Logistics

Reminder: class web links and infrastructure

- Check in with our web page (http://j.ee.washington.edu/~bilmes/classes/ee596a_winter_2013/) for up to date announcements, homeworks, etc.
- All homeworks will be due via our dropbox (https://catalyst.uw.edu/collectit/dropbox/bilmes/25379)
- All questions should be posted to our discussion board (https://catalyst.uw.edu/gopost/board/bilmes/31332)
- You can contact me anonymously if you wish via anonymous email (https://catalyst.uw.edu/umail/form/bilmes/4144)
Announcements, Assignments, and Reminders

- Recall, HW was due last Monday, 1/21. Last problem required ideas of elimination, OK to turn in/update this problem this evening (but no later).
- Extra office hour today, 3:45-4:45pm.
- Will still **either** have video lecture makeup class, or actual lecture makeup. Stay tuned to web page/email for announcements relevant to this.

Relevant Readings: Readings/handouts

- Readings are in a sub-directory “reading_drafts” directly below our web page (http://j.ee.washington.edu/~bilmes/classes/ee596a_winter_2013/).
- uid is this class name (lower case) and pwd are the quarter/year of the class.
- Note, the PDF file is password protected. Send me email if you have trouble (adobe reader should have no problems reading it).
The quantities we typically wish to compute for an HMM include:

- Compute $p(q_t|x_{1:t})$, or the **filtering** problem.
- Compute $p(q_t|x_{1:s})$, with $t > s$, or the **prediction** problem.
- Compute $p(q_t|x_{1:u})$, with $t < u$, or the **smoothing** problem.
- Above three named from linear systems literature in EE (e.g., Kalman filters).
- Note: above includes $p(q_t|x_{1:T})$ for $t \in \{1, 2, \ldots, T\}$.
- Also needed query is $p(q_t, q_{t+1}|x_r:s)$ (often $r = 1$ and $s = T$).
- In all above cases, we need to sum out hidden variables from joint distributions. E.g., $p(q_t|x_{1:T}) = p(q_t, x_{1:T})/p(x_{1:T})$, so also need $p(x_{1:T})$. I.e., we compute both the numerator and denominator in each of the above queries.
- Next few slides show how this relates to clique potentials in the HMM graph.
HMMs and elimination - forward recursion

From the last line of the elimination (when we sum out \( q_{r-1} \), we see \( \alpha \)-recursion, which is:

\[
\alpha_{t+1}(j) = \sum_i \alpha_t(i)p(Q_{t+1} = j|Q_t = i)p(x_{t+1}|Q_{t+1} = j) \tag{5.22}
\]

and

\[
\alpha_1(j) = p(Q_1 = j)p(\bar{x}_1|Q_1 = j) \tag{5.23}
\]

We have that

\[
\alpha_1(Q_1 = j) = p(Q_1 = j)p(\bar{x}_1|Q_1 = j) = p(\bar{x}_1, Q_1 = j), \quad \text{and}
\]

\[
\alpha_1(q_1) = p(\bar{x}_1, q_1).
\]

Also, \( \alpha_2(q_2) = \sum_{q_1} p(q_2|q_1)p(\bar{x}_2|q_2)\alpha(q_1) = \sum_{q_1} p(q_2|q_1)p(\bar{x}_2|q_2)p(\bar{x}_1, q_1) = \sum_{q_1} p(q_1, q_2, \bar{x}_1, \bar{x}_2) = p(\bar{x}_1, \bar{x}_2, q_2) \)

In general, the forward (\( \alpha \)) recursion has meaning

\[
\alpha_t(j) = p(x_{1:t}, Q_t = j)
\]

So \( \alpha \) (forward) recursion is just an instance of the elimination algorithm run on the GM for the HMM graph.

HMMs and elimination - backward recursion

\[
\beta_T(q_T) = 1 \tag{5.23}
\]

This corresponds to the \( \beta \) recursion

\[
\beta_t(q_t) = \sum_{q_{t+1}} \beta_{t+1}(q_{t+1})p(q_{t+1}|q_t)p(\bar{x}_{t+1}|q_{t+1}) \tag{5.22}
\]

\[
\beta_T(q_T) = 1
\]
HMMs and elimination

- So we have the forward recursion as an elimination order and the backwards recursion as an elimination order.

![Graph showing HMM structure]

Green order is \( \alpha \)-recursion, and blue order is \( \beta \)-recursion.

- Since HMM is a tree, there are no additional fill-in edges via the elimination orders we have chosen.

