A Tutorial on Hidden Markov Models

by Lawrence R. Rabiner
in Readings in speech recognition (1990)

Marcin Marszałek

Visual Geometry Group

16 February 2009

Figure: Andrey Markov

Signals and signal models

- Real-world processes produce signals, i.e., observable outputs
  - discrete (from a codebook) vs continous
  - stationary (with const. statistical properties) vs nonstationary
  - pure vs corrupted (by noise)
- Signal models provide basis for
  - signal analysis, e.g., simulation
  - signal processing, e.g., noise removal
  - signal recognition, e.g., identification
- Signal models can be
  - deterministic – exploit some known properties of a signal
  - statistical – characterize statistical properties of a signal
- Statistical signal models
  - Gaussian processes
  - Poisson processes
  - Markov processes
  - Hidden Markov processes

Signals and signal models

- Real-world processes produce signals, i.e., observable outputs
  - discrete (from a codebook) vs continous
  - stationary (with const. statistical properties) vs nonstationary
  - pure vs corrupted (by noise)

Assumption

Signal can be well characterized as a parametric random process, and the parameters of the stochastic process can be determined in a precise, well-defined manner

  - deterministic – exploit some known properties of a signal
  - statistical – characterize statistical properties of a signal
- Statistical signal models
  - Gaussian processes
  - Poisson processes
  - Markov processes
  - Hidden Markov processes

Discrete (observable) Markov model

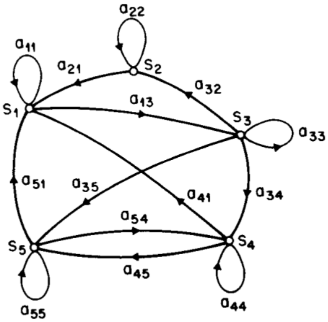


Figure: A Markov chain with 5 states and selected transitions

- N states: $S_1, S_2, ..., S_N$
- In each time instant $t = 1, 2, ..., T$ a system changes (makes a transition) to state q_t

Discrete (observable) Markov model

- For a special case of a first order Markov chain

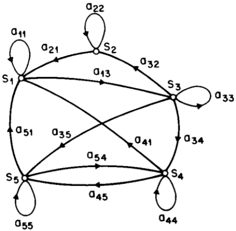
$$P(q_t = S_j | q_{t-1} = S_i, t_{t-2} = S_k, \ldots) = P(q_t = S_j | q_{t-1} = S_i)$$

- Furthermore we only assume processes where right-hand side is time independent – const. state transition probabilities

$$a_{ij} = P(q_t = S_j | q_{t-1} = S_i) \qquad 1 \leq i, j \leq N$$

where

$$a_{ij} \geq 0 \qquad \sum_{j=1}^{N} a_{ij} = 1$$



Discrete hidden Markov model (DHMM)

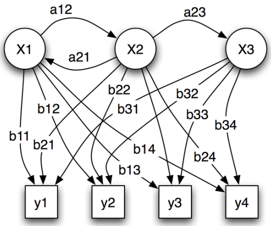
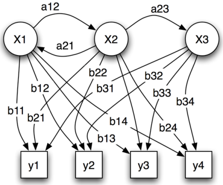


Figure: Discrete HMM with 3 states and 4 possible outputs

- An observation is a probabilistic function of a state, i.e., HMM is a doubly embedded stochastic process
- A DHMM is characterized by
  - N states S_j and M distinct observations v_k (alphabet size)
  - State transition probability distribution A
  - Observation symbol probability distribution B
  - Initial state distribution π

Discrete hidden Markov model (DHMM)

