1228 Probabilistic Graphical Models
H. Zhu and J. Boedecker
April 30, 2026

Markov Models

Contents

1 Markov chains

We consider a stochastic process

$$\{X_0, X_1, \ldots, X_t, \ldots\}$$

with random variables $X_0, X_1, \ldots, X_t, \ldots$ sharing the same domain $\mathbf{dom}\, X$. Suppose the conditional probability of any future *state* $x_{t+1} \in X$ given the past states $x_0, \ldots, x_{t-1}$ and present state x_t, is independent of the past states and depends only on the present state,

$$\mathbf{P}(x_{t+1} \mid x_0, \ldots, x_t) = \mathbf{P}(x_{t+1} \mid x_t), \tag{1.1}$$

then this stochastic process is called a *Markov chain*. The equation (1.1) is called the *Markov property*. The random variable X is called the *state space* of the Markov chain.[1] Specifically, if the conditional probability $\mathbf{P}(X_{t+1} = x_j \mid X_t = x_i)$ is independent of time t given the same x_i and x_j, the Markov chain is *time-homogeneous*. Note that in the following discussion we will use subscripts to denote both the state instance at time t, denoted as x_t, and the ith instance of random variable X (suppose we assign each entry in X a unique index), denoted as x_i. The intended meaning of the notation should be inferred from the context of its usage. We hope it will cause no confusion.

[1] In this course we will only consider the case where the state space is finite and discrete.

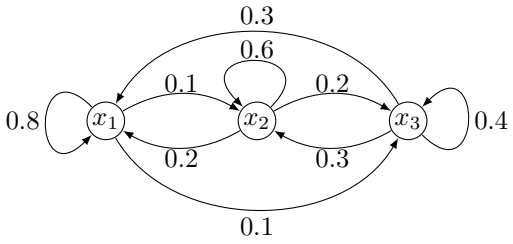

Figure 1 Example of state transition diagram of a Markov chain.

For a time-homogeneous Markov chain, let P be a square matrix with each entry

$$P_{ij} = \mathbf{P}(x_j \mid x_i),$$

for all $x_i, x_j \in X$, representing the probability that the next state is x_j given the present state x_i. Considering an m-dimensional state space X, *i.e.*, $|X| = m$, clearly the matrix P has the following property:

- $P \in \mathbf{R}^{m\times m}$.
- $P_{ij} \geq 0$ for all $i = 1, \ldots, m$, $j = 1, \ldots, m$.
- $\mathbf{1}^T P_{i:} = 1$ for all $i = 1, \ldots, m$.

Here we use $P_{i:}$ to denote the ith row of matrix P, represented as a column vector. The matrix P is called the *transition matrix* of the Markov chain, which can be represented graphically with a *state transition diagram*. The state transition diagram is a directed graph where each note is a state and the arcs represent possible transitions between states, weighted by corresponding transition probabilities. If an arc between state x_i and x_j does not appear in the diagram, it means that the corresponding transition probability P_{ij} is zero. Figure 1 shows an example of a Markov chain with a 3-dimensional state space. Note that although the state transition diagram and the graphical model diagram are both represented with graphs, they have completely different interpretations. The former represents the transition probability between states, *i.e.*, different instances of the random variable X, while the latter represents the probabilistic dependencies between different random variables.

Except for the transition matrix P, to uniquely determine an m-dimensional Markov chain[2], we will also need the vector of initial state distribution:

$$\rho = (\ldots, \mathbf{P}(X_0 = x_i), \ldots), \quad i = 1, \ldots, m.$$

Obviously, the initial state distribution $\rho \in \mathbf{R}^m$ has to satisfy $\rho \succeq 0$ and $\mathbf{1}^T\rho = 1$, where the generalized inequality symbol $\succeq$ denotes the componentwise inequality

[2]We will always assume the Markov chain to be time-homogeneous except mentioned specifically.