HMM, posteriors

- But as mentioned before, we want more than just \( p(x_{1:T}) \).
- We need clique posteriors \( \gamma_t(i) = p(Q_t = i|x) \) and \( \xi_t(i, j) = p(Q_{t-1} = i, Q_t = j|x) \). We can also get these from \( \alpha \) and \( \beta \).

\[
\gamma_t(q_t) = p(q_t|x_{1:T}) = p(x_{1:T}|q_t)p(q_t)/p(x_{1:T}) \\
= p(x_{1:t}, x_{t+1:T}|q_t)p(q_t)/p(x_{1:T}) \\
= p(x_{t+1:T}|q_t, x_{1:t})p(x_{1:t}|q_t)p(q_t)/p(x_{1:T}) \\
= p(x_{1:t}, q_t)p(x_{t+1:T}|q_t)/p(x) \\
= \alpha(q_t)\beta(q_t)/p(x) \\
= \alpha(q_t)\beta(q_t)/\sum q_t' \alpha(q_t')\beta(q_t')
\]
If a graph is not a tree, nodes can be clustered in such a way that the clusters (seen themselves as nodes) from “super nodes” in a tree-structure (junction tree), and then inference is run on a tree as normal.

For this to work, the tree-structure just satisfy the “running intersection property” — all cluster nodes \( C \in \mathcal{P} \) on the unique path \( \mathcal{P} \) between any two cluster nodes \( A, B \) must contain the the intersection between \( A \cap B \).

Hugin inference in Junction trees.

For more details, see 2011 lectures [http://j.ee.washington.edu/~bilmes/classes/ee512a_fall_2011/](http://j.ee.washington.edu/~bilmes/classes/ee512a_fall_2011/) (see lectures 14 and 15, in lecture 14 the section header entitled “Inference on JT”) and also see doc.pdf (described from pages 180—184).
HMM, and junction tree - right-state 3-clique

- Can view as junction tree with cliques/separators. One solution:
  - node a separator, & a 3-clique for a state & observation variable at time $t$, & another state to the right (& a final 2-clique at the end).

\[
\begin{align*}
Q_1 & \rightarrow Q_2 \rightarrow Q_3 \rightarrow Q_4 \rightarrow Q_5 \rightarrow Q_T \\
X_1 & \rightarrow X_2 \rightarrow X_3 \rightarrow X_4 \rightarrow X_5 \\
\end{align*}
\]

HMM, and junction tree - left-state 3-clique

- Another solution has each node a separator, and a 3-clique for a state and observation variable at time $t$, and another state to the left (and an initial 2-clique at the beginning).

\[
\begin{align*}
Q_1 & \leftarrow Q_2 \leftarrow Q_3 \leftarrow Q_4 \leftarrow Q_5 \leftarrow Q_T \\
X_1 & \leftarrow X_2 \leftarrow X_3 \leftarrow X_4 \leftarrow X_5 \\
\end{align*}
\]
HMM, and junction tree - two 2-cliques

- Yet another solution has cliques for successive states and for state/observation pair at each \( t \).

\[
\begin{align*}
Q_1 &\rightarrow Q_2 & Q_3 &\rightarrow Q_4 & Q_5 &\rightarrow Q_T \\
X_1 & & X_2 & & X_3 & & X_4 & & X_5 & & \cdots \\
& & & & & \cdots & & & & & \\
& & \cdots & & \cdots & & \cdots & & \cdots & & \\
& & Q_1 &\rightarrow Q_2 & Q_3 &\rightarrow Q_4 & Q_5 &\rightarrow Q_T \\
X_1 & & X_2 & & X_3 & & X_4 & & X_5 & & X_T
\end{align*}
\]

Discussion of the above three

- What are the advantages/disadvantages of the above (if any)?
- State space is the same.
- What are implications of summing over states observations if observation distribution is complex (e.g., if \( p(x|q) \) with \( x \) high-dimensional real-valued Gaussian mixture).
Using either of these junction trees, we can define a forward and/or backwards recursion, and we’ll again, in some cases, either recover exactly the $\alpha, \beta$ recursions, or computations that are quite similar.

For example, Hugin message passing computes the backward messages based on the forward (if we go right and then left), or computes the forward messages based on the backwards (if we go left and then right).

