CMPT 413
Computational Linguistics

Anoop Sarkar

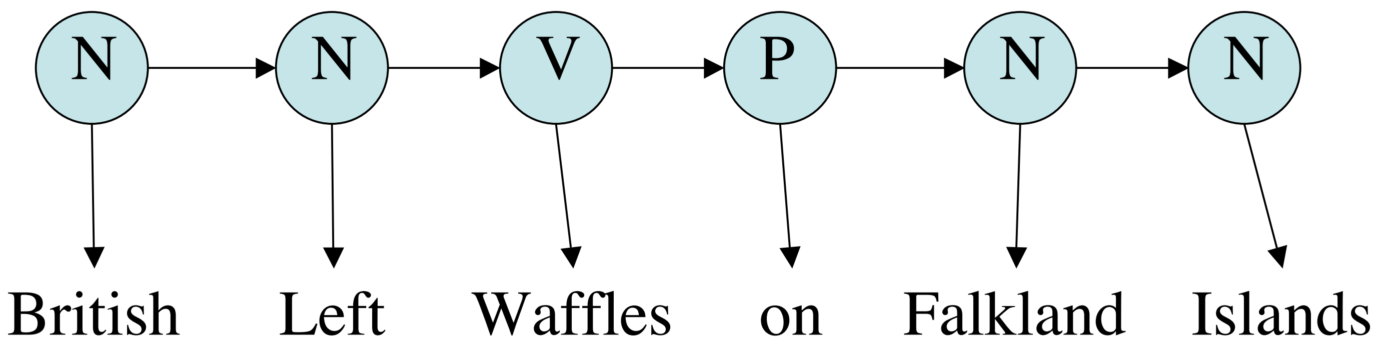
`http://www.cs.sfu.ca/~anoop`

Sequence Learning

- British Left Waffles on Falkland Islands
  - (N, N, V, P, N, N)
  - (N, V, N, P, N, N)
- Segmentation 中国十四个边境开放城市经济建设成就显著
  - (s, b, i, b, i, b, b, i, b, i, b, i, b, i, b, i, b, i, b, i)

中国	十四	个	边境	开放	城市	经济	建设	成就	显著
China	's	14	open	border	cities	marked	economic	achievements	

