Hidden Markov Models

Gaurav Sharma

University of Rochester

Motivation

- Recall our two basic probabilistic models
  - IID models
    - Appropriate for modeling phenomena without dependency/memory
  - Markov models
    - Appropriate for modeling phenomena with dependency/memory
    - Future independent of past given present
    - Factorization of joint PMF

$$p(x_1, x_2, \ldots x_n) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_2) \ldots p(x_n \mid x_{n-1})$$

- Model appears rather specific
- but is quite general. In particular, "higher-order" Markov models are also Markov models. Recall assignment on Markov models.

Finite-state Discrete-Time Time-Invariant Markov Chains

- Recall Characterization
  - States $1, 2, \ldots L$
  - State transition probability matrix $\mathsf{P} = [p_{ij}] = [p_{i \to j}]$
  - Initial state probability $\boldsymbol{\pi} = [\pi_1, \pi_2, \ldots \pi_L]$
  - State probabilities at time n

$$\mathsf{p}^{(n)} = \mathsf{p}^{(n-1)}\mathsf{P} = \boldsymbol{\pi}\mathsf{P}^{(n-1)}$$

Latent/Hidden Variables

- All quantities of interest are often not observable
- Though the "world" of interest may be undergoing Markovian evolution, full state is often not observed directly
- Power of the model for inference is often improved by allowing for the observations to be indirectly dependent on states
  - Stochastic functions of states
    - Hidden/latent variables to model relation between underlying state and observations
- For IID models, we saw the power of this methodology with the Expectation Maximization (EM) algorithm
- Hidden Markov Models (HMMs) offer analogous generalization/extension of Markov models
  - Are extremely powerful tools for probabilistic inference
  - Vast number of applications
    - Fields: Machine learning, signal processing, statistics, computer vision, ...
    - Applications: Speech and natural language understanding, communications and error control coding, bioinformatics, ...

State and Trellis Diagrams

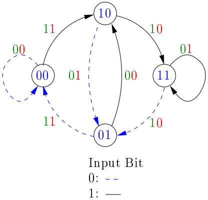


Figure: State Transition Diagram

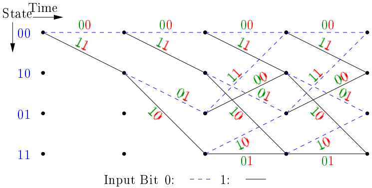


Figure: Trellis Diagram

HMMs: Four "Coin" Motivating Example

- Communication with a friend who exists in two states: office (0) and home (1)
  - Excellent connectivity at work and poor connectivity at home
  - Errors in communication determined by state
    - Few errors when in office 0 and many errors when at home 1
- You observe only errors/no-errors
  - Friend's state is not known to you
  - Friend transitions between states 0 and 1
    - Would an iid model be appropriate here?

HMMs: Four “Coin” Example: Abstraction

- Communicating bits over channel with two states
  - Good state (0) low bit error probability p_g, bad state (1) high bit error probability p_b
  - State has memory (Markovian evolution) transition probability parameters a, b
    - What does persistence of state imply for a and b?

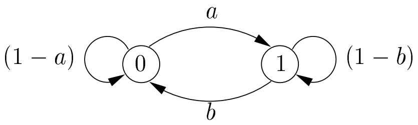


Figure: State Transition Diagram

- Convention used: “emission” occurs after “transition” to state
- You do not observe states directly only errors/no errors
  - a string of 0’s and 1’s where the 0’s correspond to no errors and 1’s correspond to errors
    - XORing with transmitted data gives the received bits
  - Q: How could errors be observed? (Assume all 0’s sent)

HMMs: Four "Coin" Example: Abstraction

- Communicating bits over channel with two states

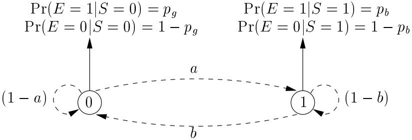


Figure: State Transition Diagram

- Why 4 "coin" example?
- "Doubly" stochastic process
  - Stochastic state process
  - Stochastic output process from state

Three Alternative HMM Representations

- 1. Random function of a Markov process
- 2. Deterministic function of a Markov process
- 3. Joint Markov process (with unobserved state) plus transition probabilities depend only on state
- Three models are conceptually equivalent, any HMM can be represented in either of three representations.
- Three models differ from a computational perspective
  - Representation size (RS): RS(2) $\geq$ RS(1) $\geq$ RS(3)
  - Representation (2) computationally least economical (larger state size) despite appearing conceptually elegant
  - Representation (1) is most commonly utilized

HMM as Random Function of a Markov Process

- Doubly Stochastic Process
  - Unobserved Markov process: $z_1, z_2, \dots z_N$
    - Also called state sequence/process
    - State possibilities: $\mathrm{S} = \{S_1, S_2, \dots, S_L\}$, L = number of states
    - State transition probability matrix: $\mathrm{P} = \{p_{ij}\}$
    - Initial state probabilities: $\pi_i = p(z_0 = S_i), \boldsymbol{\pi} = [\pi_i]$
  - Observed HMM output sequence $x_1, x_2, \dots x_N$
    - Output possibilities: $\mathrm{v} = \{v_1, v_2, \dots, v_M\}$
    - State dependent emission probabilities: $G = \{g_i(v_k)\}$
  - Model parameters $\boldsymbol{\theta} = (\mathrm{P}, \mathrm{G}, \boldsymbol{\pi})$

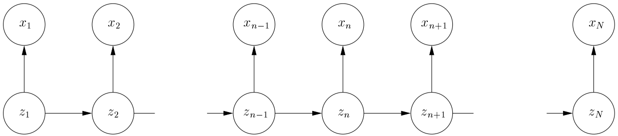


Figure: Illustration of dependency structure for a HMM. The sequence of observations $x_1, x_2, \dots, x_N$ forms a hidden Markov model, where the unobserved state sequence $z_1, z_2, \dots, z_N$ forms a Markov process.

Example HMM as Doubly Stochastic Process

- Four "coins" example:
  - Unobserved Markov state process: $z_1, z_2, \ldots z_N \equiv$ sequence of coin identities
    - State possibilities: $\mathrm{S} = \{0, 1\}$. Number of states $L = |S| = 2$.
    - State transition probability matrix: $\mathrm{P} \equiv \{a, b\}$
    - Initial state probabilities: $\pi_z = p(z_0 = z), \boldsymbol{\pi} = [\pi_0, \pi_1]$
  - Observations: $x_1, x_2, \ldots x_N \equiv$ sequence of coin outcomes
    - State dependent emission probabilities:
    - $g_0(0) = (1 - p_g), g_0(1) = p_g$
    - $g_1(0) = (1 - p_b), g_1(1) = p_b$
- Model parameters $\boldsymbol{\theta} = (a, b, p_g, p_b, \boldsymbol{\pi})$

HMM Example: Inference and Estimation

- Three basic problems
  - Likelihood evaluation for a given observation sequence
    - Given an observed sequence (of errors) $\mathrm{x} = x_1, x_2, \ldots$ and a model, what is the probability (or likelihood) of x, $p(\mathrm{x}|\boldsymbol{\theta})$ given the model?
  - State sequence estimation/decoding:
    - Given an observed sequence (of errors) $\mathrm{x} = x_1, x_2, \ldots$ and a model, what is the state sequence $\mathrm{z} = z_1, z_2, \ldots$ that best explains the observations?
  - Model parameter estimation
    - Given an observed sequence (of errors), x, what are the optimal model parameters that maximize $p(\mathrm{x}|\boldsymbol{\theta})$?