between two vectors. Thus the parameter set of a Markov chain can be denoted as a 2-dimensional set

$$\Theta = \{\rho, P\}.$$

1.1 Inference and parameter learning

If the parameter set Θ of a Markov chain is known, we can calculate the probability of observing any sequence of states $\zeta = \{X_0 = x_i, X_1 = x_j, X_2 = x_k, \ldots\}$ generated by that Markov chain, which is basically the product of the transition probabilities of the sequence of states:

$$\mathbf{P}(\zeta \mid \rho, P) = \rho_i P_{ij} P_{jk} \cdots .$$

On the other hand, if the parameters are unknown, but we are given a set of observed state sequences $\mathcal{D} = \{\zeta_1, \zeta_2, \ldots\}$ from that Markov chain, where $\zeta = \{x_0, \ldots, x_N\}$ for all $\zeta \in \mathcal{D}$, we can estimate the ρ and P for the Markov chain according to:

$$\rho_i = \mathop{\mathbf{E}}_{\zeta \sim \mathcal{D}} \mathbf{P}(X_0 = x_i \mid \zeta), \quad i = 1, \ldots, m, \tag{1.2}$$

and

$$P_{ij} = \frac{\mathbf{E}_{\zeta \sim \mathcal{D}, t} \mathbf{P}(X_t = x_i, X_{t+1} = x_j \mid \zeta)}{\mathbf{E}_{\zeta \sim \mathcal{D}, t} \mathbf{P}(X_t = x_i \mid \zeta)}, \quad i = 1, \ldots, m, \quad j = 1, \ldots, m. \tag{1.3}$$

Note that for the last observed state x_N in each sequence we do not observe the next state, so the above expectations are estimated across $t = 0, \ldots, N-1$.

Remark. The equations (1.2) and (1.3), which are used to learn parameters of a Markov chain based on observations, intuitively align with our understanding. It's validity can also be shown analytically as follows.

The problem of estimating the initial state distribution ρ and the transition matrix P of a Markov chain according to the set of observations $\mathcal{D}$ can be formally defined as a *maximum likelihood estimation* (MLE) problem,

$$\begin{array}{ll} \text{maximize} & l_{\mathcal{D}}(\Theta) = \mathbf{E}_{\zeta \sim \mathcal{D}} \log \mathbf{P}(\zeta \mid \rho, P) \\ \text{subject to} & \rho \succeq 0, \ \mathbf{1}^T \rho = 1 \\ & P_{ij} \geq 0, \quad i = 1, \ldots, m, \quad j = 1, \ldots, m \\ & \mathbf{1}^T P_{i:} = 1, \quad i = 1, \ldots, m, \end{array} \tag{1.4}$$

where $\Theta = \{\rho, P\}$ is the optimization variable and $\mathcal{D}$ is the problem data. The function $l_{\mathcal{D}}(\Theta)$ is called the *log-likelihood function* of model parameter Θ given the observation $\mathcal{D}$. Note that the objective function of (1.4) can be written as

$$\begin{aligned} l_{\mathcal{D}}(\Theta) &= \mathop{\mathbf{E}}_{\zeta \sim \mathcal{D}} \log \mathbf{P}(\zeta \mid \rho, P) \\ &= \mathop{\mathbf{E}}_{\zeta \sim \mathcal{D}} \left(\log \mathbf{P}(x_0 \mid \rho) \prod_{t=0}^{N-1} \mathbf{P}(x_{t+1} \mid x_t, P) \right) \end{aligned}$$

$$
\begin{aligned}
&= \mathop{\mathbf{E}}_{\zeta\sim\mathcal{D}} \log \mathbf{P}(x_0 \mid \rho) + \mathop{\mathbf{E}}_{\zeta\sim\mathcal{D}} \left(\sum_{t=0}^{N-1} \log \mathbf{P}(x_{t+1} \mid x_t, P) \right) \\
&= \mathop{\mathbf{E}}_{\zeta\sim\mathcal{D}} \left(\sum_{i=1}^{m} I_{x_i}(x_0) \log \rho_i \right) + \mathop{\mathbf{E}}_{\zeta\sim\mathcal{D}} \left(\sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{t=0}^{N-1} I_{x_i}(x_t) I_{x_j}(x_{t+1}) \log P_{ij} \right),
\end{aligned} \tag{1.5}
$$

where $I_{x_i}(x)$ is an indicator function with $I_{x_i}(x) = 1$ if $x = x_i$, and 0 otherwise. Thus problem (1.4) can be transformed into the following two optimization problems

$$
\begin{array}{ll}
\text{maximize} & \mathbf{E}_{\zeta\sim\mathcal{D}} \left(\sum_{i=1}^{m} I_{x_i}(x_0) \log \rho_i \right) \\
\text{subject to} & \rho \succeq 0,\ \mathbf{1}^T \rho = 1,
\end{array}
$$

with optimization variable ρ, and

$$
\begin{array}{ll}
\text{maximize} & \mathbf{E}_{\zeta\sim\mathcal{D}} \left(\sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{t=0}^{N-1} I_{x_i}(x_t) I_{x_j}(x_{t+1}) \log P_{ij} \right) \\
\text{subject to} & P_{ij} \geq 0, \quad i = 1, \ldots, m, \quad j = 1, \ldots, m \\
& \mathbf{1}^T P_{i:} = 1, \quad i = 1, \ldots, m,
\end{array}
$$

with optimization variable P, which are maximized by equation (1.2) and (1.3) (according to the Gibbs' inequality), respectively.