- Clique marginals are needed for standard learning procedures (EM and gradient calculation)
- Forward ($\alpha$) recursion is elimination from left-to-right (equivalently, just LBP message passing)
- Backward ($\beta$) recursion is elimination from right-to-left (again also LBP)
- “Inward” inference also possible via alternating elimination orders.
- HMM as a junction tree, various options for the cliques without increasing state space.
HMMs
Trellis
Other HMM queries
MPE
Sampling
What HMMs can do
Summary
Scratch

HMM, other recursions

It is possible to derive a temporal recursion for quantities other than $\alpha$ and $\beta$. E.g., here is a $\gamma_t(j) = p(Q_t = j|x_{1:T})$ backwards recursion.

$$\gamma(q_t) = \sum_{q_{t+1}} p(q_t, q_{t+1}|x_{1:T}) = \sum_{q_{t+1}} p(q_t|q_{t+1}, x_{1:T}) p(q_{t+1}|x_{1:T})$$

$$= \sum_{q_{t+1}} p(q_t|q_{t+1}, x_{1:t}) \gamma(q_{t+1}) = \sum_{q_{t+1}} p(q_t|q_{t+1}, x_{1:t}) \gamma(q_{t+1})$$

$$= \sum_{q_{t+1}} \frac{p(q_t, q_{t+1}, x_{1:t})}{\sum_{q_t} p(q_t, q_{t+1}, x_{1:t})} \gamma(q_{t+1})$$

$$= \sum_{q_{t+1}} \frac{p(q_{t+1}|q_t) p(q_t, x_{1:t})}{\sum_{q_t} p(q_{t+1}|q_t) p(q_t, x_{1:t})} \gamma(q_{t+1})$$

$$= \sum_{q_{t+1}} \frac{p(q_{t+1}|q_t) \alpha_t(q_t)}{\sum_{q_t} p(q_{t+1}|q_t) \alpha_t(q_t)} \gamma(q_{t+1})$$

Therefore, there is a backwards pass recursion using just the $\alpha$’s without directly touching the observations again (better memory).

HMM Trellis/Lattice/Grid

- We'll see that it is very useful to view the HMM state-space as a trellis.
- It is sometimes called an HMM grid, or even a “lattice” although it should not be confused with a Birkhoff-style lattice in mathematics (which is quite different).
- Key thing about a trellis representation is that it shows the Markov transition matrix but time is also explicitly represented.
- Lets say there are $|D_Q| = N$ states and $T$ time steps.
- Then the trellis is an (often rectangular) $N \times T$ grid.
A trellis for words

A trellis really corresponds to a language — any string generated by the trellis corresponds to a sentence in the language. Note that the language “words” are listed on the edges.

A trellis for peptides

List of peptides
ACDEEMQRSTGFQMD
ACDEEMQRSTMPRAIFQMD
ACDEEMQRSTLLADSTCFQMD
ACDEAQSMPHQMPPRAIFQMD
ACDEAQSMPHQLLADSTCFQMD
ACDEAQSAFQWMRSTCFQMD

Lattice (a partial ordering of peptide sub-segments) for the strings on the left.

Peptides are fragments of proteins. Lattices can be used for this too.
Consider the following four-state state-transition diagram for a 1st order Markov chain.

- when seen as an HMM, this shows only the Markov chain but not the observation distributions.
- Also, this doesn’t show the “data flow” of the computation of the $\alpha$-recursion.
- It is possible to use yet another graph (a trellis graph) that can show this.

On the left we have the trellis (4 states and $T$ time frames).
- on the right we see the state transition diagrams.
- Both are graphs (but display very different information), the left graph shows the allowable state transitions at each $t \in \{1, 2, \ldots, T\}$.
- Left (trellis) would depict a time-inhomogeneous Markov chain if edges are different at each time step, or if the transition probabilities for the same pattern changes — edges indicate potential non-zero values, lack of edges indicate a zero transition value.
Informally, we can annotate each state with an observation distribution of the form \( p(x|Q = j) = b_j(x) \).

The trellis too can be seen with observation distributions:

- Time homogeneity is indicated by transition probabilities being the same.
- Output distributions are dependent on state, and observation at time \( t \).
HMM Trellis with Output Distributions