HMM Example: Inference and Estimation

Example Problem 1: Likelihood evaluation for a given observation sequence

- How to obtain $p(\mathrm{x}|\boldsymbol{\theta})$?
- Cannot directly write an expression for $p(\mathrm{x}|\boldsymbol{\theta})$
  - Problem analogous to coin mixing problem used for EM
- Observations are "incomplete" and something is missing/hidden
  - What is missing?
  - State of channel
- Fix?
  - Introduce latent variables for state sequence: $z_1, z_2, \ldots, z_N$
  - What us the assumption on the state sequence?
    - Follows two state Markov Chain
- Helpful/necessary for all three problems

P1: Likelihood Evaluation for a given observation sequence

- Brute force: Enumerate all the state sequences and accumulate the probabilities of realizations of x:

$$p(\mathrm{x}|\boldsymbol{\theta}) \quad = \quad \sum_{z_1, z_2, \ldots, z_N} p(z_1, z_2, \ldots, z_N, x_1, x_2, \ldots x_N|\boldsymbol{\theta})_{\text{(Write Down!)}}$$

- What is the complexity of this computation?
  - This computation requires computation of $2 \times N \times 2^N$ computations.
  - Exponentially increasing with length of x. Infeasible!
  - For $N = 256$, $\approx 2^{256}$ computations required
- Alternative to brute force computation?
  - Use Markov structure underlying state evolution
    - Most readily understood in terms of Trellis diagram derived from state transition diagram

P1: Likelihood Evaluation for a given observation sequence

- Trellis diagram
  - Rolling out in time of state diagram
- Implication of Markov structure
  - Future independent of past given present state
  - States serve as checkpoints across time
    - Only links in trellis to states at time $(n+1)$ are from states at time n
    - Time hopping links between states are disallowed
  - Computed quantities for each state in trellis allow recursive computation over time
    - **without requiring consideration of past**
    - IMPORTANT: converts exponential complexity in time to linear

P1: Likelihood Evaluation for a Given Observation Sequence

- Trellis diagram

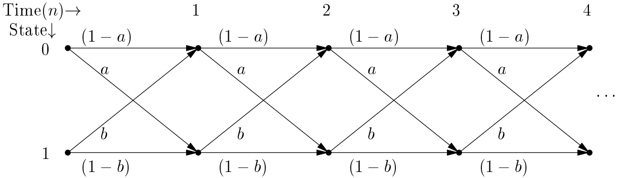


Figure: Trellis Diagram for 4 "coin" HMM example

- Mathematical implication: recursion

$$
\begin{aligned}
p(\mathrm{x}|\boldsymbol{\theta}) &= p(x_1, x_2, \ldots x_N|\boldsymbol{\theta}) \\
p(\mathrm{x}_1^n|\boldsymbol{\theta}) &= p(x_1, x_2, \ldots x_n|\boldsymbol{\theta}) \\
&= \sum_{z_n} p(\mathrm{x}_1^n, z_n|\boldsymbol{\theta})
\end{aligned}
$$

- Example: Evaluation of $p(0100 \mid \boldsymbol{\theta})$
  - Recall parameters a, b, p_g, p_b

Forward Recursion for Computation of Likelihood

- Forward recursion term

$$\alpha_n(z_n) \equiv \alpha\left(x_1^n, z_n\right) \stackrel{\text{def}}{=} p(x_1^n, z_n|\boldsymbol{\theta}) \tag{6}$$

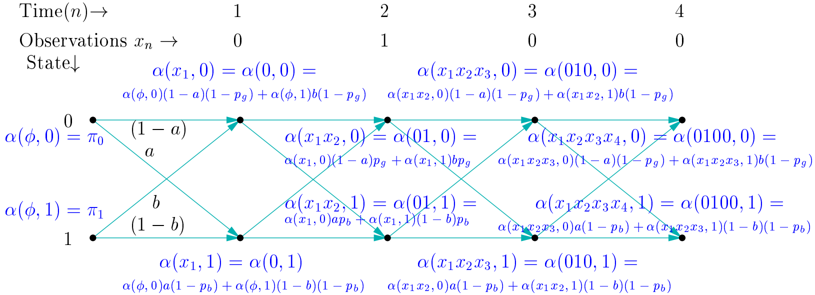


Figure: Example: Forward recursion for computation of the likelihood $p(x|\boldsymbol{\theta}) = \sum_{z_N} \alpha\left(x_1^N, z_N\right) = \sum_{z_N} \alpha_N(z_N)$.

P1: Likelihood Evaluation Algebraic Development

- Notational Conventions
  - $N =$ number of observations
  - $n =$ index
  - $\mathrm{x}_1^n =$ leading subsequence of observations upto time n
  - Complete observation sequence: $\mathrm{x} = \mathrm{x}_1^N$
  - Recall Bayes' Rule: $P(A|B) = \frac{P(A \bigcap B)}{P(B)}$
  - Conditional form of Bayes' Rule: $P(A|B \bigcap C) = \frac{P(A \bigcap B|C)}{P(B|C)}$
  - Event (set intersections) probabilities vs variables (comma separated list)
    - Joint probabilities: $P(A \bigcap B)$ vs. $p(x_1, x_2)$

P1: Likelihood Evaluation, Forward recursion

$$
\begin{aligned}
p(\mathrm{x}_1^n, z_n|\boldsymbol{\theta}) &= \sum_{z_{n-1}} p(\mathrm{x}_1^n, z_n, z_{n-1}|\boldsymbol{\theta}) \\
&= \sum_{z_{n-1}} p(\mathrm{x}_1^n \mid z_n, z_{n-1}, \boldsymbol{\theta}) p(z_n, z_{n-1} \mid \boldsymbol{\theta}) \\
&= \sum_{z_{n-1}} (p(x_n, \mathrm{x}_1^{n-1} \mid z_n, z_{n-1}, \boldsymbol{\theta}) \times \\
& p(z_n \mid z_{n-1}, \boldsymbol{\theta}) p(z_{n-1} \mid \boldsymbol{\theta})) \\
&= \sum_{z_{n-1}} (p(x_n \mid z_n, \boldsymbol{\theta}) p(\mathrm{x}_1^{n-1} \mid z_{n-1}, \boldsymbol{\theta}) \times \\
& p(z_n \mid z_{n-1}, \boldsymbol{\theta}) p(z_{n-1} \mid \boldsymbol{\theta})) \\
&= \sum_{z_{n-1}} p(x_n \mid z_n, \boldsymbol{\theta}) p(\mathrm{x}_1^{n-1}, z_{n-1} \mid \boldsymbol{\theta}) p(z_n \mid z_{n-1}, \boldsymbol{\theta})
\end{aligned}
$$

- Recognize recursive pattern

Derivation Justification: HMM Likelihood Evaluation, Forward Recursion

- Algebraic demonstration: using Markov property
  - Facts:
    - $x_n \perp \mathrm{x}_1^{n-1} \mid (z_n, z_{n-1})$
    - $x_n \perp z_{n-1} \mid z_n$
    - $\mathrm{x}_1^{n-1} \perp z_n \mid z_{n-1}$
    - Additional fact: $v_n = (x_n, z_n)$ is also a Markov process (Show!)

$$
\begin{aligned}
& p(x_n, \mathrm{x}_1^{n-1} \mid z_n, z_{n-1}, \boldsymbol{\theta}) && (7)\\
=& p(x_n \mid z_n, z_{n-1}, \boldsymbol{\theta}) \times p(\mathrm{x}_1^{n-1} \mid z_n, z_{n-1}, \boldsymbol{\theta}) && (8)\\
=& p(x_n \mid z_n, \boldsymbol{\theta}) p(\mathrm{x}_1^{n-1} \mid z_{n-1}, \boldsymbol{\theta}) && (9)
\end{aligned}
$$

- Graphical illustration based on trellis diagram

P1: Likelihood Evaluation, Forward recursion

- Recursion for forward probability

$$
\begin{aligned}
\alpha_n(z_n) &\equiv \alpha\left(\mathrm{x}_1^n, z_n\right) \\
&\stackrel{\text{def}}{=} p(\mathrm{x}_1^n, z_n|\boldsymbol{\theta}) \\
&= \sum_{z_{n-1}} p(x_n \mid z_n, \boldsymbol{\theta}) p(\mathrm{x}_1^{n-1}, z_{n-1} \mid \boldsymbol{\theta}) p(z_n \mid z_{n-1}, \boldsymbol{\theta}) \\
&= \sum_{z_{n-1}} g_{z_n}(x_n)\alpha\left(\mathrm{x}_1^{n-1}, z_{n-1}\right) p_{z_{n-1}z_n} \\
&= \sum_{z_{n-1}} p_{z_{n-1}z_n} g_{z_n}(x_n)\alpha_{n-1}(z_{n-1})
\end{aligned}
$$

