Statistical sequence recognition

Deterministic sequence recognition

- Last time, temporal integration of local distances via DP
  - Integrates local matches over time
  - Normalizes time variations
  - For cts speech, segments as well as classifies
- Limitations
  - End-point detection required
  - Choosing local distance (effect on global)
  - Doesn't model context effect between words

Statistical vs deterministic sequence recognition

- Statistical models can also be used (rather than examples) with DP
  - Still integrate local matches over time
  - Normalize time variations
  - For cts speech, segment as well as classify
- Helping with DTW Limitations
  - End-point detection not as critical
  - Local "distance" comes from the model
  - Cross-word modeling is straightforward (though it does require enough data to train good models)

Statistical sequence recognition

- Powerful tools exist
  - Density estimation
  - Training data alignment
  - Recognition given the models
- Increases generality over deterministic
  - Any distance choice is equivalent to implicit model
  - Sufficiently detailed statistics can model any distribution
  - In recognition, find MAP choice for sequence
  - In practice, approximations used

Probabilistic problem statement

- Bayes relation for models

$$P(M_j \mid X) = \frac{P(X \mid M_j)P(M_j)}{P(X)}$$

where M_j is the j[th] stat model for a sequence of speech units

And X is the sequence of feature vectors

– Minimum probability of error if j chosen to maximize $P(M_j \mid X)$

Decision rule

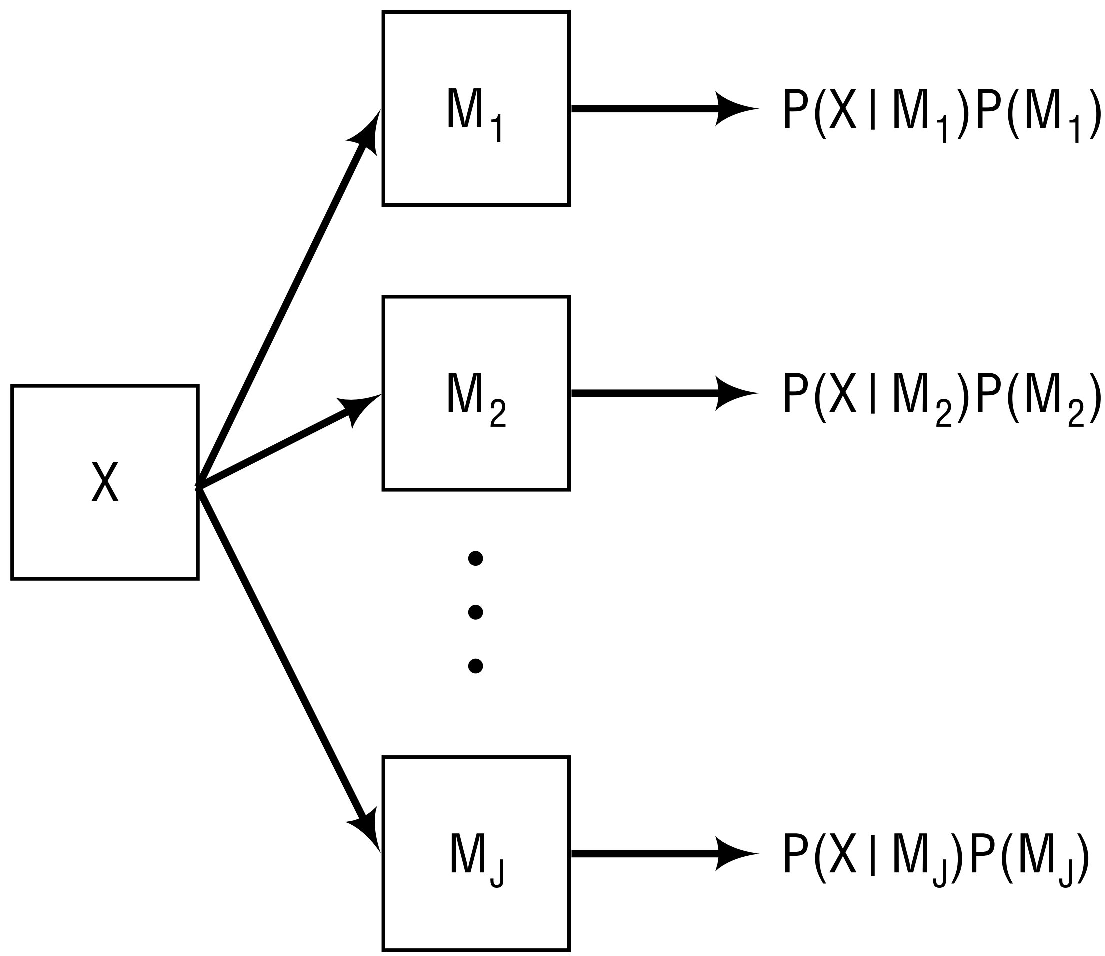
- Bayes relation for models

$j_{best} = \text{argmax}_j \quad P(M_j \mid X)$

$= \text{argmax}_j \quad P(X \mid M_j)P(M_j)$

since X is fixed for all choices of j

Bayes decision rule for models



Acoustic and language models

- So far, no assumptions
- But how do we get the probabilities?
- Estimate from training data
- Then, first assumption: acoustic parameters independent of language parameters

$$j_{best} = \arg\max_j P(M_j | X, \theta) = \arg\max_j P(X | M_j, \theta_A) P(M_j | \theta_L)$$

where θ are parameters estimated from training data

The three problems

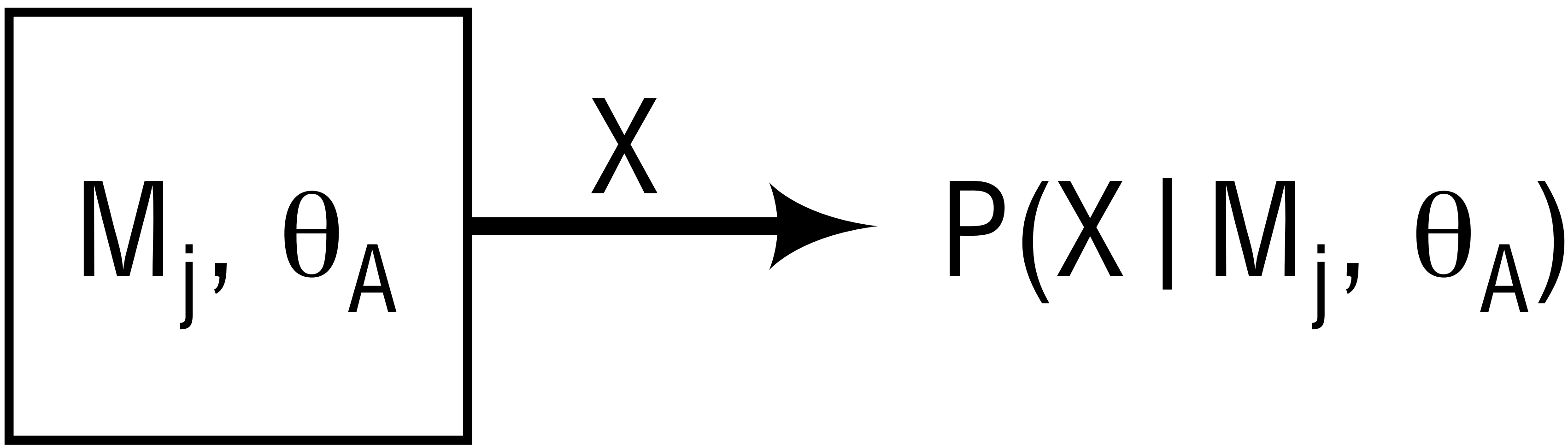
(1) How should $P(X \mid M_j, \theta_A)$ be computed?