Sequence Learning

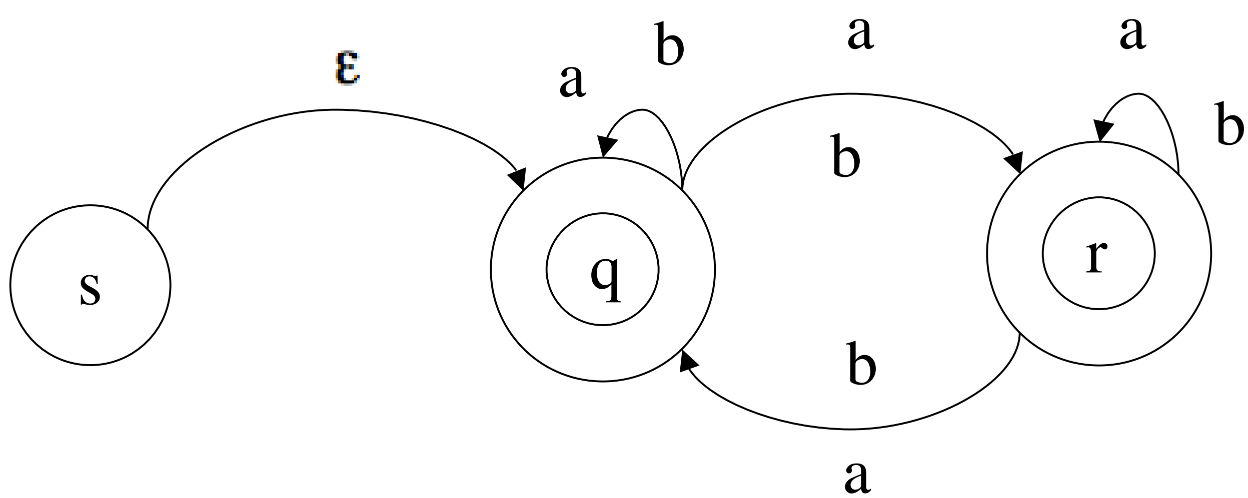


3 states: N, V, P

Observation sequence: $(o_1, \ldots o_6)$

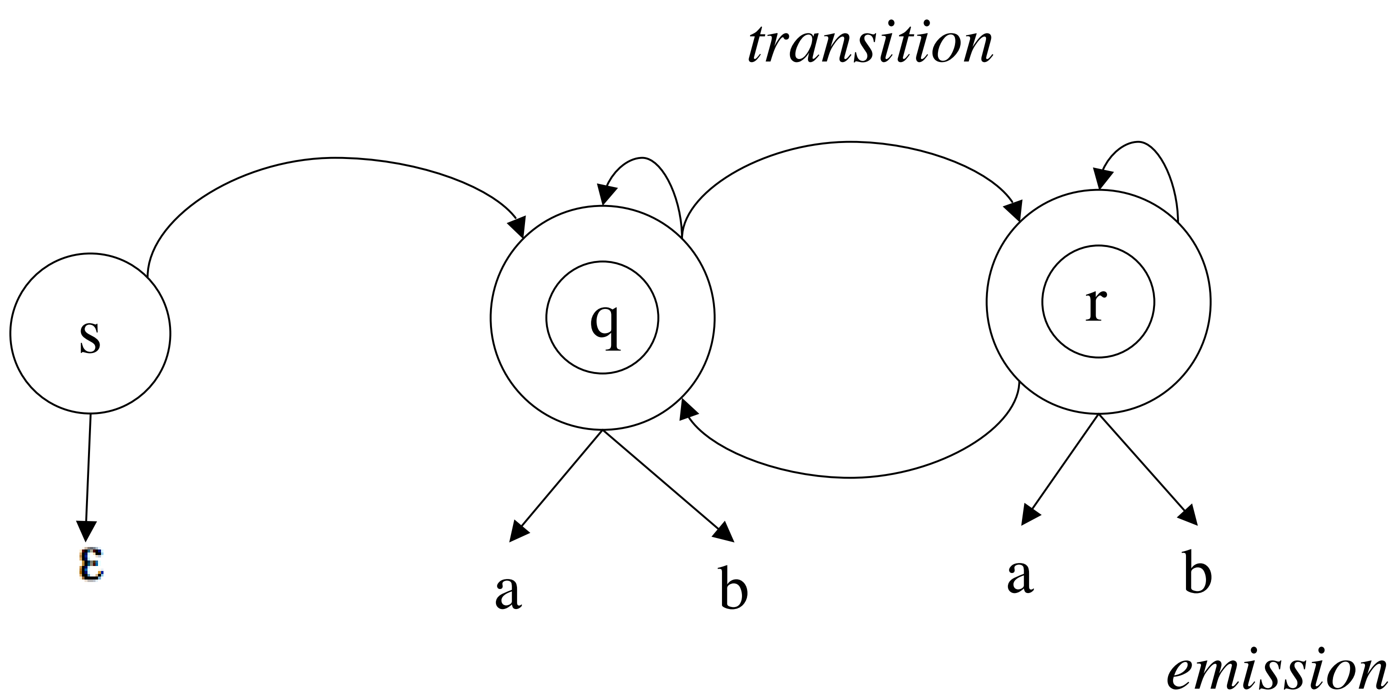
State sequence (6+1): *(Start, N, N, V, P, N, N)*