Example. Consider that we have the following observation sequences generated from the Markov chain described in figure 1:

- $(x_2, x_2, x_3, x_3, x_3, x_3, x_1)$.
- $(x_1, x_3, x_2, x_3, x_3, x_3, x_3)$.
- (x_3, x_3, x_2, x_2).
- $(x_2, x_1, x_2, x_2, x_1, x_3, x_1)$.

According to these observations, the initial state distribution can be estimated as:

$$
\rho = \left(\frac{1}{4}, \frac{2}{4}, \frac{1}{4} \right) = (0.25, 0.5, 0.25)\,,
$$

and the transition matrix of the Markov chain is

$$
P = \begin{bmatrix}
\frac{0}{3} & \frac{1}{3} & \frac{2}{3} \\
\frac{2}{7} & \frac{3}{7} & \frac{2}{7} \\
\frac{2}{11} & \frac{2}{11} & \frac{7}{11}
\end{bmatrix}.
$$

1.2 Convergence

Convergence is another useful property of Markov chains. That says, if a Markov chain with $|X| = m$ satisfies the following two requirements:

- *Irreducible.* From every state $x_i \in X$ there is a probability $P_{ij} > 0$ of transiting to any state $x_j \in X$.
- *Aperiodic.* For any given state, there isn't a fixed interval after which the chain will return to the same state.

Then this Markov chain will reduce to a unique stationary state distribution $\pi \in \mathbf{R}_+^m$ with $\mathbf{1}^T\pi = 1$ when $t \to \infty$, such that

$$\pi P = \pi.$$

In this case, the matrix P^t (the transition matrix P to the tth power) converges to a rank-one matrix in which each row is the stationary distribution π:

$$\lim_{t\to\infty} P^t = \mathbf{1}\pi^T.$$

The rate of convergence of a Markov chain is determined by the second largest eigen-value of the transition matrix P.

2 Hidden Markov models

A *hidden Markov model* (HMM) consists of a Markov chain $\{X_0, X_1, \ldots, X_t, \ldots\}$ with domain $\mathbf{dom}\, X$ whose states are not directly observable, and an observable stochastic process $\{Y_0, Y_1, \ldots, Y_t, \ldots\}$ with domain $\mathbf{dom}\, Y$ whose outcomes depend only on the present instance of X in a known way,

$$\mathbf{P}(y_t \mid x_0, \ldots, x_t) = \mathbf{P}(y_t \mid x_t).$$

The set Y is called the *observation space* of the hidden Markov model. For example, in weather forecasting, the weather cannot be directly measured; in reality, the weather is estimated based on the results from a series of sensors — temperature, pressure, wind velocity, etc. Figure 2 shows a hidden Markov model represented in a graph diagram.

We first consider a standard hidden Markov model where both the state space X and observation space Y are discrete and finite with $|X| = m$ and $|Y| = n$, respectively. To uniquely determine a hidden Markov model, the latent Markov chain can be well defined with the initial state distribution ρ and the transition matrix P. Besides, to describe the relationship between each latent state $x \in X$ and observation $y \in Y$, let B be a matrix with each entry

$$B_{ij} = \mathbf{P}(y_j \mid x_i),$$

for all $x_i \in X$, $y_i \in Y$, representing the probability of observing y_j under state x_i. The matrix B is called the *emission matrix* of the hidden Markov model. Similar to the transition matrix P, the emission matrix B has the following property:

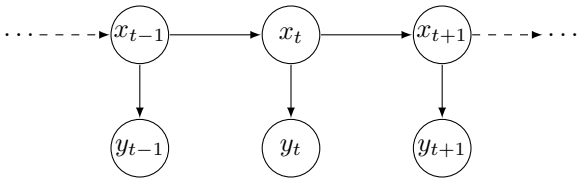

Figure 2 Graphical model representing a hidden Markov model. The top variables represent the hidden states and the nodes on the bottom are the observations.

- $B \in \mathbf{R}^{m \times n}$.
- $B_{ij} \geq 0$ for all $i = 1, \ldots, m$, $j = 1, \ldots, n$.
- $\mathbf{1}^T B_{i:} = 1$ for all $i = 1, \ldots, m$.

