Lecture Pattern Analysis

Part 19: HMMs Algorithms 1 and 2

Christian Riess
IT Security Infrastructures Lab, Friedrich-Alexander-Universität Erlangen-Nürnberg
June 17, 2021

Algorithm 1: How Well Matches an Input Sequence the HMM?

- Calculate $p(\mathbf{x}_1, \ldots, \mathbf{x}_T | \lambda)$ by marginalizing over all state sequences $z_1, \ldots, z_T$
- Naive marginalization is exponential in the sequence length,

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_T) = \sum_{z_1=S_1}^{S_N} \sum_{z_2=S_1}^{S_N} \ldots \sum_{z_T=S_1}^{S_N} \pi_{z_1} b_{z_1}(\mathbf{x}_1) a_{z_1 z_2} b_{z_2}(\mathbf{x}_2) \ldots a_{z_{T-1} z_T} b_{z_T}(\mathbf{x}_T) \quad , \tag{1}$$

  with a computational cost of $O(N^T)$
- However, the chain structure of dependencies admits an efficient dynamic programming solution

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_T) = \sum_{z_1=S_1}^{S_N} \pi_{z_1} b_{z_1}(\mathbf{x}_1) \sum_{z_2=S_1}^{S_N} a_{z_1 z_2} b_{z_2}(\mathbf{x}_2) \ldots \sum_{z_T=S_1}^{S_N} a_{z_{T-1} z_T} b_{z_T}(\mathbf{x}_T) \tag{2}$$

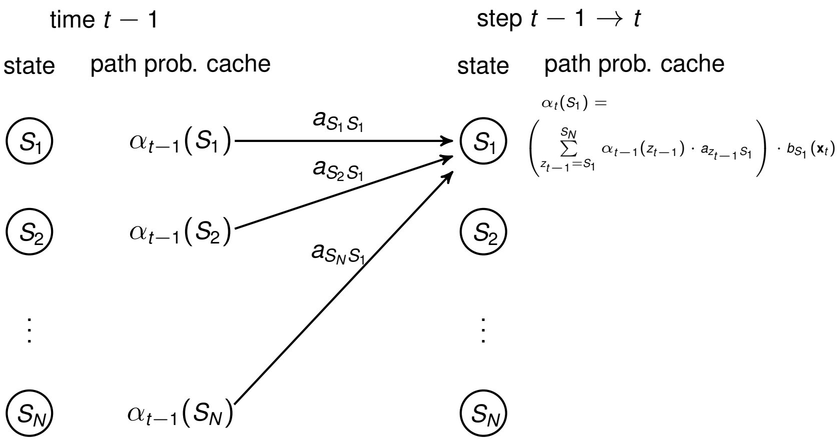
  with caching of the N accumulated state probabilities at time t.
  Each state transition requires $O(N^2)$ evaluations, the total cost is $O(N^2 T)$

The Forward Algorithm and the Backward Algorithm

- The forward algorithm directly implements Eqn. 2:
  1. Initialization (here and below, S_i implicitly creates a list of N entries):
     $$t = 1: \qquad \alpha_1(z_1 = S_i) = \pi_{z_1} b_{z_1}(\mathbf{x}_1)$$
  2. Iteration:
     $$2 \leq t \leq T: \alpha_t(z_t = S_i) = \left(\sum_{z_{t-1}=S_1}^{S_N} \alpha_{t-1}(z_{t-1}) \cdot a_{z_{t-1} z_t} \right) b_{z_t}(\mathbf{x}_t)$$
  3. Summation over all end points:
     $$t = T: \qquad p(\mathbf{x}_1, \ldots, \mathbf{x}_T) = \sum_{z_T=S_1}^{S_N} \alpha_T(z_t)$$
- Training requires the (almost identical) backward alg., starting at time T:
  1. Initialization:
     $$t = T: \qquad \beta_T(z_1 = S_i) = 1$$
  2. Iteration: $1 \leq t \leq T$: $\beta_t(z_t = S_i) = \sum_{z_{t+1}=S_1}^{S_N} \beta_{t+1}(z_{t+1}) b_{z_{t+1}}(\mathbf{x}_{t+1}) a_{z_t z_{t+1}}$
- The backward iteration does not include $\mathbf{x}_1$, which is OK for its training use

Forward Algorithm: State-Time Diagram

- This diagram over states (y-axis) and time (x-axis) illustrates the caching:



Algorithm 2: What is the Most Likely State Sequence?

- Finding $\underset{z_t = S_j}{\operatorname{argmax}}\, p(\mathbf{x}_1, \ldots, \mathbf{x}_T, z_1, \ldots, z_T | \lambda)$ is almost identical to forward alg.
  - Variable $\delta_t(z_i)$ caches the likelihood of the most likely path to z_i within time $1..t$
  - Variable $\psi_t(z_i)$ caches the predecessor state to reconstruct the path
- Viterbi algorithm:
  1. Initialization:
     $$t = 1: \quad \delta_1(z_1 = S_i) = \pi_{z_1} b_{z_1}(\mathbf{x}_1)$$
     $$\psi_1(z_1 = S_i) = 0$$
  2. Iteration:
     $$1 \leq t \leq T: \delta_t(z_t = S_i) = \max_{S_1 \leq z_t \leq S_N} \left(\delta_{t-1}(z_{t-1}) \cdot a_{z_{t-1} z_t}\right) b_{z_t}(\mathbf{x}_t)$$
     $$\psi_t(z_t) = \underset{S_1 \leq z_t \leq S_N}{\operatorname{argmax}}\ \delta_{t-1}(z_{t-1}) \cdot a_{z_{t-1} z_t}$$
  3. Maximum over all end points:
     $$t = T: \quad p^* = \max_{S_1 \leq z_T \leq S_N} \delta_T(z_t)$$
     $$z_T^* = \underset{S_1 \leq z_T \leq S_N}{\operatorname{argmax}}\ \delta_T(z_T)$$
  4. Path reconstruction: $z_t^* = \psi_{t+1}(z_{t+1}^*)$

Remarks

- The computational complexity of the Viterbi algorithm is also $O(N^2 T)$
- Interpretation of the caches $\alpha_t(S_i)$, $\beta_t(S_i)$, $\delta_t(S_i)$:
  - $\alpha_t(S_i)$ is the sum of probabilities of all paths from time 1 to time t that are at time t in state S_i
  - $\beta_t(S_i)$ is the sum of probabilities of all paths from time T to time t that are at time t in state S_i
  - $\delta_t(S_i)$ is the probability of the single, most likely path from time 1 to time t that is at time t in state S_i

  We will use these interpretations to explain the training procedure
- The Viterbi algorithm has also many other applications, e.g., as a decoder for partially corrupted digital watermarks (see lecture “Multimedia Security”)