Finite State Machines



Mealy Machine

Finite State Machines

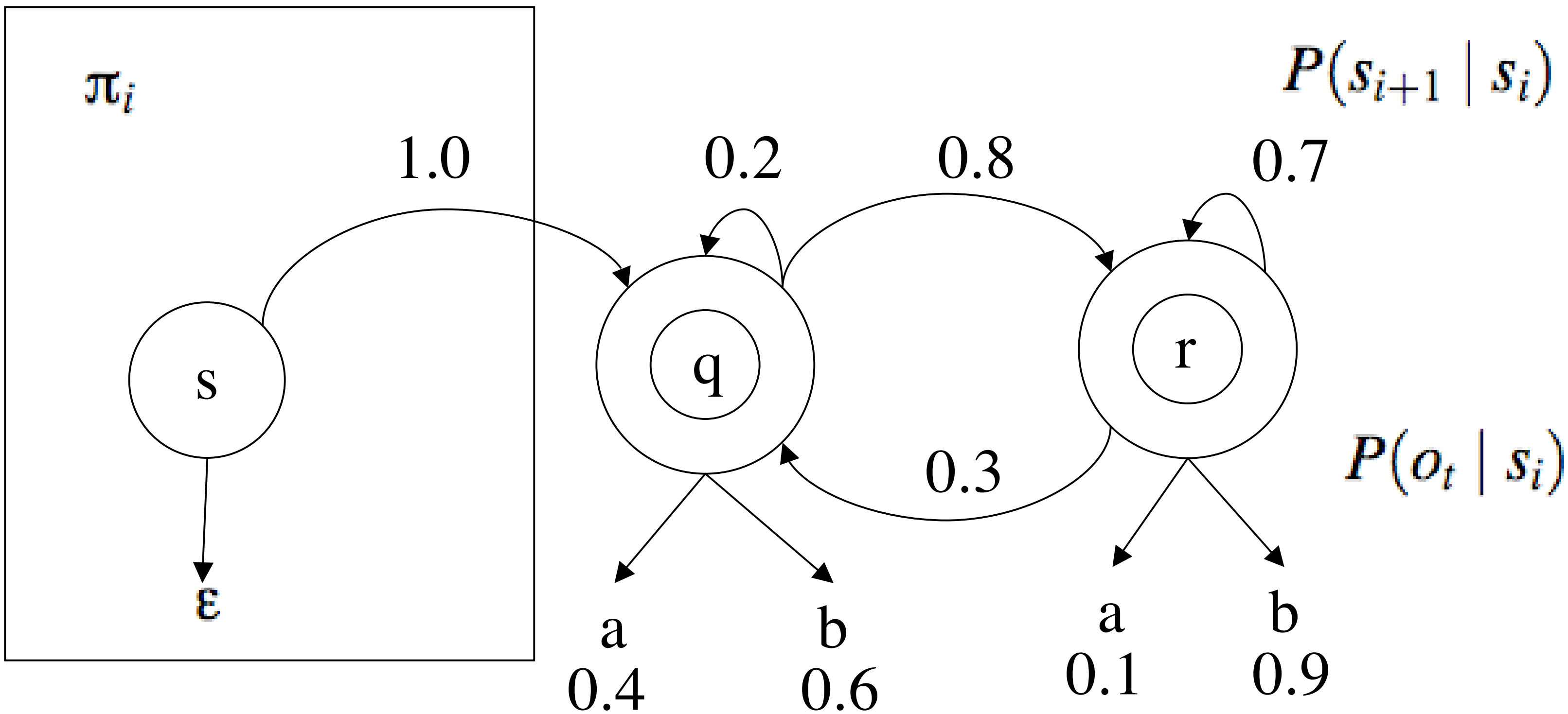


Moore Machine

Probabilistic FSMs

- Each transition is associated with a *transition probability*
- Each emission is associated with an *emission probability*
- Two conditions:
  - All outgoing transition arcs from a state must sum to 1
  - All emission arcs from a state must sum to 1

Probabilistic FSMs



$$\sum_x P(q \rightarrow x) = P(q \rightarrow r) + P(q \rightarrow q) = 1.0$$

$$\sum_x P(\text{emit}(q,x)) = P(\text{emit}(q,a)) + P(\text{emit}(q,b)) = 1.0$$

Hidden Markov Models

- There are n states $s_1, \ldots, s_i, \ldots, s_n$
- The emissions are observed (input data)
- Observation sequence $\mathbf{O}=(o_1, \ldots, o_t, \ldots, o_T)$
- The states are not directly observed (hidden)
- Data does not directly tell us which state X_t is linked with observation o_t

$$X_t \in \{s_1, \ldots, s_n\}$$

Markov Chains vs. HMMs

- Given an observation sequence

  $\mathbf{O}=(o_1, \ldots, o_t, \ldots, o_T)$
- An *n*th order Markov Chain computes the probability $P(o_1, \ldots, o_t, \ldots, o_T)$
- An HMM computes the probability