(2) How should parameters θ_A be determined?

(3) Given the model and parameters, how can we find the best sequence to classify the input sequence?

- Today's focus is on problem 1, with some on problem 3; problem 2 will be next time.

Generative model for speech



Composition of a model

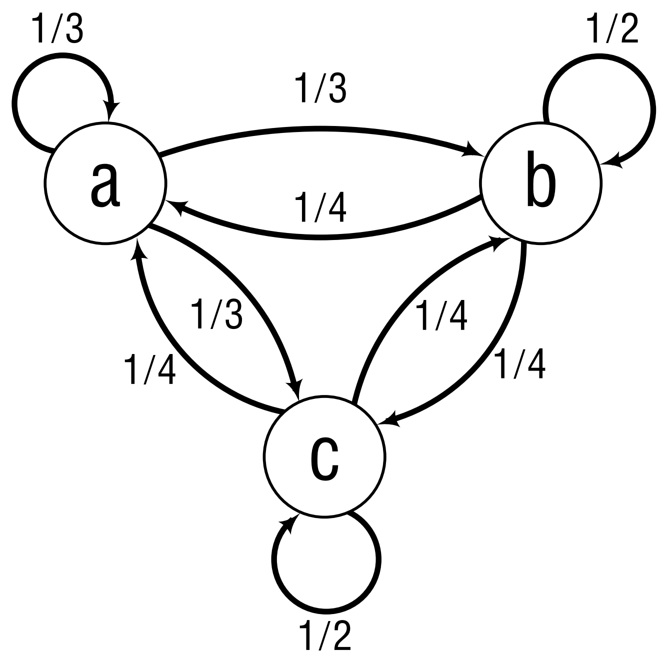
- Could collect statistics for whole words
- More typically, densities for subword units
- Models consist of states
- States have possible transitions
- States have observed outputs (feature vectors)
- States have density functions
- General statistical formulations hard to estimate
- To simplify, use Markov assumptions

Markov assumptions

- Assume finite state automaton
- Assume stochastic transitions
- Each random variable only depends on the previous *n* variables (typically *n*=1)
- HMMs have another layer of indeterminacy
- Let's start with Markov models per se