The trellis too can be seen with observation distributions:

\[ b_{q4}(x_1) \quad p(q_1|q_4) \quad b_{q4}(x_2) \quad p(q_1|q_4) \quad b_{q4}(x_3) \quad p(q_1|q_4) \quad b_{q4}(x_4) \]

\[ b_{q3}(x_1) \quad p(q_1|q_3) \quad b_{q3}(x_2) \quad p(q_1|q_3) \quad b_{q3}(x_3) \quad p(q_1|q_3) \quad b_{q3}(x_4) \]

\[ b_{q2}(x_1) \quad p(q_1|q_2) \quad b_{q2}(x_2) \quad p(q_1|q_2) \quad b_{q2}(x_3) \quad p(q_1|q_2) \quad b_{q2}(x_4) \]

\[ b_{q1}(x_1) \quad p(q_1|q_1) \quad b_{q1}(x_2) \quad p(q_1|q_1) \quad b_{q1}(x_3) \quad p(q_1|q_1) \quad b_{q1}(x_4) \]

\[ \alpha_t(j) = \sum_i \alpha_t(i)p(j|i)p(x_t|j) = p(x_t|j) \sum_i \alpha_t(i)p(j|i) \quad \text{and} \quad \alpha_1(j) = p(j)p(x_1|j) \]
HMM Trellis with and the backward recursion

Consider the beta recursion computation in the context of a trellis:

\[
\beta_t(q_t) = \sum_{q_{t+1}} \beta_{t+1}(q_{t+1}) p(q_{t+1}|q_t) p(\bar{x}_{t+1}|q_{t+1}) \quad \text{and} \quad \beta_T(q_T) = 1
\]

A few other points:

- Notice that the computation proceeds synchronously.
- Notice that at each node at each time \(t\), we in a sense “pull” values from previous times \(t\) (for \(\alpha\) computation).
- Notice that the complexity depends on the density of the transition matrix. If strictly left-to-right (i.e., \(p(Q_t = j|Q_{t-1} = i) > 0\) iff \(j \in \{i, i + 1\}\)), then computation is only \(O(TN)\).
- The trellis will be quite useful to describe other inference algorithms and strategies and an HMM’s structured generalizations as well.
Filtering

- Filtering query is \( p(q_t|x_{1:t}) \) for \( t \in \{1, 2, \ldots, T\} \).
- If we need this for one particular \( t \), this is identical to one of the posteriors we already have \( p(q_T|x_{1:T}) \).
- More likely, we need recursion from \( t \) to \( t + 1 \).
- This is obtained immediately from \( \alpha \)-recursion, since \( \alpha_q(t) = p(x_{1:t}, Q_t = q) \), we have

\[
p(q_t|x_{1:t}) = \frac{\alpha_q(t)}{\sum_{q'_t} \alpha_{q'_t}(t)} \tag{5.1}
\]

- So, normalized \( \alpha \)s are just the filtering operation.

Prediction

- \( p(q_t|x_{1:s}) \), with \( t > s \).
- This is also easily obtained from the \( \alpha \)s for we have

\[
p(q_t, x_{1:s}) = \sum_{q_s, q_{s+1}, \ldots, q_{t-1}} p(q_t, q_{t-1}, \ldots, q_{s+1}, q_s, x_{1:s}) \tag{5.2}
\]

\[
= \sum_{q_s, q_{s+1}, \ldots, q_{t-1}} p(q_t|q_{t-1}) p(q_{t-1}|q_{t-2}) \cdots p(q_{s+1}|q_s) \alpha_{q_s}(s) \tag{5.3}
\]

\[
= \sum_{q_{t-1}} p(q_{t-1}) \sum_{q_{t-2}} (q_{t-1}|q_{t-2}) \cdots \sum_{q_s} p(q_{s+1}|q_s) \alpha_{q_s}(s) \tag{5.4}
\]

- And then we just normalize.
**Prediction**

- Thus, this is like running elimination (or message passing) on a graph where some of the observations are removed, as in:

  ![Graph](image)

  for computing \( p(q_5 | x_{1:3}) \).

**Smoothing**

- \( p(q_t | x_{1:u}) \), with \( t < u \).
- If we need this for one particular \( t \), this is identical to one of the posteriors we already have \( p(q_t | x_{1:T}) \).
- Otherwise, we might need to update \( p(q_t | x_{1:u}) \) for each \( t < u \) as \( u \) increases (observations come in).