- Final likelihood by marginalization

$$
p(\mathrm{x}|\boldsymbol{\theta}) = \sum_{z_N} p(\mathrm{x}_1^N, z_N|\boldsymbol{\theta}) = \sum_{z_n} \alpha\left(\mathrm{x}_1^N, z_N\right) = \sum_{z_n} \alpha_N(z_N)
$$

P2: Estimation of Most Likely State Sequence

- Want: MAP estimate of sequence of states

$$
\begin{aligned}
\hat{\mathrm{z}} &= \arg\max_{\mathrm{z}} p(\mathrm{z} \mid \mathrm{x}, \boldsymbol{\theta}) \\
&= \arg\max_{\mathrm{z}} \frac{p(\mathrm{z}, \mathrm{x} \mid \boldsymbol{\theta})}{p(\mathrm{x} \mid \boldsymbol{\theta})} \\
&= \arg\max_{\mathrm{z}} p(\mathrm{z}, \mathrm{x} \mid \boldsymbol{\theta})
\end{aligned}
$$

- What is the corresponding ML estimate and how does the above differ?
- Again challenging too determine directly
  - Exponential number of possible state sequences. Do not allow for brute force evaluation.
    - How many state sequences for N observations?
    - Generalization for L state Markov chain?
- Solution?
  - Use state variables and trellis once again

P2: Example: Estimation of Most Likely State Sequence

- Trellis diagram
- Example evaluation of

$$\hat{\mathrm{z}} = \arg\max_{\mathrm{z}} p(\mathrm{z}, 0100|\boldsymbol{\theta})$$

- Example parameters
  $\pi_0 = \pi_1 = 1/2, a = 1/10, b = 1/7, p_g = 1/100, p_b = 1/4$

P2: Most Likely State Sequence Estimation, Traceback

- Best "path metric" upto time n

$$\delta_n(z) \equiv \delta(\mathsf{x}_1^n, z) = \max_{\mathsf{z}_1^n : z_n = z} p(\mathsf{z}_1^n, \mathsf{x}_1^n \mid \boldsymbol{\theta})$$

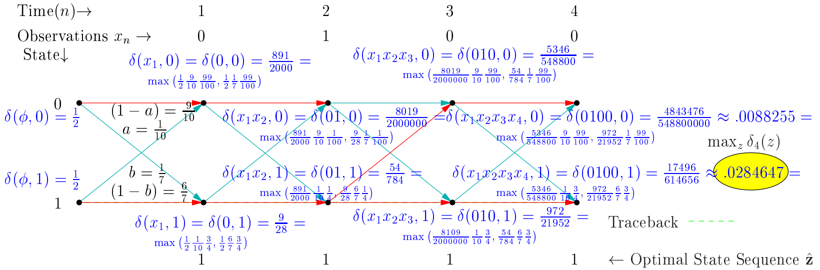


Figure: Example: Traceback following Viterbi recursion for estimation of the most likely state sequence $\hat{\mathsf{z}}$.

P2: Most Likely State Sequence: Mathematical recursion

- MAP Estimation of state sequence

$$\hat{\mathsf{z}} = \arg\max_{\mathsf{z}} p(\mathsf{z}, \mathsf{x} \mid \boldsymbol{\theta})$$

- Consider variant of objective function:

$$\delta_n(z) \equiv \delta(\mathsf{x}_1^n, z) = \max_{\mathsf{z}_1^{\mathsf{n}}: z_n = z} p(\mathsf{z}_1^{\mathsf{n}}, \mathsf{x}_1^n \mid \boldsymbol{\theta})$$

$$\max_{\mathsf{z}} p(\mathsf{z}, \mathsf{x} \mid \boldsymbol{\theta}) = \max_{z} \delta_N(z)$$

- $\delta_n(z)$ is the max probability over all possible paths on the trellis going upto time n and state z which can be recursively evaluated

P2: Most Likely State Sequence: Mathematical recursion

- Max probability over all possible paths on the trellis to state z

$$
\begin{aligned}
\delta_n(z) &= \max_{\mathrm{z}_1^{\mathrm{n}} : z_n = z} p(\mathrm{z}_1^{\mathrm{n}}, \mathrm{x}_1^{n} \mid \boldsymbol{\theta}) \\
&= \max_{\mathrm{z}_1^{\mathrm{n}}, b : z_{n-1} = b, z_n = z} p(\mathrm{z}_1^{\mathrm{n}}, \mathrm{x}_1^{n} \mid \boldsymbol{\theta}) \\
&= \max_{\mathrm{z}_1^{\mathrm{n-1}}, b : z_{n-1} = b, z_n = z} p([\mathrm{z}_1^{\mathrm{n-1}}, z], \mathrm{x}_1^{n} \mid \boldsymbol{\theta}) \\
&= \max_b \left(\max_{\mathrm{z}_1^{\mathrm{n-1}} : z_{n-1} = b} p(\mathrm{z}_1^{\mathrm{n-1}}, \mathrm{x}_1^{n-1} \mid \boldsymbol{\theta}) p_{bz} g_z(x_n) \right) \\
&= \max_b \left(\delta_{n-1}(b) p_{bz} g_z(x_n) \right)
\end{aligned}
$$

- Bellman's principle: Dynamic Programming [3, 4]
  - Solve larger problem recursively in terms of already solved smaller problems

P2: Most Likely State Sequence: Mathematical recursion

- Recursion commonly implemented in log-domain
- Multiplication becomes summation

$$\ln \delta_n(z) = \max_b \left(\ln \delta_{n-1}(b) + \ln p_{bz} + \ln g_z(x_n)\right)$$

- Also has advantage of dynamic range
- Note additive costs are log odds ratios
- Trace-back for determining MAP path

Problem 1 and 2: Similarities

- Maximization instead of product
  - Exchange operators: $\sum \prod \leftrightarrow \max \sum$
- Sum-product and Max-Sum nomenclature
- Can be shown to be instances of same fundamental underlying structure
- Both are instances of dynamic programming [3, 4]
- Can be expressed in a common framework based on Semi-ring structure (Aji and McEliece [1])

Dynamic Programming

- Technique for recursively solving larger problem by using solutions of smaller subproblems
- Example: estimation of likelihood for sub-sequence and re-using for overall sequence
- Enables polynomial time simplification of several computational problems for which brute force complexity is exponential
- Requires problem to have appropriate structure
  - Bigger problem decomposable into smaller problems
  - Decomposition "conformal" with objective function being maximized/evaluated

P3: Parameter Estimation for HMMs

- Baum-Welch Iteration [2] $\equiv$ EM Algorithm
- Intuition: Analogous to EM motivation
  - ML Parameter estimation would be easy if states were known
  - States are not known so estimate these and use estimates
- A first approach: find MAP state sequence $\hat{z}$ and then estimate parameters using ML with the complete likelihood with these known states
  - Iterate by using re-estimate parameters to re-estimate $\hat{z}$, referred to as "hard" EM
  - "Re-estimation" vs. "estimation"

P3: "Hard EM" Parameter Estimation for HMMs

- Parameters $\boldsymbol{\theta} = [\boldsymbol{\pi}, a, b, p_g, p_b]$
- Current estimate of parameters $\boldsymbol{\theta}^{(t)} = \left[\boldsymbol{\pi}^{(t)}, a^{(t)}, b^{(t)}, p_g^{(t)}, p_b^{(t)}\right]$
- Decode: Obtain MAP estimate $\hat{\mathrm{z}}$ of state sequence

$$\hat{\mathrm{z}} = \arg\max_{\mathrm{z}} p(\mathrm{z}, \mathrm{x} \mid \boldsymbol{\theta})$$