Example: Markov Model

state		output
a	⟷	sunny
b	⟷	cloudy
c	⟷	rainy



Numbers on arcs are probabilities of transitions

Markov Model

- By definition of joint and conditional probability, if

$$Q = (q^1, q^2, q^3, \ldots, q^N)$$

then

$$P(Q) = P(q^1) \prod_{i=2}^{N} P(q^i \mid q^{i-1}, q^{i-2}, \ldots, q^1)$$

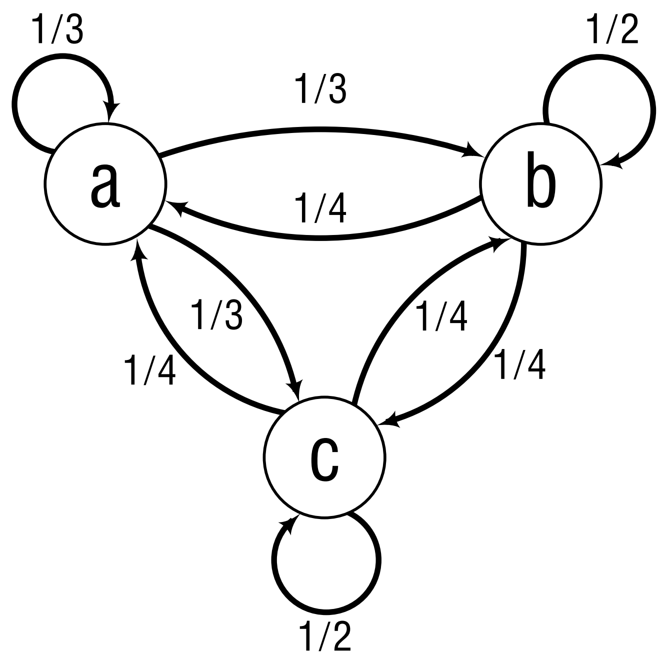
- And with 1st order Markov assumption,

$$P(Q) = P(q^1) \prod_{i=2}^{N} P(q^i \mid q^{i-1})$$

Example: Markov Model

$$P(abc) = P(c \mid b)P(b \mid a)P(a)$$

If we assume that we start with "a" so that P(a)=1, then

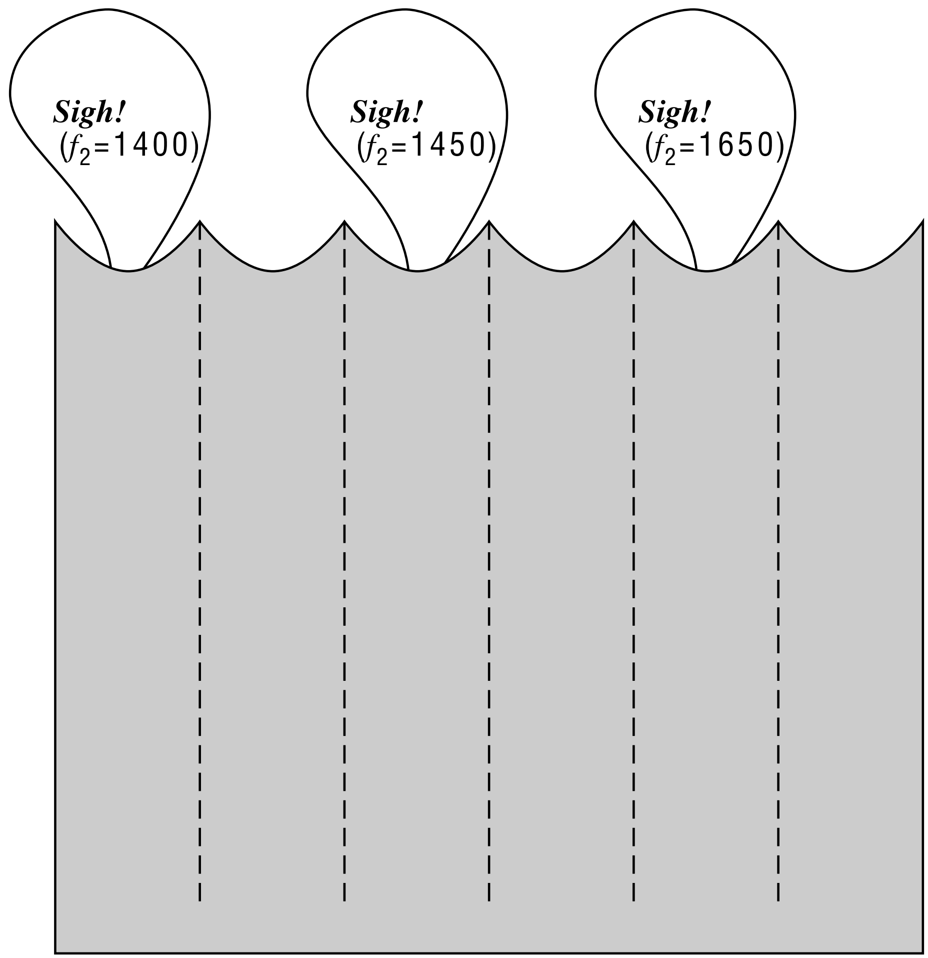


$$P(abc) = (\frac{1}{4})(\frac{1}{3}) = \frac{1}{12}$$

Hidden Markov Model (HMM)

- Outputs of Markov Model are deterministic
- For HMM, outputs are stochastic
  - Instead of a fixed value, a pdf
  - Generating state sequence is "hidden"
- "Doubly stochastic"
  - Transitions
  - Emissions