  $P(X_1, \ldots, X_{T+1}, o_1, \ldots, o_T)$ where the state sequence is *hidden*

Properties of HMMs

- Markov assumption

$$P(X_t = s_i \mid \ldots, X_{t-1} = s_j)$$

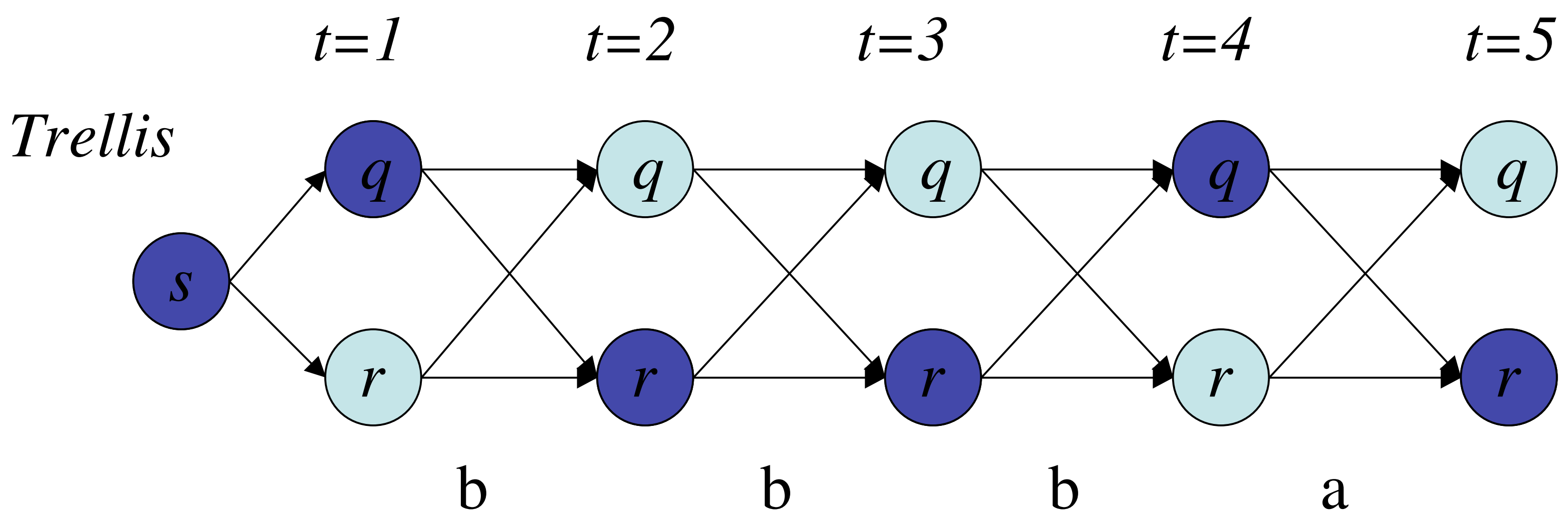
- Stationary distribution

$$P(X_t = s_i \mid X_{t-1} = s_j) = P(X_{t+l} = s_i \mid X_{t+l-1} = s_j)$$

HMM Algorithms

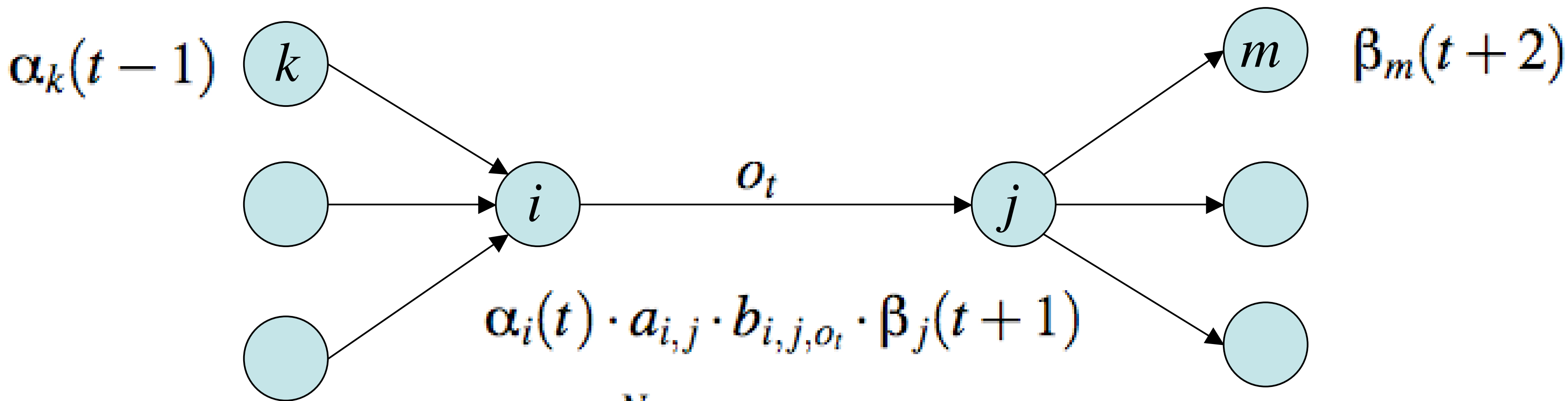
- HMM as language model: compute probability of given observation sequence
- HMM as parser: compute the best sequence of states for a given observation sequence
- HMM as learner: given a set of observation sequences, learn its distribution, i.e. learn the transition and emission probabilities