- How?
  - Solve Problem 2 with parameters set to current estimate $\boldsymbol{\theta}^{(t)}$
- Update: Update parameters to new value $\boldsymbol{\theta}^{t+1}$ by setting these to ML estimates for "complete likelihood" with state sequence set as $\hat{\mathrm{z}}$

$$\hat{\boldsymbol{\theta}}^{t+1} = \arg\max_{\boldsymbol{\theta}} p\left(\mathrm{x}, \hat{\mathrm{z}} \mid \boldsymbol{\theta}\right)$$

- How? Recall ML estimation for Bernoulli random variables
- Increment t, repeat Decode and Update steps till convergence
- Note hard decisions on states

P3: "Hard EM" Parameter Update

- Parameter update using ML estimation of four Bernoulli random variables
  - Estimated probabilities correspond to occurrence fractions

$$a^{(t+1)} = \frac{\#\text{ trans. } 0 \rightarrow 1 \text{ in } \hat{\mathrm{z}}}{\#\text{ trans. starting from } 0 \text{ in } \hat{\mathrm{z}}}$$

$$b^{(t+1)} = \frac{\#\text{ trans. } 1 \rightarrow 0 \text{ in } \hat{\mathrm{z}}}{\#\text{ trans. starting from } 1 \text{ in } \hat{\mathrm{z}}}$$

$$p_g^{(t+1)} = \frac{\#\text{ 1's in x where } \hat{\mathrm{z}} \text{ is } 0}{\#\text{ times state is } 0 \text{ in } \hat{\mathrm{z}}}$$

$$p_b^{(t+1)} = \frac{\#\text{ 1's in x where } \hat{\mathrm{z}} \text{ is } 1}{\#\text{ times state is } 1 \text{ in } \hat{\mathrm{z}}}$$

$$\pi_j^{(t+1)} = \begin{cases} (\#\text{ times } \hat{\mathrm{z}} \text{ is in state } j\ )\,/\,N & \text{ergodic} \\ \chi(\hat{z}_j) & \text{non ergodic} \end{cases}$$

P3: "Hard EM" Parameter Update Example

- Consider 4 coin HMM toy example
  - Let observation sequence be $x = 0100101010000101$
  - Hard EM Steps:
    - Estimate MAP state sequence $\hat{z}$, say $z = 0000111111000011$
    - What are the parameter estimates, given:

$$x = 0100101010000101$$
$$z = 0000111111000111$$

- How would actual EM differ?
  - Recall our toy EM example, what did we need (in terms of indicator variables)?

P3: Parameter Estimation for HMMs

- Actual EM estimate $\equiv$ Baum-Welch re-estimation procedure
- Instead of "hard" decoding use probabilistic estimates
- Replace fractions in "hard EM" with expectations
- All expectations under current estimate of parameters
- Formal derivation as EM algorithm: Will consider relation in general setting

P3: Baum-Welch Parameter Estimation for HMMs

- Hard decision estimates replaced by conditional expectations

$$a^{(t+1)} = \frac{E\left[\#\text{ trans. } 0 \to 1 \text{ in z} \mid \text{x}, \boldsymbol{\theta}^{(t)}\right]}{E\left[\#\text{ trans. starting from } 0 \text{ in z} \mid \text{x}, \boldsymbol{\theta}^{(t)}\right]}$$

$$b^{(t+1)} = \frac{E\left[\#\text{ trans. } 1 \to 0 \text{ in z} \mid \text{x}, \boldsymbol{\theta}^{(t)}\right]}{E\left[\#\text{ trans. starting from } 1 \text{ in z} \mid \text{x}, \boldsymbol{\theta}^{(t)}\right]}$$

$$p_g^{(t+1)} = \frac{E\left[\#\text{ 1's in x where z is } 0 \mid \text{x}, \boldsymbol{\theta}^{(t)}\right]}{E\left[\#\text{ times state is } 0 \text{ in z} \mid \text{x}, \boldsymbol{\theta}^{(t)}\right]}$$

$$p_b^{(t+1)} = \frac{E\left[\#\text{ 1's in x where z is } 1 \mid \text{x}, \boldsymbol{\theta}^{(t)}\right]}{E\left[\#\text{ times state is } 1 \text{ in z} \mid \text{x}, \boldsymbol{\theta}^{(t)}\right]}$$

$$\pi_j^{(t+1)} = \begin{cases} E\left[\#\text{ times z is in state } j \mid \text{x}, \boldsymbol{\theta}^{(t)}\right] / N & \text{ergodic} \\ E\left[\chi(z_j) \mid \text{x}, \boldsymbol{\theta}^{(t)}\right] & \text{non ergodic} \end{cases}$$

P3: Baum-Welch Parameter Estimation for HMMs

- Computation of expectations requires estimates of posterior probabilities of states rather than "hard" estimates of states
  - Recall EM for mixture models: E step $\equiv$ computation of posterior probabilities for belonging to a component
- How can we compute posterior probabilities for state taking a given value at a particular time instant n
- Posterior probability that state at time n is j

$$p(z_n = j|\mathrm{x}, \boldsymbol{\theta}) = \sum_{\mathrm{z}:z_n=j} p(\mathrm{z}|\mathrm{x}, \boldsymbol{\theta}) = \frac{\sum_{\mathrm{z}:z_n=j} p\left(\mathrm{x}, \mathrm{z} \mid \boldsymbol{\theta}\right)}{p\left(\mathrm{x} \mid \boldsymbol{\theta}\right)}$$

$$= \frac{p\left(\mathrm{x}, z_n = j \mid \boldsymbol{\theta}\right)}{p\left(\mathrm{x} \mid \boldsymbol{\theta}\right)} \qquad \textit{How?}$$

P3: Individual State Posterior Probability Estimation for HMMs

- Posterior probability that state at time n is j (given observations z and current estimate of parameters $\boldsymbol{\theta}$

$$p(z_n = j|\mathrm{x}, \boldsymbol{\theta}) = \frac{p\left(\mathrm{x}, z_n = j \mid \boldsymbol{\theta}\right)}{p\left(\mathrm{x} \mid \boldsymbol{\theta}\right)}$$

$$p\left(\mathrm{x}, z_n = j \mid \boldsymbol{\theta}\right) = ?$$

- Decomposition of posterior probability that state at time n is j for efficient evaluation

$$p\left(\mathrm{x}, z_n = j \mid \boldsymbol{\theta}\right) = p\left(\mathrm{x}_1^n, z_n = j \mid \boldsymbol{\theta}\right) p\left(\mathrm{x}_{n+1}^N \mid z_n = j, \boldsymbol{\theta}\right) \quad \textit{Why?}$$

P3: Individual State Posterior Probability Estimation for HMMs

$$p\left(\mathrm{x}, z_n = j \mid \boldsymbol{\theta}\right) = p\left(\mathrm{x}_1^n, z_n = j \mid \boldsymbol{\theta}\right) p\left(\mathrm{x}_{n+1}^N \mid z_n = j, \boldsymbol{\theta}\right)$$

- What term do we recognize here?

$$\alpha_n\left(z_n\right) \equiv \alpha\left(\mathrm{x}_1^n, z_n\right) \stackrel{\text{def}}{=} p(\mathrm{x}_1^n, z_n | \boldsymbol{\theta})$$

- How did we compute this?
- Remaining term: backward term

$$\beta_n\left(z_n\right) \equiv \beta\left(\mathrm{x}_{n+1}^N, z_n\right) \stackrel{\text{def}}{=} p(\mathrm{x}_{n+1}^N \mid z_n, \boldsymbol{\theta})$$

- Important: note difference between backward term and forward term. Backward term is conditioned on state

P3: Backward Recursion for HMMs

- Computation of individual state posterior probabilities requires a "backward term" in addition to already computed forward term
  - Can be computed recursively, just like forward term

$$\beta_n(z_n) \equiv \beta\left(\mathrm{x}_{n+1}^N, z_n\right) \stackrel{\text{def}}{=} p(\mathrm{x}_{n+1}^N \mid z_n, \boldsymbol{\theta})$$

- Consider 4 "coin" example trellis
  - Example evaluation of $p(0100 \mid \boldsymbol{\theta})$
    - Recall parameters a, b, p_g, p_b

Backward Recursion for Example HMM

- Backward recursion term

$$\beta_n(z_n) \equiv \beta\left(\mathsf{x}_{n+1}^N, z_n\right) \stackrel{\text{def}}{=} p(\mathsf{x}_{n+1}^N \mid z_n, \boldsymbol{\theta}) \tag{15}$$

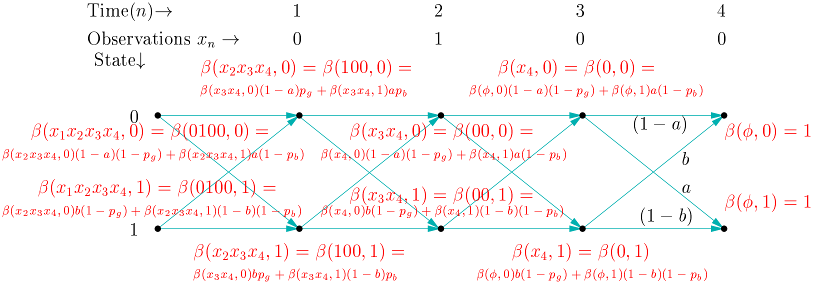


Figure: Example: Backward recursion for computation of the posterior probabilities required for Baum-Welch (EM) iterations.

