class HMM

Here we describe our rendition and Python implementation of the well known technique of a time series fitting using the Hidden Markov models. For a review see [1].

The class is initialized by the sequence `y` of length T of the observed states (here we count as $1 \cdots T$, in Python script it is $0 \cdots T-1$), the number N of the hidden states, and the spectrum `observable_states` of the observable states. It is possible to provide the probabilities π_i of the initial hidden states, as well as the seed hidden states transition matrix a_{ij} and the emission matrix b_{ik}, otherwise all these probabilities will be initialized randomly and uniformly. The triplet (π, a, b) defines a Hidden Markov Model.

We denote the spectrum of hidden states as $\{X_i\}$, $i = 1, \ldots, N$, and the spectrum of observable states as $\{Y_k\}$, $k = 1, \ldots, M$. The initial ($t = 1$) hidden states probabilities are $P(x_1 = X_i) = \pi_i$. We observed the sequence $y = y_1 \cdots y_T$ of the observable states. This can be reformulated as $k = k_1 \cdots k_T$, where $y_t = Y_{k_t}$, which is a more practical notation for matrix manipulations.

The core functionalities of the HMM class are

- **forward**. The conditional forward probabilities

$$\hat{\alpha}_{it} = P(x_t = X_i | y_1 \cdots y_t) \,. \tag{1}$$

This is done by normalizing the joined forward probabilities

$$\alpha_{it} = P(y_1 \cdots y_t, x_t = X_i) \,, \tag{2}$$

which are calculated iteratively, starting with

$$\alpha_{i1} = \pi_i \, b_{ik_1} \,, \qquad c_1 = \sum_i \alpha_{i1} = P(y_1) \,, \qquad \hat{\alpha}_{i1} = \frac{\alpha_{i1}}{c_1} = P(x_1 = X_i | y_1) \,, \tag{3}$$

and iterating as

$$c_{t+1} \, \hat{\alpha}_{i\,t+1} = \sum_j \hat{\alpha}_{jt} \, a_{ji} \, b_{i\,k_{t+1}} \,. \tag{4}$$

The last equation is a consequence of the iterative relation for the joined forward probabilities and normalization definition of the conditional forward probability

$$\alpha_{i\,t+1} = \sum_j \alpha_{jt} \, a_{ji} \, b_{i\,k_{t+1}} \,, \qquad \hat{\alpha}_{i\,t+1} = \frac{\alpha_{i\,t+1}}{c_1 \cdots c_{t+1}} \,, \tag{5}$$

where all the normalization coefficients c_t are determined from the condition $\sum_i \hat{\alpha}_{it} = 1$, and are used to define the probability $P(y_1 \cdots y_t) = c_1 \cdots c_t$, therefore

$$c_t = P(y_t | y_1 \cdots y_{t-1}) \,. \tag{6}$$

The coefficients $\{c_t\}$ are stored in the `norms` list during calculation of α.

- **backward**. The normalized backward probabilities

$$\hat{\beta}_{it} = \frac{1}{c_{t+1} \cdots c_T} P(y_{t+1} \cdots y_T | x_t = X_i) \,, \tag{7}$$

initialized as $\hat{\beta}_{iT} = 1$, and iteratively calculated as

$$\tilde{\beta}_{it} = \sum_j \hat{\beta}_{j\,t+1} \, a_{ij} \, b_{j\,k_{t+1}} \,, \qquad \hat{\beta}_{it} = \frac{\tilde{\beta}_{it}}{c_{t+1}} \,. \tag{8}$$

- **BaumWelch**. Baum-Welch iteration to update the probabilities π_i, a_{ij}, and b_{ik}, defining the HMM. First we run the forward and the backward methods. Using the so-calculated α and β we calculate the conditional probability of the hidden state $x_t = X_i$ given all of the observations

$$\gamma_{it} = P(x_t = X_i | y_1 \cdots y_T) = \frac{\alpha_{it} \beta_{it}}{c_1 \cdots c_T} = \hat{\alpha}_{it} \hat{\beta}_{it} \,, \quad t = 1, \ldots, T \,, \tag{9}$$

which satisfies the normalization condition

$$\sum_i \gamma_{it} = 1 \,, \quad t = 1, \ldots, T \,, \tag{10}$$

and the conditional probability of the hidden state $x_t = X_i$ and the hidden state $x_{t+1} = X_j$ given all of the observations

$$\xi_{tij} = P(x_t = X_i, x_{t+1} = X_j | y_1 \cdots y_T) = \frac{\alpha_{it} \, \beta_{j\,t+1} \, a_{ij} \, b_{j\,k_{t+1}}}{c_1 \cdots c_T} \tag{11}$$

$$= \frac{1}{c_{t+1}} \, \hat{\alpha}_{it} \, \hat{\beta}_{j\,t+1} \, a_{ij} \, b_{j\,k_{t+1}} \,, \quad t = 1, \ldots, T-1 \,. \tag{12}$$

Notice that due to (8) we have

$$\sum_j \xi_{tij} = \gamma_{it} \,, \quad t = 1, \ldots, T-1 \,. \tag{13}$$

Knowing the γ and ξ matrices we can update the estimates for the parameters of the Hidden Markov Model:

$$\pi_i = \gamma_{i1} \,, \qquad a_{ij} = \frac{\sum_{t=1}^{T-1} \xi_{tij}}{\sum_{t=1}^{T-1} \gamma_{it}} \,, \qquad b_{ik} = \frac{\sum_{t=1}^{T} \delta(y_t = Y_k) \gamma_{it}}{\sum_{t=1}^{T} \gamma_{it}} \,. \tag{14}$$

- **Viterbi**. Viterbi algorithm to determine the most likely path of hidden states for the (π, a, b) HMM, given the observed sequence $k_1 \cdots k_T$. The path is saved in the `estimate_x` attribute.

  Define two probability matrices, R_{it} and Q_{it}, of size $N \times T$.

  The R_{it} stores the probability of the most likely hidden sequence $x_1 \cdots x_t$ that generates observed sequence $y_1 \cdots y_t$, such that $x_t = X_i$. It is initialized as $R_{i1} = \pi_i \, b_{ik_1}$, and iterated as

  $$R_{it} = \max_j (R_{j\,t-1} \, a_{ji}) \, b_{ik_t} \, . \tag{15}$$

  Notice that we maximize joined probability of the hidden and observed sequences, which is equivalent to maximizing posterior probability of hidden sequence given the observed sequence.

  The Q_{it} stores x_{t-1} of the most likely hidden states sequence $x_1 \cdots x_t$ which generates the observed sequence $y_1 \cdots y_t$. It can be determined iteratively as

  $$Q_{it} = \arg\max{}_j (R_{j\,t-1} \, a_{ji}) \, . \tag{16}$$

  Knowing R_{iT} we can calculate the most likely last hidden state x_T as

  $$x_T = \arg\max{}_i (R_{iT}) \, . \tag{17}$$

  Then going backward in time we can calculate the rest of the most likely hidden states as

  $$x_t = Q_{x_{t+1}\,t+1} \, . \tag{18}$$

---

[1] L. R. Rabiner, A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, *Proceedings of the IEEE*, **77**, 2, 1989.