  ![Graph](image)

  Includes re-sending messages (red). Each step requires \( O(T) \) additional messages, in total \( O(T^2) \) computation.
HMMs

Trellis

Other HMM queries

MPE

Sampling

What HMMs can do

Summary

Scratch

Edge Marginals

- Need $p(q_t, q_{t+1}|x_1:T)$ for learning tasks
- Easy to obtain with both the $\alpha$ and $\beta$ quantities, since:

$$p(q_{t-1}, q_t|x_1:T) = \frac{p(q_{t-1}, q_t, x_1:T)}{p(x_1:T)}$$  \hspace{1cm} (5.5)

and

$$p(q_{t-1}, q_t, x_1:T)$$

$$= p(x_t|q_t)p(q_{t-1}, q_t, x_1:t-1, x_{t+1:T})$$  \hspace{1cm} (5.6)

$$= p(x_t|q_t)p(x_{t+1:T}|q_t)p(q_t, q_{t-1}, x_1:t-1)$$ \hspace{1cm} (5.7)

$$= p(x_t|q_t)\hat{\beta}_t(q_t)p(q_t, q_{t-1}, x_1:t-1)p(q_{t-1}, x_1:t-1)$$ \hspace{1cm} (5.8)

$$= p(x_t|q_t)\hat{\beta}_t(q_t)p(q_t|q_{t-1})\alpha_{t-1}(q_{t-1})$$ \hspace{1cm} (5.9)

$$= p(x_t|q_t)\hat{\beta}_t(q_t)p(q_t|q_{t-1})\alpha_{t-1}(q_{t-1})$$ \hspace{1cm} (5.10)

So all of the edge marginals can be computed using the standard ($\alpha$ and $\beta$) recursions.

There are several aspects of HMMs we will discuss: 1) Viterbi decoding/MPE, 2) sampling, 3) how flexible are HMMs, 4) real-world inference in HMMs (what to do when state space gets large), 5) time-space tradeoffs.
Crucial problem in HMMs is to solve MPE (also called Viterbi decoding):

\[ q_{1:T}^* \in \arg\max_{q_{1:T} \in \mathcal{D}_{Q_{1:T}}} p(\bar{x}_{1:T}, q_{1:T}) \]  (5.11)

Note that computing the value of the max can be done just with an alternate to the \( \alpha \)-recursion. Since

\[
\max_{q_{1:T} \in \mathcal{D}_{Q_{1:T}}} p(\bar{x}_{1:T}, q_{1:T}) = \max_{q_{1:T} \in \mathcal{D}_{Q_{1:T}}} \prod_{t} p(\bar{x}_t | q_t) p(q_t | q_{t-1})
\]  (5.12)

\[
= \max_{q_T} p(x_T | q_T) \ldots \left( \max_{q_2} p(\bar{x}_2 | q_2) p(q_3 | q_2) \left( \max_{q_1} p(\bar{x}_1 | q_1) p(q_2 | q_1) \right) \right)
\]  (5.13)

\[
= \max_{q_T} p(x_T | q_T) \ldots \left( p(x_3 | q_3) \max_{q_2} p(q_3 | q_2) \left( p(\bar{x}_2 | q_2) \max_{q_1} p(q_2 | q_1) (p(\bar{x}_1 | q_1)) \right) \right)
\]  (5.14)

**HMM (Viterbi) Decoding**

- Why is it called (Viterbi) decoding?
- Source-channel model of communications (from Information Theory)

Consider the source being generated by Markov chain, and the “channel” being each symbol corrupted by some channel noise (observation distribution).
Most Probable Explanation

- We can thus define a modified form of the $\alpha$-recursion that, rather than uses summation, uses a max operator.

\[
\alpha^m_q(1) = p(\bar{x}_1|Q_1 = q) \tag{5.15}
\]
\[
\alpha^m_q(t) = p(\bar{x}_t|Q_t = q) \max_r p(Q_t = q|Q_{t-1} = r) \alpha^m_r(t-1) \tag{5.16}
\]

- We get the final max can be computed from the final max marginal:

\[
\max_{q_1:T \in D, q_1:T} p(\bar{x}_{1:T}, q_{1:T}) = \max_q \alpha^m_q(T) \tag{5.17}
\]

- Max operator is similar to sum (a commutative semi-ring), in that it marginalizes out hidden variables.

Max operator is similar to sum, in that it marginalizes out hidden variables.

- Given $p(\bar{x}, q)$ when we form $\max_q p(\bar{x}, q)$ we can think of this as a marginal, the “max marginal” of the form

\[
p^m(\bar{x}) = \max_q p(\bar{x}, q) \tag{5.18}
\]

- Given this, we can view the $\alpha^m_q(t)$ as the max marginals up to time $t$

\[
\alpha^m_q(t) = p^m(x_{1:t}, Q_t = q) \tag{5.19}
\]

so that the above final maximization makes sense. We’ve defined a recursive way to compute the max marginal.