P3: Backward Recursion, Algebraic Development

$$
\begin{aligned}
p(\mathrm{x}_{n+1}^{N} \mid z_n, \boldsymbol{\theta}) &= \sum_{z_{n+1}} p(\mathrm{x}_{n+1}^{N}, z_{n+1} \mid z_n, \boldsymbol{\theta}) \\
&= \sum_{z_{n+1}} p(\mathrm{x}_{n+1}^{N} \mid z_{n+1}, z_n, \boldsymbol{\theta}) p(z_{n+1} \mid z_n, \boldsymbol{\theta}) \\
&= \sum_{z_{n+1}} p(x_{n+1}, \mathrm{x}_{n+2}^{N} \mid z_{n+1}, z_n, \boldsymbol{\theta}) p(z_{n+1} \mid z_n, \boldsymbol{\theta}) \\
&= \sum_{z_{n+1}} p(x_{n+1} \mid z_{n+1}, z_n, \boldsymbol{\theta}) \times \\
&\quad p(\mathrm{x}_{n+2}^{N} \mid z_{n+1}, z_n, \boldsymbol{\theta}) p(z_{n+1} \mid z_n, \boldsymbol{\theta}) \\
&= \sum_{z_{n+1}} p(x_{n+1} \mid z_{n+1}, \boldsymbol{\theta}) p(\mathrm{x}_{n+2}^{N} \mid z_{n+1}, \boldsymbol{\theta}) p(z_{n+1} \mid z_n, \boldsymbol{\theta})
\end{aligned}
$$

- Recognize recursive pattern

P3: Backward Recursion, Derivation Justification

- Algebraic demonstration: using Markov property
  - Facts:
    - $x_{n+1} \perp \mathrm{x}_{n+2}^{N} \mid (z_{n+1}, z_n)$
    - $x_{n+1} \perp z_n \mid z_{n+1}$
    - $\mathrm{x}_{n+2}^{N} \perp z_n \mid z_{n+1}$

$$
\begin{aligned}
& p(x_{n+1}, \mathrm{x}_{n+2}^{N} \mid z_{n+1}, z_n, \boldsymbol{\theta}) \\
=& p(x_{n+1} \mid z_{n+1}, z_n, \boldsymbol{\theta}) p(\mathrm{x}_{n+2}^{N} \mid z_{n+1}, z_n, \boldsymbol{\theta}) \\
=& p(x_{n+1} \mid z_{n+1}, \boldsymbol{\theta}) p(\mathrm{x}_{n+2}^{N} \mid z_{n+1}, \boldsymbol{\theta})
\end{aligned}
$$

P3: Backward Recursion

- Recursion for backward probability

$$
\begin{aligned}
\beta_n(z_n) &\equiv \beta\left(\mathrm{x}_{n+1}^N, z_n\right) \\
&\stackrel{\text{def}}{=} p(\mathrm{x}_{n+1}^N \mid z_n, \boldsymbol{\theta}) \\
&= \sum_{z_{n+1}} p(x_{n+1} \mid z_{n+1}, \boldsymbol{\theta}) p(\mathrm{x}_{n+2}^N \mid z_{n+1}, \boldsymbol{\theta}) p(z_{n+1} \mid z_n, \boldsymbol{\theta}) \\
&= \sum_{z_{n+1}} g_{z_{n+1}}(x_{n+1}) \beta\left(\mathrm{x}_{n+2}^N, z_{n+1}\right) p_{z_{n+1} z_n} \\
&= \sum_{z_{n+1}} g_{z_{n+1}}(x_{n+1}) p_{z_{n+1} z_n} \beta_{n+1}(z_{n+1})
\end{aligned}
$$

P3: Backward Recursion for HMMs

- Backward recursion

$$\begin{aligned}\beta_n(i) &= \sum_{j=1}^{L} p_{ij} g_j(x_{n+1}) \beta_{n+1}(j), \quad N-1 \geq n \geq 1 \\ \beta_N(i) &= 1, L \geq i \geq 1\end{aligned}$$

- Graphical illustration based on trellis diagram
  - Seen for example

P3: Baum-Welch Updates

- Use forward-backward recursion outputs to update parameters

$$
\begin{aligned}
\overline{p}_{ij} &= \frac{E\left[\#\text{ of transitions } i \to j \mid \mathrm{x}, \boldsymbol{\theta}^{(t)}\right]}{E\left[\#\text{ of transitions from } i \mid \mathrm{x}, \boldsymbol{\theta}^{(t)}\right]} \\
&= \frac{\sum_{n=1}^{N-1} \alpha_n(i) p_{ij} g_j(x_{n+1}) \beta_{n+1}(j)}{\sum_{n=1}^{N-1} \alpha_n(i) \beta_n(i)} \\
\overline{g}_i(k) &= \frac{E\left[\#\text{ of observations of symbol } v_k \text{ as output of } i \mid \mathrm{x}, \boldsymbol{\theta}^{(t)}\right]}{E\left[\#\text{ of emissions from } i \mid \mathrm{x}, \boldsymbol{\theta}^{(t)}\right]} \\
&= \frac{\sum_{n=1|x_n=v_k}^{N} \alpha_n(i) \beta_n(i)}{\sum_{n=1}^{N} \alpha_n(i) \beta_n(i)} \\
\overline{\pi}_{i_1}^{(t+1)} &= E\left[\#\text{ of } z_1 = i \mid \mathrm{x}, \boldsymbol{\theta}^{(t)}\right] \\
&= \begin{cases} \left(\sum_{n=1}^{N} \alpha_i(t) \beta_i(t)\right) / N & \text{(Ergodic)} \\ \alpha_i(1) \beta_i(1) & \text{(otw.)} \end{cases}
\end{aligned}
$$

P3: Complete Baum-Welch Parameter Estimation Procedure for HMMs

- Parameters: Initial state probs., transition probs., per state emission probs. $\boldsymbol{\theta} = \left[\boldsymbol{\pi}, \mathsf{P}, \{g_i(k)\}_{i=1}^{L}\right]$
- Current estimate of parameters $\boldsymbol{\theta}^{(t)}$
- Perform Forward-Backward iterations to obtain $\alpha_n(i) \stackrel{\mathrm{def}}{=} \alpha\left(\mathsf{x}_1^n, z_n = i\right)$ and $\beta_n(i) \stackrel{\mathrm{def}}{=} \beta\left(\mathsf{x}_{n+1}^N, z_n = i\right)$ for all n
- Update: Update parameters to new value $\boldsymbol{\theta}^{(t+1)}$ as expected number of occurrences of appropriate events
- Increment t, repeat Forward-Backward and Update steps till convergence
- Note soft decisions on states