- We define the DHMM as $\lambda = (A, B, \pi)$
  - $A = \{a_{ij}\}$ $\quad a_{ij} = P(q_{t+1} = S_j | q_t = S_i)$ $\quad 1 \le i, j \le N$
  - $B = \{b_{ik}\}$ $\quad b_{ik} = P(O_t = v_k | q_t = S_i)$ $\quad 1 \le i \le N$, $1 \le k \le M$
  - $\pi = \{\pi_i\}$ $\quad \pi_i = P(q_1 = S_i)$ $\quad 1 \le i \le N$
- This allows to generate an observation seq. $O = O_1 O_2 \ldots O_T$
  1. Set $t = 1$, choose an initial state $q_1 = S_i$ according to the initial state distribution π
  2. Choose $O_t = v_k$ according to the symbol probability distribution in state S_i, i.e., b_{ik}
  3. Transit to a new state $q_{t+1} = S_j$ according to the state transition probability distibution for state S_i, i.e., a_{ij}
  4. Set $t = t + 1$, if $t < T$ then return to step 2



Three basic problems for HMMs

Evaluation Given the observation sequence $O = O_1 O_2 \ldots O_T$ and a model $\lambda = (A, B, \pi)$, how do we efficiently compute $P(O|\lambda)$, i.e., the probability of the observation sequence given the model

Recognition Given the observation sequence $O = O_1 O_2 \ldots O_T$ and a model $\lambda = (A, B, \pi)$, how do we choose a corresponding state sequence $Q = q_1 q_2 \ldots q_T$ which is optimal in some sense, i.e., best explains the observations

Training Given the observation sequence $O = O_1 O_2 \ldots O_T$, how do we adjust the model parameters $\lambda = (A, B, \pi)$ to maximize $P(O|\lambda)$

Brute force solution to the evaluation problem

- We need $P(O|\lambda)$, i.e., the probability of the observation sequence $O = O_1 O_2 \ldots O_T$ given the model λ
- So we can enumerate every possible state sequence $Q = q_1 q_2 \ldots q_T$
- For a sample sequence Q

$$P(O|Q,\lambda) = \prod_{t=1}^{T} P(O_t|q_t,\lambda) = \prod_{t=1}^{T} b_{q_t O_t}$$

- The probability of such a state sequence Q is

$$P(Q|\lambda) = P(q_1) \prod_{t=2}^{T} P(q_t|q_{t-1}) = \pi_{q_1} \prod_{t=2}^{T} a_{q_{t-1} q_t}$$

Brute force solution to the evaluation problem

- Therefore the joint probability

$$P(O,Q|\lambda) = P(Q|\lambda)P(O|Q,\lambda) = \pi_{q_1} \prod_{t=2}^{T} a_{q_{t-1}q_t} \prod_{t=1}^{T} b_{q_t O_t}$$

- By considering all possible state sequences

$$P(O|\lambda) = \sum_{Q} \pi_{q_1} b_{q_1 O_1} \prod_{t=2}^{T} a_{q_{t-1}q_t} b_{q_t O_t}$$

- Problem: order of $2TN^T$ calculations
  - N^T possible state sequences
  - about $2T$ calculations for each sequence

Forward procedure

- We define a forward variable $\alpha_j(t)$ as the probability of the partial observation seq. until time t, with state S_j at time t

$$\alpha_j(t) = P(O_1 O_2 \dots O_t, q_t = S_j | \lambda)$$

- This can be computed inductively

$$\alpha_j(1) = \pi_j b_{jO_1} \qquad\qquad 1 \leq j \leq N$$

$$\alpha_j(t+1) = \Big(\sum_{i=1}^{N} \alpha_i(t) a_{ij}\Big) b_{jO_{t+1}} \qquad 1 \leq t \leq T-1$$

- Then with $N^2 T$ operations:

$$P(O|\lambda) = \sum_{i=1}^{N} P(O, q_T = S_i | \lambda) = \sum_{i=1}^{N} \alpha_i(T)$$

Forward procedure

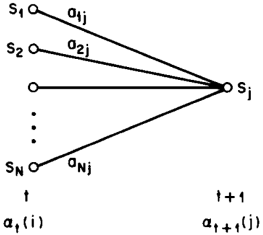


Figure: Operations for computing the forward variable $\alpha_j(t+1)$

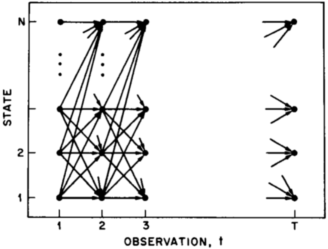
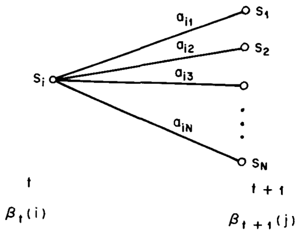