- From EE512: dynamic programming works on any commutative semi-ring, we’re just defining the $\alpha$ recursion using the max-product semi-ring rather than the previous sum-product semi-ring.
MPE - Viterbi Path

- But this computes only the value, how to get the actual states?
- Will need to do a forward-backward pass, like $\alpha, \beta$.
- $\arg\max$ also distributes in a fashion. The true max at time $t$ will depend on what the true max at time $t + 1$ is.
- We can pre-compute the max for all $q$ at time $t$ when going forward, and then when going backwards, once we know the true max at time $t + 1$, we backtrack and then used the previously computed max at time $t$.
- Repeating this from $T$ back to 1 we’ve got the MPE.

MPE - Viterbi Path Computation

$$\arg\max_{q_1:T} p(\bar{x}_{1:T}, q_{1:T}) = \arg\max_{q_1:T} \prod_t p(\bar{x}_t | q_t) p(q_t | q_{t-1})$$

$$= \arg\max_{q_T} p(x_T | q_T) \ldots \left( \arg\max_{q_2} p(\bar{x}_2 | q_2) p(q_3 | q_2) \left( \arg\max_{q_1} p(\bar{x}_1 | q_1) p(q_2 | q_1) \right) \right)$$

$$= \arg\max_{q_T} p(x_T | q_T) \ldots \left( p(x_3 | q_3) \arg\max_{q_2} p(q_3 | q_2) \left( p(\bar{x}_2 | q_2) \arg\max_{q_1} p(q_2 | q_1) p(\bar{x}_1 | q_1) \right) \right) \ldots$$

- So inner most $\arg\max$ depends on true max for $q_2$. Next inner-most $\arg\max$ depends on $q_3$, and so on.
- We define a recursion that stores these integer state indices based on max marginal.

$$\hat{\alpha}^m_q(t) \in \arg\max_r p(Q_t = q | Q_{t-1} = r) \alpha^m_r(t-1) \quad (5.20)$$

- Note that this is integer index, not a score.
MPE

- We can then compute Viterbi path by backtracking, which is entirely a deterministic process using index lookup (except for the initial case where we find the maximum state).

\begin{verbatim}
1 Compute \( q^*_T \in \arg\max_q \alpha^m_q(T) \)
2 \textbf{for} \( t = T \ldots 2 \) \textbf{do}
3 \quad \textbf{Set} \( q^*_{t-1} \leftarrow \hat{\alpha}^m_{q^*_t}(t) \)
\end{verbatim}

MPE - summary

Forward Equations

\[
\begin{align*}
\alpha^m_q(1) &= p(\bar{x}_1|Q_1 = q) \quad (5.21) \\
\alpha^m_q(t) &= p(\bar{x}_t|Q_t = q) \max_r p(Q_t = q|Q_{t-1} = r)\alpha^m_r(t - 1) \quad (5.22)
\end{align*}
\]

And the forward equation for storing the back indices:

\[
\hat{\alpha}^m_q(t) \in \arg\max_r p(Q_t = q|Q_{t-1} = r)\alpha^m_r(t - 1) \quad (5.23)
\]

Backward algorithm, to compute the Viterbi path

\begin{verbatim}
1 Compute \( q^*_T \in \arg\max_q \alpha^m_q(T) \)
2 \textbf{for} \( t = T \ldots 2 \) \textbf{do}
3 \quad \textbf{Set} \( q^*_{t-1} \leftarrow \hat{\alpha}^m_{q^*_t}(t) \)
\end{verbatim}
MPE on the HMM Trellis

The trellis is also useful to view MPE/Viterbi, and is the reason it is sometimes called “Viterbi path.”

Atypical explanations

- E.g., the 1st best might not be a good reflection of the typical states that explain the observation.
- Simple example of “typicality”, flip a $P(H) = 0.9 = 1 - P(T)$ biased coin 100 times. Most probable sequence is 100 heads. A typical sequence will be one that has about 90 heads and 10 tails. Much more likely we’ll get one of the 90,10 sequences than the all heads sequence.
- Viterbi path is potentially giving us an atypical sequence.
- Two solutions: sample from the posterior distribution $p(q_1:T|x_1:T)$, or computing the $n$-best sequences.
HMMs
Trellis
Other HMM queries
MPE
Sampling
What HMMs can do
Summary
Scratch