P3: Baum-Welch for Example HMM

- Illustration: Parameter update for $a =$ probability of transition from $0 \to 1$
- Forward-Backward iterations with current parameter estimate $\boldsymbol{\theta}^{(t)} = \left[\boldsymbol{\pi}^{(t)}, a^{(t)}, b^{(t)}, p_g^{(t)}, p_b^{(t)}\right]$ provide, for all n

$$\alpha_n(i) \stackrel{\text{def}}{=} \alpha\left(\mathrm{x}_1^n, z_n = i\right) \tag{16}$$

$$\beta_n(i) \stackrel{\text{def}}{=} \beta\left(\mathrm{x}_{n+1}^N, z_n = i\right) \tag{17}$$

- Updated parameter $a^{(t+1)}$

$$a^{(t+1)} = \frac{E\left[\#\text{ of transitions } 0 \to 1 \mid \mathrm{x}, \boldsymbol{\theta}^{(t)}\right]}{E\left[\#\text{ of transitions from state } 0 \mid \mathrm{x}, \boldsymbol{\theta}^{(t)}\right]}$$

$$= \frac{\sum_{n=1}^{N-1} \alpha_n(0) a^{(t)} g_1(x_{n+1}) \beta_{n+1}(1)}{\sum_{n=1}^{N-1} \alpha_n(0)\beta_n(0)} \tag{18}$$

Example: Forward-Backward and Baum-Welch Estimation

- Specific term: $\alpha_2(0) p_{01} g_1(x_3) \beta_3(1)$
  - Joint probability of observation x and transition from state 0 to state 1 at time 2 given (current) parameter estimates $\boldsymbol{\theta}^{(t)}$

$$
\begin{aligned}
& p(z_3 = 1, z_2 = 0, \mathrm{x} \mid \boldsymbol{\theta}^{(t)}) \\
& = p(x_1, x_2, z_2 = 0 \mid \boldsymbol{\theta}^{(t)}) p_{01}^{(t)} g_1(x_3) p(x_4 \mid z_3 = 1, \boldsymbol{\theta}^{(t)}) \\
& = \alpha_2(0) p_{01} g_1(x_3) \beta_3(1) = \alpha_2(0) a^{(t)} (1 - p_b^{(t)}) \beta_3(1) \qquad (19)
\end{aligned}
$$

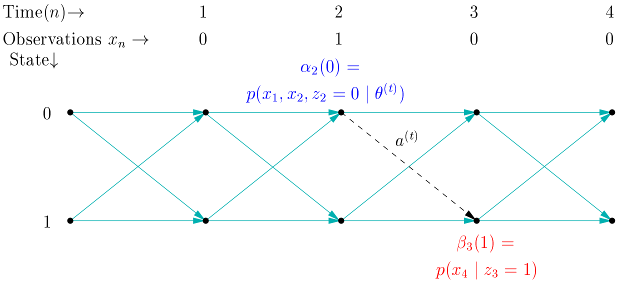


Figure: Example: Baum-Welch computation example.

Example: Forward-Backward and Baum-Welch Estimation

- Parameter update $a^{(t+1)}$ to posterior probability of a transition from state 0 to state 1 given the observations and (current) parameter estimates

$$
\begin{aligned}
a^{(t+1)} &= \frac{E\left[\#\text{ of transitions } 0 \to 1 \mid \mathrm{x}, \boldsymbol{\theta}^{(t)}\right]}{E\left[\#\text{ of transitions from state } 0 \mid \mathrm{x}, \boldsymbol{\theta}^{(t)}\right]} \\
&= \frac{\sum_{n=1}^{N-1} p(z_n = 0, z_{n+1} = 1 \mid \mathrm{x}, \boldsymbol{\theta}^{(t)})}{\sum_{n=1}^{N-1} p(z_n = 0 \mid \mathrm{x}, \boldsymbol{\theta}^{(t)})} \\
&= \frac{\sum_{n=1}^{N-1} p(\mathrm{x}, z_n = 0, z_{n+1} = 1 \mid \boldsymbol{\theta}^{(t)})}{\sum_{n=1}^{N-1} p(\mathrm{x}, z_n = 0 \mid \boldsymbol{\theta}^{(t)})} \quad \text{Why?} \\
&= \frac{\sum_{n=1}^{N-1} \alpha_n(0) a^{(t)} g_1(x_{n+1}) \beta_{n+1}(1)}{\sum_{n=1}^{N-1} \alpha_n(0) \beta_n(0)}
\end{aligned}
$$

Example: Forward-Backward and Baum-Welch Estimation

- Parameter update $a^{(t+1)}$ to posterior probability of a transition from state 0 to state 1 given the observations and (current) parameter estimates

$$a^{(t+1)} = \frac{\sum_{n=1}^{N-1} \alpha_n(0) a^{(t)} g_1(x_{n+1}) \beta_{n+1}(1)}{\sum_{n=1}^{N-1} \alpha_n(0) \beta_n(0)}$$

$$= \frac{\text{Total probability you observe x and transition } 0 \to 1 \text{ (dashed arrow links)}}{\text{Total probability observe x and transition from state 0 (green nodes)}}$$

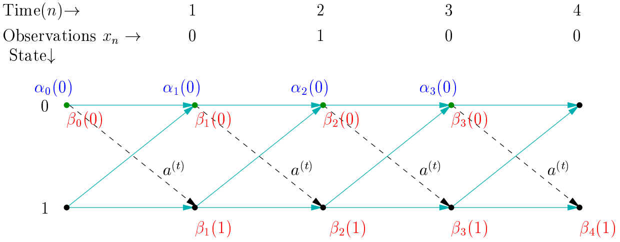


Figure: Example: Baum-Welch computation example.

Baum-Welch Estimation and EM

- Term $\alpha_n(i)p_{ij}g_j(x_{n+1})\beta_{n+1}(j)$
  - Joint probability of observation x and transition from state i to state j at time n given (current) parameter estimates

$$
\begin{aligned}
& p(\mathrm{x}, z_n = i, z_{n+1} = j \mid \boldsymbol{\theta}^{(t)}) \\
=& p(\mathrm{x}_1^n, z_n = i|\boldsymbol{\theta})p_{ij}g_j(x_{n+1})p(\mathrm{x}_{n+2}^N \mid z_{n+1} = j, \boldsymbol{\theta}) \\
=& \alpha_n(i)p_{ij}g_j(x_{n+1})\beta_{n+1}(j)
\end{aligned} \tag{20}
$$

- Term $\alpha_n(i)\beta_n(i)$
  - Joint probability of observation x and transition from state i at time n given (current) parameter estimates

$$
\begin{aligned}
& p(\mathrm{x}, z_n = i \mid \boldsymbol{\theta}^{(t)}) \\
=& p(\mathrm{x}_1^n, z_n = i|\boldsymbol{\theta})p(\mathrm{x}_{n+1}^N \mid z_n = i, \boldsymbol{\theta}) \\
=& \alpha_n(i)\beta_n(i)
\end{aligned} \tag{21}
$$

Baum-Welch Estimation and EM

- Posterior probability of a transition from state i to state j given the observations and (current) parameter estimates $= \frac{\sum_{n=1}^{N-1} p(\mathrm{x}, z_n = i, z_{n+1} = j | \boldsymbol{\theta}^{(t)})}{\sum_{n=1}^{N-1} p(\mathrm{x}, z_n = i | \boldsymbol{\theta}^{(t)})} = \frac{\sum_{n=1}^{N-1} \alpha_n(i) p_{ij} g_j(x_{n+1}) \beta_{n+1}(j)}{\sum_{n=1}^{N-1} \alpha_n(i) \beta_n(i)}$
- Recall indicator variables in EM formulation and posterior probabilities of the indicator variables corresponding to conditional expectations
  - The Baum-Welch posterior probabilities are completely analogous
  - The Baum-Welch algorithm is an instance of EM
    - Can show relation more pedantically and formally by introducing corresponding indicator variables
- Convergence: Can show that $p(\mathrm{x}|\boldsymbol{\theta}^{(t+1)}) \geq p(\mathrm{x}|\boldsymbol{\theta}^{(t)})$, i.e., the observed data likelihood is a non-decreasing function with successive re-estimation iterations
  - Iteratively re-estimating parameters yields a local maxima of the likelihood