Thus the parameter set of a hidden Markov model can be denoted as a 3-dimensional set

$$\Theta = \{\rho, P, B\}.$$

Given a hidden Markov model representation of a certain domain, there are three basic questions that are of interest in most applications.

1. *Inference.* Given a model, estimate the probability of a sequence of observations.
2. *Decoding.* Given a model and a particular observation sequence, estimate the most probable state sequence that produced the observations.
3. *Parameter learning.* Given some sequences of observations, estimate the parameters of the model.

2.1 Inference

The inference problem of a hidden Markov model consists in determining the posterior probability of observing some sequence $\varphi = \{y_0, \ldots, y_N\}$, given the model parameters $\Theta = \{\rho, P, B\}$, that is, estimating the conditional probability $\mathbf{P}(\varphi \mid \Theta)$.

2.1.1 Direct method

Note that a sequence of observations $\varphi = \{y_0, \ldots, y_N\}$ can be generated by different state sequences $\zeta = \{x_0, \ldots, x_N\}$. Thus, to calculate the posterior probability of a given observation sequence, we can estimate the probability for a certain state sequence, and then add together the estimations for all the possible state sequences, resulting in

$$\mathbf{P}(\varphi \mid \Theta) = \sum_{\zeta} \mathbf{P}(\varphi, \zeta \mid \Theta).$$

Given a possible state sequence $\zeta = \{x_0, \ldots, x_N\}$, the probability $\mathbf{P}(\varphi, \zeta \mid \Theta)$ can be obtained by first calculating the probability of generating the specific state sequence, and then multiplying to the probability of observing corresponding observations from the state sequence, *i.e.*,

$$\mathbf{P}(\varphi, \zeta \mid \Theta) = \mathbf{P}(x_0 \mid \rho)\, \mathbf{P}(y_0 \mid x_0, B) \prod_{t=0}^{N-1} \mathbf{P}(x_{t+1} \mid x_t, P)\, \mathbf{P}(y_{t+1} \mid x_{t+1}, B),$$

where the probabilities on the right hand side are respective entries of the vector ρ and matrices P and B. Put together, the direct method estimates the posterior probability of observing some sequence $\varphi = \{y_0, \ldots, y_N\}$ according to

$$\mathbf{P}(\varphi \mid \Theta) = \sum_{x_0, \ldots, x_N} \mathbf{P}(x_0 \mid \rho)\, \mathbf{P}(y_0 \mid x_0, B) \prod_{t=0}^{N-1} \mathbf{P}(x_{t+1} \mid x_t, P)\, \mathbf{P}(y_{t+1} \mid x_{t+1}, B).$$

For a model with m states and an observation length of N, the direct method requires a number of operations in the order of $2Nm^N$ (or simply m^N) to solve the inference problem. This can be less practical when the length of observations is relatively large.

2.1.2 Iterative method

The basic idea of the iterative method, also known as the *forward algorithm*, is to estimate the probabilities of the observations per time step. That is, starting from $t = 0$, calculate the posterior probability of a partial sequence of observations until time t,

$$\mathbf{P}(\varphi_{0:t} \mid \Theta) = \mathbf{P}(y_0, \ldots, y_t \mid \Theta),$$

and based on this partial result, calculate it for time $t + 1$, until the end of the observation sequence where $t = N$.

First we introduce an auxiliary variable $\alpha_t \in \mathbf{R}_+^m$, and each of its entries $\alpha_t(i)$ is defined as

$$\alpha_t(i) = \mathbf{P}(y_0, \ldots, y_t, X_t = x_i \mid \Theta), \quad i = 1, \ldots, m.$$

The auxiliary variable α_t is called the *forward probability*, representing the posterior probability of observing the partial observation sequence up until time t, and the state under which the tth observation was generated is x_i. The forward probability can be calculated recursively with

$$\alpha_t(i) = \begin{cases} \rho_i\, \mathbf{P}(y_0 \mid X_0 = x_i, B), & t = 0 \\ \alpha_{t-1}^T P_{:i}\, \mathbf{P}(y_t \mid X_t = x_i, B), & t > 0, \end{cases}$$

for all $i = 1, \ldots, m$. Then the posterior probability of observing the whole sequence $\varphi = \{y_0, \ldots, y_N\}$ given model parameters Θ can be obtained by summing up all the entries of the forward probability evaluated at time $t = N$,

$$\mathbf{P}(\varphi \mid \Theta) = \mathbf{1}^T \alpha_N.$$

The pseudocode of the forward algorithm is shown in algorithm 1.