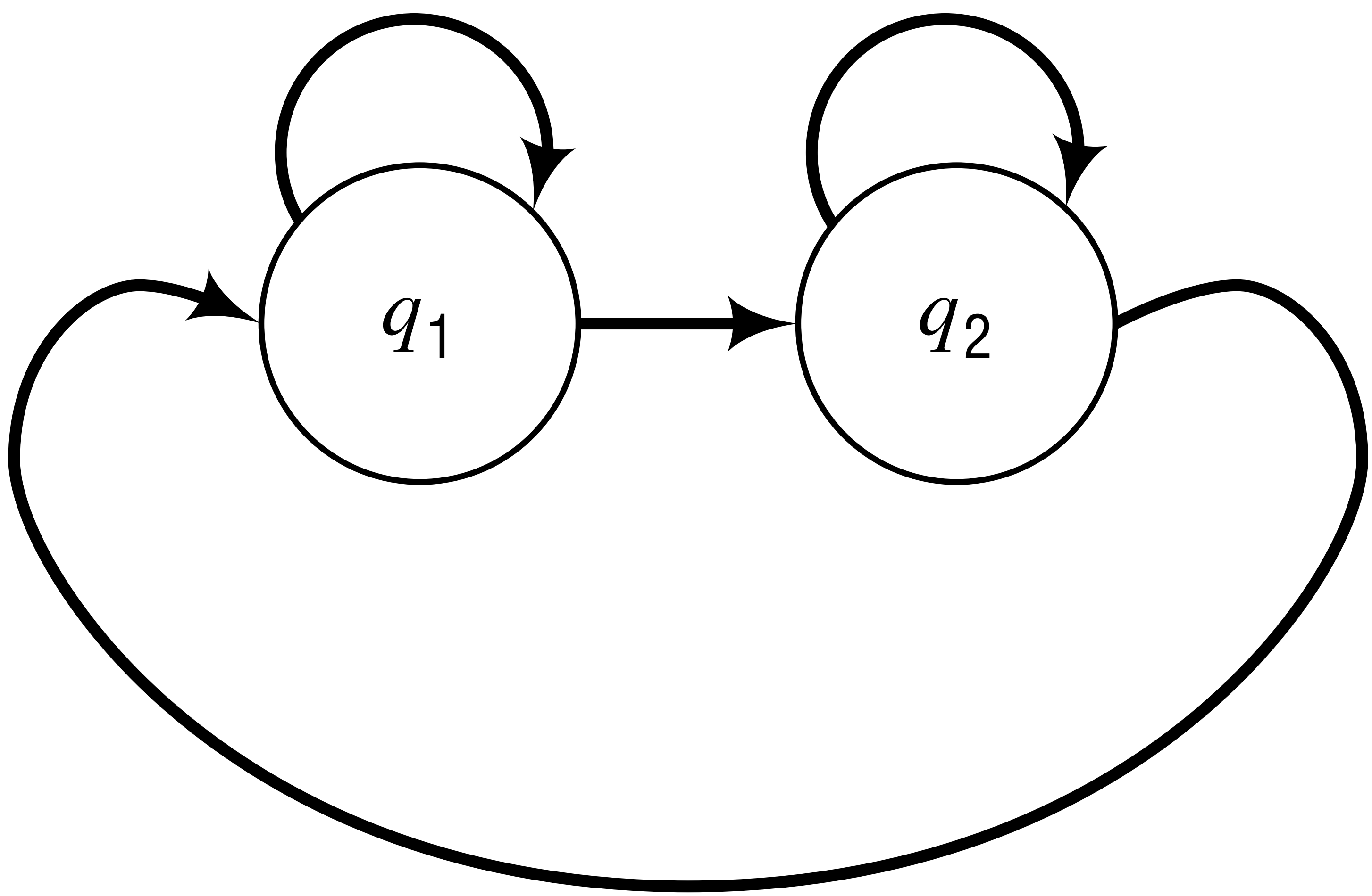
Line-up: ASR researchers and Swedish basketball players