Posterior sampling - recursion

- We wish to draw samples of the form \( q_{1:T} \sim p(q_{1:T} | \bar{x}_{1:T}) \) for observation sequence \( \bar{x}_{1:T} \).
- In general, to sample from \( p(a, b) \) can sample from \( p(a) \) and then from \( p(b|a) \), for sample \( a \) initially drawn.
- We can sample from \( p(q_1 | \bar{x}_{1:T}) \) and then sample from \( p(q_2 | q_1, \bar{x}_{1:T}) \), and thus from \( p(q_t | \bar{q}_{1:t-1}, \bar{x}_{1:T}) \)
- \( p(q_1 | \bar{x}_{1:T}) \propto \alpha_1(q_1) \beta_1(q_1) = p(q_1, \bar{x}_{1:T}) \) so this is readily available.
- To get \( p(q_2 | \bar{q}_1, \bar{x}_{1:T}) \) we can compute

\[
p(q_2, \bar{q}_1, \bar{x}_{1:T}) = p(\bar{x}_2 | q_2) \beta_2(q_2) p(q_2 | \bar{q}_1) \alpha_1(\bar{q}_1) \tag{5.24}
\]

- To get \( p(q_t | \bar{q}_{1:t-1}, \bar{x}_{1:T}) \), construct \( p(q_t, \bar{q}_{1:t-1}, \bar{x}_{1:T}) \) and recursion for \( \alpha \)-like quantity

\[
p(q_t, \bar{q}_{1:t-1}, \bar{x}_{1:T}) = p(\bar{x}_t | q_t) \beta_t(q_t) p(q_t | \bar{q}_{t-1}) p(\bar{q}_{1:t-1}, \bar{x}_{1:t-1}) \tag{5.25}
\]

Posterior sampling - recursion

- Is computing the r.h.s. \( p(\bar{q}_{1:t}, \bar{x}_{1:t}) \) for various \( t \) easy?
- Yes, everything is essentially observed and factorizes, so we keep multiplying new observed values for resulting computed sample from left-to-right, as in:

\[
p(\bar{q}_{1:t}, \bar{x}_{1:t}) = p(\bar{x}_t | q_t) p(\bar{q}_t | \bar{q}_{t-1}) p(\bar{q}_{1:t-1}, \bar{x}_{1:t-1}) \tag{5.26}
\]
What HMMs can do

- HMMs are more powerful than you might think.
- In fact, many DGMs can be represented by HMMs.
- Thus, before studying DGMs, it is worthwhile to understand how flexible and powerful HMMs are (and then as we go through course, we’ll see what the penalties are for making such HMM representations).
- We next visit a set of properties about HMMs that should be remembered.

Observations are not i.i.d.

- Joint probability under an HMM:
  \[ p(X_{t:t+h} = x_{t:t+h}) \]
  \[ = \sum_{q_{t:t+h}} \prod_{j=t}^{t+h} p(X_j = x_j | Q_j = q_j) a_{q_{j-1}q_j}. \]
- Unless only one state is possible, observations do not factorize.
- In an HMM, there are no statements of the form \( A \perp \perp B \).
- HMMs have conditional independence properties (like all DGMs have).

\[ p(X_{t:t+h} = x_{t:t+h}|Q_{t:t+h} = q_{t:t+h}) \]
\[ = \prod_{\tau=t}^{t+h} p(X_\tau = x_\tau | Q_\tau = q_\tau). \]
The Viterbi path (most-probable explanation) of an HMM is defined:

$$q^*_{1:T} \in \arg\max q_{1:T} p(X_{1:T} = x_{1:T}, q_{1:T})$$ (5.28)

This is a standard method for finding a mapping from observations $$\bar{x}_{1:T}$$ to candidate answers $$q^*_{1:T}$$, also called a “decoding.”

Other times, we might have $$k$$ HMMs, corresponding to $$k$$ classes

$$\{p_k(x_{1:T}, q_{1:T}) : k = 1 \ldots K\}$$ (5.29)

and we wish to perform classification amongst the $$K$$ discrete set of classes as follows

$$(k^*, q^*_{1:T}) = \arg\max_{q_{1:T}, k} p_k(x_{1:T}, q_{1:T})$$ (5.30)

and $$k^*$$ becomes part of the hypothesized answer.
Observations are not “Viterbi i.i.d.”