Algorithm 1 FORWARD ALGORITHM.

given hidden Markov model parameters Θ, observation sequence φ.

for $i = 1, \ldots, m$:

$\alpha_0(i) \coloneqq \rho_i \, \mathbf{P}(y_0 \mid X_0 = x_i, B)$.

for $t = 1, \ldots, N$, $i = 1, \ldots, m$:

$\alpha_t(i) \coloneqq \alpha_{t-1}^T P_{:i} \, \mathbf{P}(y_t \mid X_t = x_i, B)$.

return $\mathbf{P}(\varphi \mid \Theta) = \mathbf{1}^T \alpha_N$.

We now analyze the time complexity of the iterative method. Each iteration requires approximately m multiplications and m additions, so for the N iterations, the number of floating point operations is in the order of Nm^2, or simply m^2. Thus, the time complexity is reduced from exponential in N for the direct method to quadratic in m for the iterative method.

2.2 Decoding

Given a sequence of observations $\varphi = \{y_0, \ldots, y_N\}$ and model parameters Θ, the decoding problem of hidden Markov models can be interpreted in two ways:

- *Optimal state prediction.* Finding the most probable state x_t^* at time t.
- *Optimal sequence prediction.* Finding the most probable state sequence $\zeta^* = \{x_0^*, \ldots, x_N^*\}$ that generated observation sequence.

2.2.1 Optimal state prediction

Similar to the forward probability α_t, let $\beta_t \in \mathbf{R}_+^m$ be the *backward probability*, where each of its entry $\beta_t(i)$ is defined as

$$\beta_t(i) = \mathbf{P}(y_{t+1}, \ldots, y_N \mid X_t = x_i, \Theta), \quad i = 1, \ldots, m.$$

The backward probability β_t represents the posterior probability of observing the partial observation sequence after time t until the end of the sequence given that the state at that time is x_i. By defining the backward probability of the last observation ($t = N$) to be equal to 1, the others can be calculated recursively as

$$\beta_t(i) = \begin{cases} \sum_{j=1}^m \beta_{t+1}(j) P_{ij} \, \mathbf{P}(y_{t+1} \mid X_{t+1} = x_j, B), & t < N \\ 1, & t = N, \end{cases}$$

for all $i = 1, \ldots, m$. Figure 3 illustrates the computation of forward probability α_t and backward probability β_t across the graph diagram of a hidden Markov model. Note that by combining the forward probability and backward probability, we can

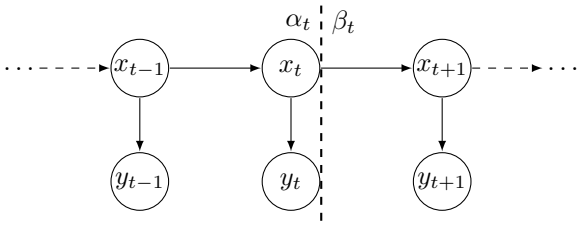

Figure 3 Computation of forward probability α_t and backward probability β_t of a hidden Markov model. Vertical dashed line indicates the separation point of the two probabilities at time t.

obtain the objective of the inference problem from §2.1, $\mathbf{P}(\varphi \mid \Theta)$, at any time t along the observation sequence, according to

$$\mathbf{P}(\varphi \mid \Theta) = \alpha_t^T \beta_t,$$

for all $t = 0, \ldots, N$.

Now we define another auxiliary variable $\gamma_t \in \mathbf{R}_+^m$, which represents the conditional probability of being in state x_i given the whole observation sequence:

$$\gamma_t(i) = \mathbf{P}(X_t = x_i \mid \varphi, \Theta), \quad i = 1, \ldots, m.$$

According to Bayes' rule, the vector γ_t can be written in terms of α_t and β_t as:

$$\gamma_t(i) = \frac{\mathbf{P}(X_t = x_i, \varphi \mid \Theta)}{\mathbf{P}(\varphi \mid \Theta)} = \frac{\alpha_t(i)\beta_t(i)}{\alpha_t^T \beta_t},$$

for all $i = 1, \ldots, m$. Then the most probable state x_t^* at time t given the observation sequence φ and parameters Θ of a hidden Markov model can be obtained by

$$x_t^* = \operatorname*{argmax}_{x_i} \gamma_t(i).$$

2.2.2 Optimal sequence prediction

The optimal sequence prediction problem of a hidden Markov model consists in finding a sequence of states $\zeta^* = \{x_0^*, \ldots, x_N^*\}$ such that

$$x_0^*, \ldots, x_N^* = \operatorname*{argmax}_{x_0, \ldots, x_N} \mathbf{P}(x_0, \ldots, x_N \mid \varphi, \Theta),$$

given the observation sequence $\varphi = \{y_0, \ldots, y_N\}$ and model parameters Θ. This problem can be solved approximately by a simple concatenation of the optimal states at each time t, *i.e.*,

$$\tilde{\zeta}^* = \left\{ \operatorname*{argmax}_{x_i} \gamma_t(i) \;\middle|\; t = 0, \ldots, N \right\}. \tag{2.1}$$

However, (2.1) is not guaranteed to provide a global optimal result of the state sequence since it does not consider the transition between states.