Best Path (Viterbi) Algorithm



- Key Idea 1: storing just the best path doesn't work
- Key Idea 2: store the best path upto *each* state

Forward-Backward Algorithm

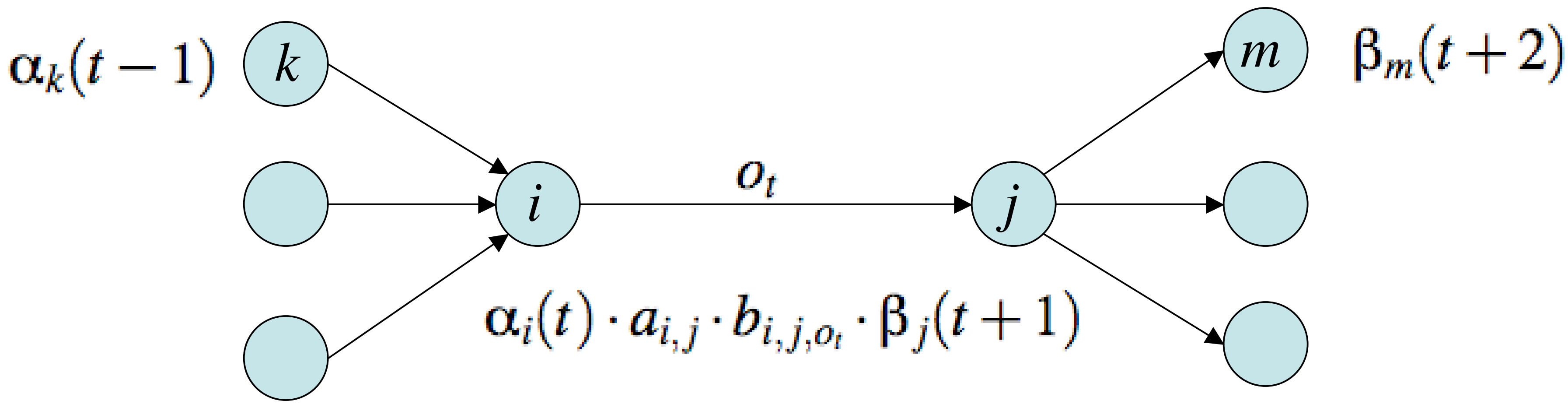


$$\alpha_i(t) = \sum_{k=1}^{N} a_{k,i} \cdot b_{k,i,o_{t-1}} \cdot \alpha_k(t-1)$$

$$\beta_j(t+1) = \sum_{m=1}^{N} a_{j,m} \cdot b_{j,m,o_{t+1}} \cdot \beta_m(t+2)$$

$$P(o_1, \ldots, o_T) = \sum_{i=1}^{N} \alpha_i(T+1) = \sum_{i=1}^{N} \pi_i \cdot \beta_i(1)$$

Forward-Backward Algorithm



$$\hat{f}(i,j,o_t) = \frac{\alpha_i(t) \cdot a_{i,j} \cdot b_{i,j,o_t} \cdot \beta_j(t+1)}{P(o_1, \ldots, o_T)} \qquad \hat{f}(i,j) = \sum_{t=1}^{T} \hat{f}(i,j,o_t)$$

$$a'_{i,j} = \frac{\hat{f}(i,j)}{\sum_{j=1}^{N} \hat{f}(i,j)} \qquad b'_{i,j,k} = \frac{\sum_{t=1}^{T} \hat{f}(i,j,o_t = k)}{\sum_{j=1}^{N} \hat{f}(i,j)}$$

Forward-Backward Algorithm

- Each iteration provides new values for all the *parameters*
- But are the new parameters any better? How can we tell?
- Compute cross entropy
- For HMMs, Baum 1977 shows that cross entropy will always be non-increasing (later generalized to the more general EM algorithm)
- Same as likelihood is non-decreasing

$$KL(\mu_{i+1} \,||\, D) \leq KL(\mu_i \,||\, D)$$