P3: Individual State Posterior Probability Estimation for HMMs

- Independent utility of obtaining MAP estimate of state at any given time n

$$
\begin{aligned}
\hat{z}_n &= \arg\max_i p(z_n = i | \mathrm{x}, \boldsymbol{\theta}) \\
p(z_n = i | \mathrm{x}, \boldsymbol{\theta}) &= \sum_{\mathrm{z}: z_n = i} p(\mathrm{z} | \mathrm{x}, \boldsymbol{\theta})
\end{aligned}
$$

- Also enabled by forward-backward recursion
- Application: in MAP decoding for convolutional codes for error correction and in modified form in Turbo decoding

P3: Individual State Posterior Probability Estimation for HMMs

- MAP estimate of state at any given time n

$$p(z_n = i|\mathrm{x}, \boldsymbol{\theta}) = \sum_{\mathrm{z}:z_n=i} p(\mathrm{z}|\mathrm{x}, \boldsymbol{\theta}) = \frac{\sum_{\mathrm{z}:z_n=i} p(\mathrm{z}, \mathrm{x} \mid \boldsymbol{\theta})}{p(\mathrm{x} \mid \boldsymbol{\theta})}$$

$$\begin{aligned}
\sum_{\mathrm{z}:z_n=i} p(\mathrm{z}, \mathrm{x} \mid \boldsymbol{\theta}) &= \sum_{\mathrm{z}:z_n=i} p(\mathrm{z}, \mathrm{x} \mid \boldsymbol{\theta}) = p(\mathrm{x}, z_n = i \mid \boldsymbol{\theta}) \\
&= p(\mathrm{x}_1^n, \mathrm{x}_{n+1}^N, z_n = i \mid \boldsymbol{\theta}) \\
&= p(\mathrm{x}_1^n, \mathrm{x}_{n+1}^N \mid z_n = i, \boldsymbol{\theta}) p(z_n = i \mid \boldsymbol{\theta}) \\
&= p(\mathrm{x}_1^n \mid z_n = i, \boldsymbol{\theta}) p(\mathrm{x}_{n+1}^N \mid z_n = i, \boldsymbol{\theta}) p(z_n = i \mid \boldsymbol{\theta}) \\
&= p(\mathrm{x}_1^n, z_n = i \mid \boldsymbol{\theta}) p(\mathrm{x}_{n+1}^N \mid z_n = i, \boldsymbol{\theta}) \\
&= \alpha_n(i)\beta_n(i) \qquad (22)
\end{aligned}$$

- Obtained directly from forward-backward recursion results
  - Also seen earlier in Baum-Welch

General HMM Formulation

- Straightforward generalization of toy example
- Recall general HMM defining elements
  - Unobserved Markov state process: $z_1, z_2, \ldots z_N$
    - State possibilities: $\mathrm{S} = \{S_1, S_2, \ldots, S_L\}$, L = number of states
    - State transition probability matrix: $\mathrm{P} = \{p_{ij}\}$
    - Initial state probabilities: $\pi_i = p(z_0 = S_i), \boldsymbol{\pi} = [\pi_i]$
  - Observed HMM output sequence $x_1, x_2, \ldots x_N$
    - Output possibilities: $\mathrm{v} = \{v_1, v_2, \ldots, v_M\}$
    - State dependent emission probabilities: $G = \{g_i(v_k)\}$
  - Model parameters $\boldsymbol{\theta} = (\mathrm{P}, \mathrm{G}, \boldsymbol{\pi})$

HMM Example in Standardized Notation

- Underlying two state Markov chain with observations stochastically dependent on state
  - Transition probabilities p_{ij}
  - State dependent emission probabilities $g_i(x) =$ probability symbol x emitted in state i
  - Note parameterization is over-specified and constraints apply for the parameters

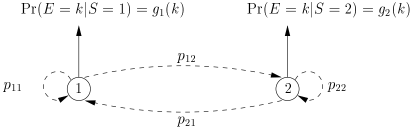


Figure: From example to generic model for HMMs.

Recall Three Basic Problems for HMMs

- Likelihood evaluation for a given observation sequence: Given an observation sequence $\mathrm{x} = x_1, x_2, \ldots$ and model parameters, what is the probability (or likelihood) of x, $p(\mathrm{x}|\boldsymbol{\theta})$ given the model parameters?
- State sequence estimation/decoding: Given an observation sequence $\mathrm{x} = x_1, x_2, \ldots$ and model parameters, what is the state sequence $\mathrm{z} = z_1, z_2, \ldots$ that best explains the observations?
- Model parameter estimation: Given an output sequence, x, what are the optimal model parameters that maximize $p(\mathrm{x}|\boldsymbol{\theta})$?

Trellis Representation

- Useful for all three problems

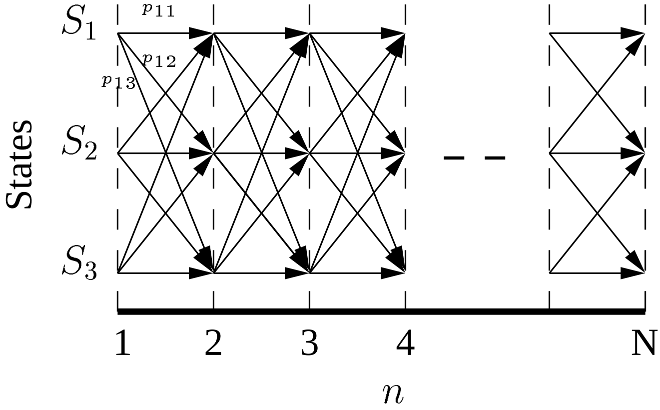


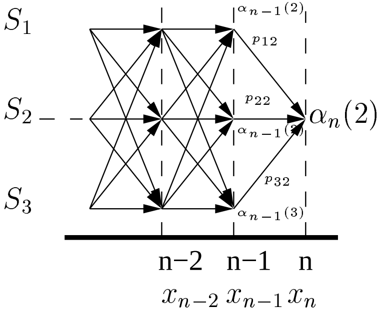
Figure: Trellis of states for a 3 state HMM

Forward HMM Recursion

- Joint probability of being at state S_i and emitting at $x_1, x_2, \ldots, x_n$ at time instant n

$$\alpha_n(i) \stackrel{\text{def}}{=} p(x_1, x_2, \ldots, x_n, z_n = S_i|\boldsymbol{\theta})$$
$$= \sum_{j=1}^{L} \alpha_{n-1}(j) p_{ji} g_i(x_n), \quad N \geq n \geq 1, \quad L \geq i \geq 1$$
$$\alpha_0(i) = \pi_i$$

- Enables computation of likelihood
  - $p(\mathrm{x}|\boldsymbol{\theta}) = \sum_{i=1}^{L} \alpha_N(i)$
- Sum product algorithm
- Also part of Baum-Welch Parameter re-estimation process



Viterbi Algorithm: Most Likely State Sequence

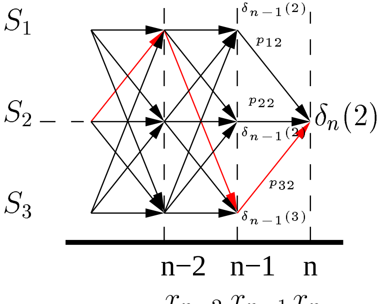
- Best "Path Metric" variable
- Maximum probability over all state sequences of observing $x_1, x_2, \ldots, x_n$ and ending up in state S_i at time instant n

$$\delta_n(i) = \max_{z_1, z_2, \ldots, z_{n-1}} p(z_1, z_2, \ldots, z_n = i, x_1, x_2, \ldots, x_n | \boldsymbol{\theta})$$

$$= \max_j [\delta_{n-1}(j) p_{ji}] g_i(x_n), N \geq n \geq 1, L \geq i \geq 1$$

$$\delta_0(i) = \pi_i$$

- Implemented in log domain
  - Max sum algorithm
- Traceback for obtaining optimal state sequence