To obtain the exact solution of the optimal state sequence prediction problem, note that

$$\mathbf{P}(\zeta \mid \varphi, \Theta) = \frac{\mathbf{P}(\zeta, \varphi \mid \Theta)}{\mathbf{P}(\varphi \mid \Theta)} \propto \mathbf{P}(\zeta, \varphi \mid \Theta),$$

thus maximizing the posterior probability $\mathbf{P}(\zeta \mid \varphi, \Theta)$ over ζ is equivalent to maximizing the joint probability $\mathbf{P}(\zeta, \varphi \mid \Theta)$ of the state and observation sequence over ζ. This problem can be solved with the *Viterbi algorithm*. Let $\delta_t \in \mathbf{R}_+^m$ be the maximum value of the joint probability of a subsequence of states until time $t-1$, and observations until time t, with the state at time t is x_i,

$$\delta_t(i) = \max_{x_0,\ldots,x_{t-1}} \mathbf{P}(x_0, \ldots, x_{t-1}, X_t = x_i, y_0, \ldots, y_t \mid \Theta), \quad i = 1, \ldots, m.$$

This probability δ_t can be interpreted as the probability of being in state x_i at time t given that the state subsequence up until $t-1$ is optimal with respect to the partial observation sequence $\{y_0, \ldots, y_{t-1}\}$. The probability δ_t can also be obtained recursively as

$$\delta_t(i) = \max_{j=1,\ldots,m} (\delta_{t-1}(j) P_{ji})\, \mathbf{P}(y_t \mid X_t = x_i, B),$$

for all $i = 1, \ldots, m$. Let $\psi_t \in \mathbf{Z}_{++}^m$ store the index j of the previous state x_j at time $t-1$ that gives the maximum probability $\delta_{t-1}(j)P_{ji}$, for each state i at time t, *i.e.*,

$$\psi_t(i) = \operatorname*{argmax}_{j=1,\ldots,m} \delta_{t-1}(j) P_{ji}, \quad i = 1, \ldots, m,$$

which is used to reconstruct the state sequence by backtracking from the last state. Let p^* be the probability of obtaining the given observation sequence under the optimal state sequence $\zeta^* = \{x_0^*, \ldots, x_N^*\}$, the complete procedure of the Viterbi algorithm is shown in algorithm 2.

Algorithm 2 VITERBI ALGORITHM

given hidden Markov model parameters Θ, observation sequence φ.

1. *Initialization.* **for** $i = 1, \ldots, m$:
   $\delta_0(i) := \rho_i\, \mathbf{P}(y_0 \mid X_0 = x_i, B)$.
2. *Recursion.* **for** $t = 1, \ldots, N$, $i = 1, \ldots, m$:
   $\delta_t(i) := \max_{j=1,\ldots,m} (\delta_{t-1}(j) P_{ji})\, \mathbf{P}(y_t \mid X_t = x_i, B)$.
   $\psi_t(i) := \operatorname{argmax}_{j=1,\ldots,m} \delta_{t-1}(j) P_{ji}$.
3. *Termination.*
   $p^* := \max_{i=1,\ldots,m} \delta_N(i)$.
   $i_N^* := \operatorname{argmax}_{i=1,\ldots,m} \delta_N(i)$.
   $x_N^* := x_{i_N^*}$.
4. *Backtracking.* **for** $t = N, \ldots, 1$:
   $i_{t-1}^* := \psi_t(i_t^*)$.
   $x_{t-1}^* := x_{i_{t-1}^*}$.

2.3 Parameter learning

The *expectation-maximization algorithm* (EM) is commonly used in estimating the parameters of models involving latent variables. The idea is to start with some initial parameters for the model, which can be initialized randomly or based on some domain knowledge. In the E-step, some likelihood function with respect to the current model parameters is calculated. Then in the M-step, these parameters are optimized to maximizing an MLE objective. This cycle is repeated until convergence; *e.g.*, until the difference between the parameters for the model from one step to the next is below a certain threshold.