MM vs HMM

- MM: bbplayer->1400 Hz; researcher->1600 Hz
HMM: bbplayer-> 1400 Hz mean, 200 Hz std dev etc. (Gaussian)
- Outputs are observed in both cases; but only one possible for MM, >1 possible for HMM
- For MM, then directly know the state; for HMM, probabilistic inference
- In both cases, two states, two possible transitions from each

Two-state HMM



Associated functions: $P(x_n \mid q_i)$ and $P(q_j \mid q_i)$

Emissions and transitions

- $P(x_n \mid q_1)$ could be density for F2 of bbplayer (emission probability)
- $P(x_n \mid q_2)$ could be density for F2 of researcher
- $P(q_2 \mid q_1)$ could be probability for transition of bbplayer->researcher in the lineup
- $P(q_1 \mid q_2)$ could be probability for transition of researcher->bbplayer in the lineup
- $P(q_1 \mid q_1)$ could be probability for transition of bbplayer->bbplayer in the lineup
- $P(q_2 \mid q_2)$ could be probability for transition of researcher->researcher in the lineup

States vs speech classes

- State is just model component; could be "tied" (same class as a state in a different model)
- States are often parts of a speech sound, (e.g., three to a phone)
- States can also correspond to specific contexts (e.g., "uh" with a left context of "b")
- States can just be repeated versions of same speech class – enforces minimum duration
- In practice state identities are learned

Temporal constraints

- Minimum duration from shortest path
- Self-loop probability vs. forward transition probability
- Transitions not in models not permitted
- Sometimes explicit duration models are used

Estimation of P(X|M)

- Given states, topology, probabilities
- Assume 1st order Markov
- For now, assume transition probabilities are known, emission probabilities estimated
- How to estimate likelihood function?
- Hint: at the end, one version will look like DTW

“Total” likelihood estimation

- i^{th} path through model M is Q_i
- N is the number of frames in the input
- X is the data sequence
- L(M) is the number of states in the model
- Then expand to

$$P(X \mid M) = \sum_{all\ Q_i\ in\ M\ of\ length\ N} P(Q_i, X \mid M)$$

- But this requires $O(N\ L(M)^N)$ steps
- Fortunately, we can reuse intermediate results

"Forward" recurrence (1)

- Expand to joint probability at last frame only

$$P(X \mid M) = \sum_{l=1}^{L(M)} P(q_l^N, X \mid M)$$

- Decompose into local and cumulative terms, where X can be expressed as X_1^N
- Using P(a,b|c)=P(a|b,c)P(b|c), get

$$P(q_l^n, X_1^n \mid M) = \sum_{k=1}^{L(M)} P(q_k^{n-1}, X_1^{n-1} \mid M) P(q_l^n, x_n \mid q_k^{n-1}, X_1^{n-1}, M)$$

"Forward" recurrence (2)

- Now define a joint probability for state at time *n* being q_l , and the observation sequence:

$$\alpha_n(l \mid M) = P(q_l^n, X_1^n \mid M)$$

- Then, restating the forward recurrence,

$$\alpha_n(l \mid M) = \sum_{k=1}^{L(M)} \alpha_{n-1}(k) P(q_l^n, x_n \mid q_k^{n-1}, X_1^{n-1}, M)$$

“Forward” recurrence (3)

- The “local” term can be decomposed further:

$$P(q_l^n, x_n \mid q_k^{n-1}, M) = P(q_l^n \mid q_k^{n-1}, X_1^{n-1}, M) P(x_n \mid q_l^n, q_k^{n-1}, X_1^{n-1}, M)$$

- But these are very hard to estimate. So we need to make two assumptions of conditional independence