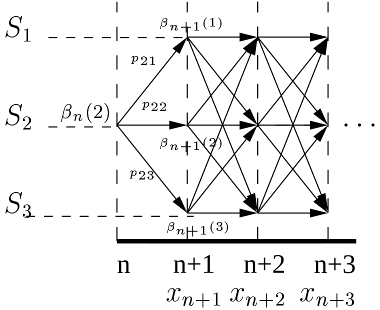


Backward HMM Recursion

- Conditional probability of observing $x_{n+1}, x_{n+2} \dots x_N$ given that state at time instant n is S_i

$$\beta_n(i) = p(x_{n+1}, x_{n+2}, \dots, x_N | z_n = S_i, \boldsymbol{\theta})$$
$$= \sum_{j=1}^{L} p_{ij} g_j(x_{n+1}) \beta_{n+1}(j), \quad N-1 \geq n \geq 0$$
$$\beta_N(i) = 1, L \geq i \geq 1$$

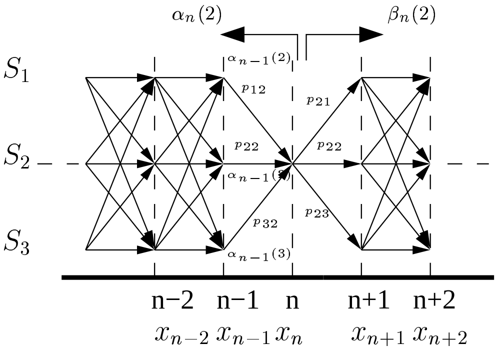
- Part of Baum-Welch Parameter re-estimation process



Maximization of APP of each state

- Choose the state sequence $\mathrm{z} = \{z_1, z_2, \dots z_N\}$ such that:

$$
\begin{aligned}
z_n &= \arg\max_{S_i} p(z_n = S_i|\mathrm{x}, \boldsymbol{\theta}) \\
p(z_n = S_i|\mathrm{x}, \boldsymbol{\theta}) &= \sum_{\mathrm{z}:z_n=S_i} p(\mathrm{z}|\mathrm{x}, \boldsymbol{\theta}) \\
&= \sum_{\mathrm{z}:z_n=S_i} p(\mathrm{z}, \mathrm{x}|\boldsymbol{\theta})/p(\mathrm{x}|\boldsymbol{\theta}) \\
&= \left(\alpha_n(i)\beta_n(i)\right)/p(\mathrm{x}|\boldsymbol{\theta})
\end{aligned}
$$



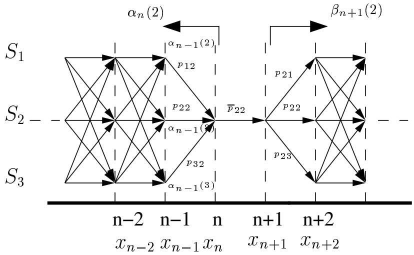
Baum-Welch Parameter Estimation

- Forward-Backward recursions to obtain arrays α and β
- Re-estimation of model parameters

$$\overline{p}_{ij} = \frac{\sum_{n=1}^{N-1} \alpha_n(i) p_{ij} g_j(x_{n+1}) \beta_{n+1}(j)}{p(\mathrm{x}|\boldsymbol{\theta})}$$

$$\overline{g}_i(k) = \frac{E(\text{frequency of observing symbol } v_k \text{ as output of } S_i)}{E(\text{frequency of transitions from } S_i)}$$

$$= \frac{\sum_{n=1|x_n=v_k}^{N} \alpha_n(i)\beta_n(i)}{\sum_{n=1}^{N} \alpha_n(i)\beta_n(i)}$$



HMM Implementation Issues: Dynamic Range

- Re-consider our 4 "coin" toy example
  - Consider magnitude of forward recursion values for increasing n
- Values will decrease as you proceed to larger n along the sequence
  - Nature of decrease?
    - Pretty rapid: decrease is exponential in n
  - Computational implication: values will underflow
    - Eventually becoming smaller than machine ϵ
  - How to address?

HMM Implementation Issues: Dynamic Range

- Accommodating dynamic range of recursion values without underflow
  - Two approaches
    - Log-domain computation (log makes exponential fall-off linear), plus following identity for numerical stability
    - $\log\left(\sum_i \exp(t_i)\right) = a + \log\left(\sum_i \exp(t_i + a)\right), \forall a \in \mathbb{R}$, used with $a = \max t_i$
    - Scaling - by an exponentially increasing scale factor
    - Scale factor accounted for separately. Not required for a number of inference tasks, see Rabiner's tutorial [10] for details.
- Absolutely critical for any HMM implementation

HMM Implementation Issues

- Description assumed observed symbols are emitted after transition to state
- Alternative assumptions
  - Observed symbols are emitted during transition and depend on originating state
    - Instead of state to which transition is occurring
  - Minor changes in details
- One or other convention may be more suitable/natural for a given problem
  - Available software toolkits (see Reading List) invariably require adaptation to problem setting

Belief Propagation

- The HMM forward-backward algorithms allow us to compute the marginal probability of being in a state at a given point in time
  - These computations correspond to an instance of "Belief Propagation"
    - Methodology for propagation of belief about related quantities to iteratively estimate desired marginals
    - Formalized and defined in a general framework by Judea Pearl [8, 9]
- Provides exact solutions for marginal probabilities on Directed Acyclic Graphs (DAGs)
  - The trellis representations we used for HMMs are examples of (linear) DAGs
- Have also been used effectively for graphs with cycles
- Cycles capture inter-dependencies rather than one way dependency

Belief Propagation

- In the presence of cycles, belief propagation is not guaranteed to converge and marginal probabilities computed by belief propagation may not be correct
- For many interesting and challenging problems, however, "loopy" belief propagation provides good results
  - Example: Error correction decoding using LDPC and Turbo codes
  - In these settings, Belief propagation provides a framework for understanding algorithms derived from other simplifications/intuition/heuristics as an approximation
- Time permitting will visit an example using Turbo/LDPC codes

Hidden Markov Models: Extensions/Theory

- Our discussion focused entirely on discrete situations
  - Discrete state space
  - Discrete time
- Generalizations exist where either or both the state space and time may be continuous
  - Often referred to as "Continuous Time HMMs"
  - Conceptually similar to the HMMs we discussed
  - Mathematical formulation and development is, however, much more involved [7, 6]
    - Transition probabilities $\rightarrow$ transition rates
    - Stochastic differential equations define the evolution
- Hidden Markov Models/Processes (theoretical considerations) [5]

References I

[1] S. M. Aji and R. J. McEliece. “The generalized distributive law”. In: *IEEE Trans. on Inform. Theory* 46.2 (2000), pp. 325–343.

[2] L. E. Baum et al. “A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains”. In: *Ann. Math. Statist.* 41.1 (1970), pp. 164–171.

[3] Richard Bellman. “Dynamic Programming”. In: *Science* 153.3731 (1966), pp. 34–37. ISSN: 0036-8075. DOI: 10.1126/science.153.3731.34.

[4] Dimitri P Bertsekas. *Dynamic programming and optimal control*. Belmont, MA: Athena Scientific, 1995.

[5] Y. Ephraim and N. Merhav. “Hidden Markov processes”. In: *IEEE Trans. on Inform. Theory* 48.6 (2002), pp. 1518–1569. DOI: 10.1109/TIT.2002.1003838.

References II

[6] Y Liu et al. "Efficient continuous-time hidden Markov model for disease modeling". In: *Proc. Advances in Neural Info. Proc. Sys. (NIPS)*. 2015.

[7] Yu-Ying Liu et al. "Learning Continuous-Time Hidden Markov Models for Event Data". In: *Mobile Health: Sensors, Analytic Methods, and Applications*. Cham: Springer International Publishing, 2017, pp. 361–387. ISBN: 978-3-319-51394-2. DOI: 10.1007/978-3-319-51394-2_19.

[8] J. Pearl. "Fusion, propagation and structuring in belief networks". In: *Artificial Intelligence* 29.3 (1986), pp. 241–288.

[9] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann, 1988.

References III

[10] L.R. Rabiner. "A tutorial on hidden Markov models and selected applications in speech recognition". In: *Proc. IEEE* 77.2 (1989), pp. 257–286.