To learn the parameters of a hidden Markov model, suppose we are given a set of observation sequences $\mathcal{D} = \{\varphi_1, \varphi_2, \ldots\}$ from that hidden Markov model, where $\varphi = \{y_0, \ldots, y_N\}$ for all $\varphi \in \mathcal{D}$. First we should note that the cardinalities of the state space and observation space of the model have to be known or previously defined. Let $\Theta = \{\rho, P, B\}$ be the set of estimated parameters of the model from the previous EM-iteration, and $\Theta^+ = \{\rho^+, P^+, B^+\}$ be the new set of parameters to be updated. For each EM-iteration, we first calculate the current estimation of hidden state sequence $\zeta = \{x_0, \ldots, x_N\}$ with parameters Θ. Then similar to learning the parameters of a Markov chain (§1.1), the parameters ρ^+, P^+, and B^+ of a hidden Markov model can be updated according to

$$\rho_i^+ = \mathop{\mathbf{E}}_{\varphi \sim \mathcal{D}} \mathbf{P}(X_0 = x_i \mid \varphi, \Theta), \quad i = 1, \ldots, m, \tag{2.2}$$

$$P_{ij}^+ = \frac{\mathbf{E}_{\varphi \sim \mathcal{D}, t}\, \mathbf{P}(X_t = x_i, X_{t+1} = x_j \mid \varphi, \Theta)}{\mathbf{E}_{\varphi \sim \mathcal{D}, t}\, \mathbf{P}(X_t = x_i \mid \varphi, \Theta)}, \quad i = 1, \ldots, m, \quad j = 1, \ldots, m, \tag{2.3}$$

and

$$B_{ij}^+ = \frac{\mathbf{E}_{\varphi \sim \mathcal{D}, t}\, \mathbf{P}(X_t = x_i, Y_t = y_j \mid \varphi, \Theta)}{\mathbf{E}_{\varphi \sim \mathcal{D}, t}\, \mathbf{P}(X_t = x_i \mid \varphi, \Theta)}, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n. \tag{2.4}$$

Again, these updating equations align with our intuition, but can also be derived analytically. To obtain the probability term on the numerator of (2.3), let $\xi_t \in \mathbf{R}_+^{m \times m}$ represent the probability of transitioning from state x_i at time t to state x_j at time $t+1$ given an observation sequence φ:

$$\begin{aligned} \xi_t(i,j) &= \mathbf{P}(X_t = x_i, X_{t+1} = x_j \mid \varphi, \Theta) \\ &= \frac{\mathbf{P}(X_t = x_i, X_{t+1} = x_j, \varphi \mid \Theta)}{\mathbf{P}(\varphi \mid \Theta)}, \quad i = 1, \ldots, m, \quad j = 1, \ldots, m. \end{aligned}$$

The denominator $\mathbf{P}(\varphi \mid \Theta)$ is just a normalization factor, which can be calculated from:

$$\mathbf{P}(\varphi \mid \Theta) = \sum_{i=1}^{m} \sum_{j=1}^{m} \mathbf{P}(X_t = x_i, X_{t+1} = x_j, \varphi \mid \Theta).$$

The auxiliary variable ξ_t can be written in terms of the forward probability α_t and the backward probability β_t:

$$\xi_t(i,j) = \frac{\alpha_t(i) P(i,j)\, \mathbf{P}(y_{t+1} \mid X_{t+1} = x_j, B) \beta_{t+1}(j)}{\sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_t(i) P(i,j)\, \mathbf{P}(y_{t+1} \mid X_{t+1} = x_j, B) \beta_{t+1}(j)},$$

for all $i = 1, \ldots, m$, $j = 1, \ldots, m$. Clearly, the probability γ_t can also be written in terms of ξ_t:

$$\gamma_t(i) = \sum_{j=1}^{m} \xi_t(i, j),$$

for all $i = 1, \ldots, m$. As a result, the equations (2.2) to (2.4) can be represented compactly with the previously defined auxiliary variables as

$$\rho_i^+ = \mathop{\mathbf{E}}_{\varphi \sim \mathcal{D}} \gamma_0(i), \quad i = 1, \ldots, m, \tag{2.5}$$

$$P_{ij}^+ = \frac{\mathbf{E}_{\varphi \sim \mathcal{D}, t}\, \xi_t(i, j)}{\mathbf{E}_{\varphi \sim \mathcal{D}, t}\, \gamma_t(i)}, \quad i = 1, \ldots, m, \quad j = 1, \ldots, m, \tag{2.6}$$