Figure: Computing $\alpha_j(t)$ in terms of a lattice

Backward procedure

Figure: Operations for computing the backward variable $\beta_i(t)$



- We define a backward variable $\beta_i(t)$ as the probability of the partial observation seq. after time t, given state S_i at time t

$$\beta_i(t) = P(O_{t+1}Ot+2...O_T|q_t = S_i, \lambda)$$

- This can be computed inductively as well

$$\beta_i(T) = 1 \qquad 1 \leq i \leq N$$

$$\beta_i(t-1) = \sum_{j=1}^{N} a_{ij} b_{jO_t} \beta_j(t) \qquad 2 \leq t \leq T$$

Uncovering the hidden state sequence

- Unlike for evaluation, there is no single "optimal" sequence
  - Choose states which are individually most likely (maximizes the number of correct states)
  - Find the single best state sequence (guarantees that the uncovered sequence is valid)
- The first choice means finding $\operatorname{argmax}_i \gamma_i(t)$ for each t, where

$$\gamma_i(t) = P(q_t = S_i | O, \lambda)$$

- In terms of forward and backward variables

$$\gamma_i(t) = \frac{P(O_1 \ldots O_t, q_t = S_i | \lambda) P(O_{t+1} \ldots O_T | q_t = S_i, \lambda)}{P(O|\lambda)}$$

$$\gamma_i(t) = \frac{\alpha_i(t)\beta_i(t)}{\sum_{j=1}^{N} \alpha_j(t)\beta_j(t)}$$

Viterbi algorithm

- Finding the best single sequence means computing $\text{argmax}_Q P(Q|O,\lambda)$, equivalent to $\text{argmax}_Q P(Q,O|\lambda)$
- The Viterbi algorithm (dynamic programming) defines $\delta_j(t)$, i.e., the highest probability of a single path of length t which accounts for the observations and ends in state S_j

$$\delta_j(t) = \max_{q_1,q_2,\ldots,q_{t-1}} P(q_1 q_2 \ldots q_t = j, O_1 O_2 \ldots O_t|\lambda)$$

- By induction

$$\begin{aligned} \delta_j(1) &= \pi_j b_{jO_1} & 1 \leq j \leq N \\ \delta_j(t+1) &= \left(\max_i \delta_i(t) a_{ij}\right) b_{jO_{t+1}} & 1 \leq t \leq T-1 \end{aligned}$$

- With backtracking (keeping the maximizing argument for each t and j) we find the optimal solution

Backtracking

Figure: Illustration of the backtracking procedure © G.W. Pulford

Estimation of HMM parameters

- There is no known way to analytically solve for the model which maximizes the probability of the observation sequence
- We can choose $\lambda = (A, B, \pi)$ which locally maximizes $P(O|\lambda)$
  - gradient techniques
  - Baum-Welch reestimation (equivalent to EM)
- We need to define $\xi_{ij}(t)$, i.e., the probability of being in state S_i at time t and in state S_j at time $t+1$

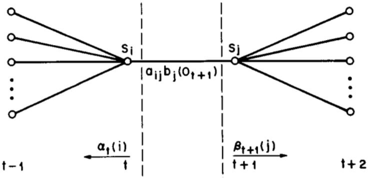
$$\xi_{ij}(t) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda)$$

$$\xi_{ij}(t) = \frac{\alpha_i(t) a_{ij} b_{jO_{t+1}} \beta_j(t+1)}{P(O|\lambda)} =$$

$$= \frac{\alpha_i(t) a_{ij} b_{jO_{t+1}} \beta_j(t+1)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_i(t) a_{ij} b_{jO_{t+1}} \beta_j(t+1)}$$

