cdot \mathbb{P}_{\theta'}(X_0 = x_0 \mid O_{0:T}) \tag{A.2.5}$$

$$T_A = \sum_{t=1}^{T} \sum_{x_{t-1}, x_t} \log a_{x_{t-1} x_t} \cdot \mathbb{P}_{\theta'}(X_{t-1} = x_{t-1},\ X_t = x_t \mid O_{0:T}) \tag{A.2.6}$$

$T_\pi$, $T_A$ and $T_B$ depend on distinct variables, and can be maximized independently.

Let us maximize $T_\pi$ under the constraint $\sum_n \pi_n = 1$ by introducing the lagrangian multiplier $\nu$:

$$\frac{\partial}{\partial \pi_i} \left( T_\pi - \nu \left( \sum_n \pi_n - 1 \right) \right) = \frac{\partial T_\pi}{\pi_i} - \nu = \frac{1}{\pi_i} \times \mathbb{P}_\theta(X_0 = i \mid O_{0:T}) - \nu$$

Hence:

$$\hat{\pi}_x = \frac{\mathbb{P}_\theta(X_0 = x \mid O_{0:T})}{\sum_{\tilde{x}} \mathbb{P}_\theta(X_0 = \tilde{x} \mid O_{0:T})} \tag{A.2.7}$$

Similarly:

$$\hat{a}_{x,x'} = \frac{\sum_{t=1}^{T} \mathbb{P}_\theta(X_{t-1} = x,\ X_t = x' \mid O_{0:T})}{\sum_{\tilde{x}} \sum_{t=1}^{T} \mathbb{P}_\theta(X_{t-1} = x,\ X_t = \tilde{x} \mid O_{0:T})} \tag{A.2.8}$$

$$\hat{b}_{x,o} = \frac{\sum_{t=0}^{T} \delta_{o,o_t} \mathbb{P}_\theta(X_t = x \mid O_{0:T})}{\sum_{\tilde{o}} \sum_{t=0}^{T} \delta_{\tilde{o},o_t} \mathbb{P}_\theta(X_t = x \mid O_{0:T})} \tag{A.2.9}$$

In the case of the CHMM as described in section 3, the complete derivation of the maximization step yields [3, 10, 1]:

$$\hat{\pi}_i = \frac{\sum_{l=1}^{M} \mathbb{P}_{\lambda^s}(x_1 = i \mid o^l_{1:T})}{M} \tag{A.2.10}$$

$$\hat{a}_{ij} = \frac{\sum_{l=1}^{M} \sum_{t=1}^{T^l - 1} \mathbb{P}_{\lambda^s}(x_t = i,\ x_{t+1} = j \mid o^l_{1:T})}{\sum_{l=1}^{M} \sum_{t=1}^{T^l - 1} \mathbb{P}_{\lambda^s}(x_t = i \mid o^l_{1:T})} \tag{A.2.11}$$

$$\hat{g}_{ik} = \frac{\sum_{l=1}^{M} \sum_{t=1}^{T^l} \mathbb{P}_{\lambda^s}(x_t = i,\ m_t = k \mid o^l_{1:T})}{\sum_{l=1}^{M} \sum_{t=1}^{T^l} \mathbb{P}_{\lambda^s}(x_t = i \mid o^l_{1:T})} \tag{A.2.12}$$

$$\hat{\mu}_{kd} = \frac{\sum_{l=1}^{M} \sum_{i=1}^{N} \sum_{t=1}^{T^l} \mathbb{P}_{\lambda^s}(x_t = i,\ m_t = k \mid o^l_{1:T}) \cdot o^l_{td}}{\sum_{l=1}^{M} \sum_{i=1}^{N} \sum_{t=1}^{T^l} \mathbb{P}_{\lambda^s}(x_t = i,\ m_t = k \mid o^l_{1:T})} \tag{A.2.13}$$

$$\hat{\sigma}_{k,d_1 d_2} = \frac{\sum_{l=1}^{M} \sum_{i=1}^{N} \sum_{t=1}^{T^l} \mathbb{P}_{\lambda^s}(x_t = i,\ m_t = k \mid o^l_{1:T}) \cdot \mathrm{var}_{t,d_1 d_2}}{\sum_{l=1}^{M} \sum_{i=1}^{N} \sum_{t=1}^{T^l} \mathbb{P}_{\lambda^s}(x_t = i,\ m_t = k \mid o^l_{1:T})} \tag{A.2.14}$$

where $\mathrm{var}_{t,d_1 d_2} = (o^l_{td_1} - \hat{\mu}_{kd_1})(o^l_{td_2} - \hat{\mu}_{kd_2})$ and $\hat{\Sigma} = \begin{pmatrix} \hat{\sigma}_{k,11} & \cdots & \hat{\sigma}_{k,1D} \\ \vdots & \ddots & \vdots \\ \hat{\sigma}_{k,D1} & \cdots & \hat{\sigma}_{k,DD} \end{pmatrix}$

The quantities necessary to compute the parameters of the model are easily derived from the Forward-Backward algorithm. Note that in its original form, this algorithm suffers from severe underflow problems, which make it difficult to scale. Thus, we need to apply a normalized version of the algorithm, for example the one presented by Rabiner in his seminal paper [8].