and

$$B_{ij}^+ = \frac{\mathbf{E}_{\varphi \sim \mathcal{D}, t}(\gamma_t(i) I_j(y_t))}{\mathbf{E}_{\varphi \sim \mathcal{D}, t}\, \gamma_t(i)}, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n, \tag{2.7}$$

where $I_j(y)$ is an indicator function with $I_j(y) = 1$ if the index of y in observation space Y is equal to j, and 0 otherwise. Note that similar to (1.3), the expectations in (2.3) and (2.6) over time t are estimated across $t = 0, \ldots, N - 1$. The update equations (2.5) to (2.7) are called the *Baum-Welch algorithm*. The pseudocode for the whole procedure is shown in algorithm 3.

Algorithm 3 BAUM-WELCH ALGORITHM.

given the estimated hidden Markov model parameters Θ from previous EM-iteration, the set of observation sequences $\mathcal{D}$.

1. Estimate the initial state distribution ρ^+ according to (2.5).
2. Estimate the transition matrix P^+ according to (2.6).
3. Estimate the emission matrix B^+ according to (2.7).
4. Estimate the emission matrix B^+ according to (2.7).

return the updated set of parameters $\Theta^+ = \{\rho^+, P^+, B^+\}$.

As the last point, it should be noted that while the EM algorithm is guaranteed to converge to a local optimum, it does not ensure a global optimal solution. The convergence point of EM depends on its initial conditions.

2.4 Continuous observation space

In many applications the observation space is continuous. In this case, an alternative to discretization is to work directly with the continuous features, assuming them to be Gaussian distributed. This assumption leads to the *Gaussian hidden Markov models*, where the initial state ditribution vector and the transition matrix are as those in standard hidden Markov models, but the map from each state to the observation space are modeled as a Gaussian distribution. Suppose the domain of

the observation space $\mathbf{dom}\, Y = \mathbf{R}$, then the emission map of the hidden Markov model is given by the Gaussian density function $\mathcal{N}(\mu_i, \sigma_i^2)$:

$$p(y \mid X = x_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{(y-\mu_i)^2}{2\sigma_i^2}\right), \quad i = 1, \ldots, m. \tag{2.8}$$

This idea can be further extended to vector observations, for example $y \in \mathbf{R}^n$ for all $y \in Y$, by considering the following joint Gaussian density function $\mathcal{N}(\mu_i, \Sigma_i)$ as an alternative of (2.8):

$$p(y \mid X = x_i) = \frac{1}{\sqrt{(2\pi)^n \det \Sigma_i}} \exp\left(-\frac{1}{2}(y-\mu_i)^T \Sigma_i^{-1}(y-\mu_i)\right), \quad i = 1, \ldots, m,$$

with the vector of distribution mean $\mu_i \in \mathbf{R}^n$ and the covariance matrix $\Sigma_i \in \mathbf{S}_{++}^n$. Moreover, sometimes the observation space can not be described by a single Gaussian distribution, in this case we can use a *Gaussian mixture model*, which consists of multiple Gaussian distributions that are combined to represent the desired distribution.

The algorithms for solving the three basic problems (inference, decoding, and parameter learning) of a Gaussian hidden Markov model are essentially the same as those for standard hidden Markov models, just considering that the observations are modeled as a Gaussian distribution or a Gaussian mixture model.

Bibliography

A general introduction to Markov chains is provided in the article [KS69]. The convergence property of Markov chains is a corollary of the *Perron-Frobenius theorem*, which is discussed in detail in [HJ12].

The Viterbi algorithm was initially introduced by Andrew Viterbi in decoding convolutional codes, which is widely used in communications. The original paper [Vit67] contains some details about the underlying idea of dynamical programming and related mathematical derivations.

The convergence analysis about the expectation-maximization algorithm can be found in [LR19].

About some other variants of hidden Markov models and their applications, one can refer to [ASMP11].

References

[ASMP11] H. H. Avilés-Arriaga, L. E. Sucar-Succar, C. E. Mendoza-Durán, and L. A. Pineda-Cortés. A comparison of dynamic naive Bayesian classifiers and hidden Markov models for gesture recognition. *Journal of Applied Research and Technology*, 9(1):81–102, 2011.

[HJ12] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 2012.

[KS69] J. G. Kemeny and J. L. Snell. *Finite Markov chains*, volume 26. van Nostrand Princeton, 1969.

[LR19] R. J. A. Little and D. B. Rubin. *Statistical Analysis with Missing Data*, volume 793. John Wiley & Sons, 2019.

[Vit67] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967.