Estimation of HMM parameters

Figure: Operations for computing the $\xi_{ij}(t)$



- Recall that $\gamma_i(t)$ is a probability of state S_i at time t, hence

$$\gamma_i(t) = \sum_{j=1}^{N} \xi_{ij}(t)$$

- Now if we sum over the time index t
  - $\sum_{t=1}^{T-1} \gamma_i(t)$ = expected number of times that S_i is visited*
    = expected number of transitions from state S_i
  - $\sum_{t=1}^{T-1} \xi_{ij}(t)$ = expected number of transitions from S_i to S_j

Baum-Welch Reestimation

- Reestimation formulas

$$\bar{\pi}_i = \gamma_i(1) \qquad \bar{a_{ij}} = \frac{\sum_{t=1}^{T-1} \xi_{ij}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)} \qquad \bar{b_{jk}} = \frac{\sum_{O_t=v_k} \gamma_j(t)}{\sum_{t=1}^{T} \gamma_j(t)}$$

- Baum et al. proved that if current model is $\lambda = (A, B, \pi)$ and we use the above to compute $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$ then either
  - $\bar{\lambda} = \lambda$ – we are in a critical point of the likelihood function
  - $P(O|\bar{\lambda}) > P(O|\lambda)$ – model $\bar{\lambda}$ is more likely
- If we iteratively reestimate the parameters we obtain a maximum likelihood estimate of the HMM
- Unfortunately this finds a local maximum and the surface can be very complex

Non-ergodic HMMs

- Until now we have only considered ergodic (fully connected) HMMs
  - every state can be reached from any state in a finite number of steps

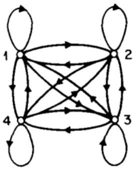


Figure: Ergodic HMM

- Left-right (Bakis) model good for speech recognition
  - as time increases the state index increases or stays the same
  - can be extended to parallel left-right models

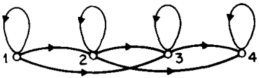


Figure: Left-right HMM

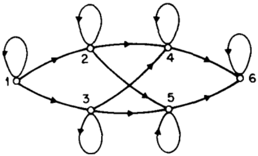


Figure: Parallel HMM

Gaussian HMM (GMMM)

- HMMs can be used with continous observation densities
- We can model such densities with Gaussian mixtures

$$b_{j\mathbf{O}} = \sum_{m=1}^{M} c_{jm}\mathcal{N}(\mathbf{O}, \boldsymbol{\mu}_{jm}, \mathbf{U}_{jm})$$

- Then the reestimation formulas are still simple

$$\gamma_t(j, k) = \left[\frac{\alpha_t(j)\, \beta_t(j)}{\sum_{j=1}^{N} \alpha_t(j)\, \beta_t(j)} \right] \left[\frac{c_{jk}\mathfrak{N}(\mathbf{O}_t, \boldsymbol{\mu}_{jk}, \mathbf{U}_{jk})}{\sum_{m=1}^{M} c_{jm}\mathfrak{N}(\mathbf{O}_t, \boldsymbol{\mu}_{jm}, \mathbf{U}_{jm})} \right]$$

$$\overline{c}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j, k)}{\sum_{t=1}^{T} \sum_{k=1}^{M} \gamma_t(j, k)}$$

$$\overline{\boldsymbol{\mu}}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j, k) \cdot \mathbf{O}_t}{\sum_{t=1}^{T} \gamma_t(j, k)}$$

$$\overline{\mathbf{U}}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j, k) \cdot (\mathbf{O}_t - \boldsymbol{\mu}_{jk})(\mathbf{O}_t - \boldsymbol{\mu}_{jk})'}{\sum_{t=1}^{T} \gamma_t(j, k)}$$

More fun

- Autoregressive HMMs
- State Duration Density HMMs
- Discriminatively trained HMMs
  - maximum mutual information instead of maximum likelihood
- HMMs in a similarity measure
- Conditional Random Fields can loosely be understood as a generalization of an HMMs
  - constant transition probabilities replaced with arbitrary functions that vary across the positions in the sequence of hidden states

Figure: Random Oxford fields © R. Tourtelot