Assumption 1: 1st order Markov

- State chain: state of Markov chain at time n depends only on state at time n-1, conditionally independent of the past

$$P(q_l^n \mid q_k^{n-1}, X_1^{n-1}, M) = P(q_l^n \mid q_k^{n-1}, M)$$

Assumption 2: conditional independence of the data

- Given the state, observations are independent of the past states and observations

$$P(x_n \mid q_l^n, q_k^{n-1}, X_1^{n-1}, M) = P(x_n \mid q_l^n, M)$$

“Forward” recurrence (4)

- Given those assumptions, the local term is

$$P(q_l^n \mid q_k^{n-1}, M)P(x_n \mid q_l^n, M)$$

- And the forward recurrence is

$$\alpha_n(l \mid M) = \sum_{k=1}^{L(M)} \alpha_{n-1}(k \mid M)P(q_l^n \mid q_k^{n-1}, M)P(x_n \mid q_l^n, M)$$

- Or, suppressing the dependence on M,

$$\alpha_n(l) = \sum_{k=1}^{L} \alpha_{n-1}(k)P(q_l^n \mid q_k^{n-1})P(x_n \mid q_l^n)$$

How do we start it?

- Recall definition

$$\alpha_n(l \mid M) = P(q_l^n, X_1^n \mid M)$$

- Set *n*=1

$$\alpha_1(l) = P(q_l^1, X_1^1) = P(q_l^1)P(x_1 \mid q_l^1)$$

How do we finish it?

- Recall definition

$$\alpha_n(l \mid M) = P(q_l^n, X_1^n \mid M)$$

- Sum over all states in model for n=N

$$\sum_{l=1}^{L} \alpha_N(l \mid M) = \sum_{l=1}^{L} P(q_l^N, X_1^N \mid M) = P(X \mid M)$$

Forward recurrence summary

- Decompose data likelihood into sum (over predecessor states) of product of local and global probabilities
- Conditional independence assumptions
- Local probability is product of emission and transition probabilities
- Global probability is a cumulative value
- A lot like DTW!

Forward recurrence vs DTW

- Terms are probabilistic
- Predecessors are model states, not observation frames
- Predecessors are always the previous frame
- Local and global factors are combined by product, not sum (but you could take the log)
- Combination of terms over predecessors are done by summation rather than finding the maximum (or min for distance/distortion)

Viterbi Approximation

- Summing very small products is tricky (numerically)
- Instead of total likelihood for model, can find best path through states
- Summation replaced by max
- Probabilities can be replaced by log probabilities
- Then summations can be replaced by min of the sum of negative log probabilities

$$-\log P(q_l^n, X_1^n) = \min_k[-\log P(q_k^{n-1}, X_1^{n-1}) - \log P(q_l^n \mid q_k^{n-1}) - \log P(x_n \mid q_l^n)]$$

Similarity to DTW

$$-\log P(q_l^n, X_1^n) = \min_k[-\log P(q_k^{n-1}, X_1^{n-1}) - \log P(q_l^n \mid q_k^{n-1}) - \log P(x_n \mid q_l^n)]$$

- Negative log probabilities are the distance! But instead of a frame in the reference, we compare to a state in a model, i.e.,

$$D(n, q_l^n) = \min_k[D(n-1, q_k^{n-1}) + d(n, q_l^n) + T(q_l^n, q_k^{n-1})]$$

- Note that we also now have explicit transition costs

Viterbi vs DTW

- Models, not examples
- The distance measure is now dependent on estimating probabilities – good tools exist
- We now have explicit way to specify priors
  - State sequences: transition probabilities
  - Word sequences: P(M) priors come from a *language model*

Assumptions required

- Language and acoustic model parameters are separable
- State chain is first-order Markov
- Observations independent of past
- Recognition via best path (best state sequence) is good enough – don't need to sum over all paths to get best model
- If all were true, then resulting inference would be optimum