- We can view this as $K$ distributions over just the observations, i.e.,
  
  $$p^k_{\text{vit}}(x_{1:T}) \propto p_k(x_{1:T}, q^*_1:T)$$  
  $$\propto \max_{q_1:T} p_k(x_{1:T}, q_1:T)$$ (5.31)  
  $$\propto \max_{q_1:T} \prod_{t=1}^{T} p_k(x_t|q_t)p_k(q_t|q_{t-1})$$ (5.32)  
  $$\propto \max_{q_1:T} \prod_{t=1}^{T} p_k(x_t|q_t)p_k(q_t|q_{t-1})$$ (5.33)

- Note that this max-marginal can be completely different than the standard summation marginal
  
  $$\sum_{q_1:T} \prod_{t=1}^{T} p_k(x_t|q_t)p_k(q_t|q_{t-1})$$ (5.34)

- This is just a different semi-ring. The resulting distribution over observations does not in general factorize, so no i.i.d. here either. The distribution $p^k_{\text{vit}}(x_{1:T})$ also does not in general have any (marginal) independence assumptions.

HMMs and stationarity

- HMM is a stochastic process over $\{X_t\}$
- An HMM is stationary whenever
  
  $$p(X_{t_1+h:n+h} = x_{1:n}) = p(X_{t_1:n} = x_{1:n})$$ (5.35)

  When might this be the case?
- We have
  
  $$p(X_{t_1:n+h} = x_{1:n})$$
  $$= \sum_{q_1:n} p(X_{t_1:n+h} = x_{1:n}, Q_{t_1:n+h} = q_1:n)$$
  $$= \sum_{q_1:n} p(Q_{t_1+h} = q_1)p(X_{t_1+h} = x_1|Q_{t_1+h} = q_1)$$
  $$\prod_{i=2}^{n} p(X_{t_i+h} = x_i|Q_{t_i+h} = q_i)p(Q_{t_i+h} = q_i|Q_{t_{i-1}+h} = q_{i-1})$$
HMMs and stationarity (cont.)

\[
\sum_{q_1} p(Q_{t_1+h} = q_1)p(X_{t_1+h} = x_1|Q_{t_1+h} = q_1) = \sum_{q_1} p(Q_{t_1} = q_1)p(X_{t_1} = x_1|Q_{t_1} = q_1) \quad (5.36)
\]

\[
\sum_{q_2} \prod_{i=2}^n p(X_{t_i+h} = x_i|Q_{t_i+h} = q_i)p(Q_{t_i+h} = q_i|Q_{t_{i-1}+h} = q_{i-1}) = \sum_{q_1} p(Q_{t_1+h} = q_1)p(X_{t_1+h} = x_1|Q_{t_1+h} = q_1) \quad (5.37)
\]

\[
\sum_{q_2} \prod_{i=2}^n p(X_{t_i} = x_i|Q_{t_i} = q_i)p(Q_{t_i} = q_i|Q_{t_{i-1}} = q_{i-1}) = \sum_{q_1} p(Q_{t_1+h} = q_1)p(X_{t_1} = x_1|Q_{t_1} = q_1)f(x_{2:n}, q_1) \quad (5.38)
\]

where \( f(x_{2:n}, q_1) \) is a function that is independent of the variable \( h \).

- For HMM stationarity to hold, it is required that 
  \( p(Q_{t_1+h} = q_1) = p(Q_{t_1} = q_1) \) for all \( h \).

HMMs stationarity depends on MC

- Therefore, the HMM’s stationarity condition is entirely determined by the stationarity condition of the underlying hidden Markov chain.
- Consider the way in which HMMs are often used:
  - Long chains
  - Chains with cycle transition matrices
  - Chains with upper-triangular matrices
  - Chains with strictly left-to-right transitions. Ex: speech recognition

<table>
<thead>
<tr>
<th>Word</th>
<th>Pronunciation</th>
<th>Word</th>
<th>Pronunciation</th>
</tr>
</thead>
<tbody>
<tr>
<td>cat</td>
<td>[kæt]</td>
<td>goose</td>
<td>[gus]</td>
</tr>
<tr>
<td>cats</td>
<td>[kæts]</td>
<td>geese</td>
<td>[gis]</td>
</tr>
<tr>
<td>pig</td>
<td>[pɪɡ]</td>
<td>hedgehog</td>
<td>[ˈhɛdɡ hoɡ]</td>
</tr>
<tr>
<td>pigs</td>
<td>[pɪɡz]</td>
<td>hedgehogs</td>
<td>[ˈhɛdɡ hɔɡz]</td>
</tr>
<tr>
<td>fox</td>
<td>[fɔks]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>foxes</td>
<td>[fɔkses]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Hence, in only rare cases, when HMMs are used, are they stationary stochastic processes.
Gaussian Mixture HMM

- One of the most widely used HMMs in practice is one where the observation distributions are Gaussian mixtures, where

$$p(x|q) = \sum_c p(x|q,c)p(c|q)$$  \hspace{1cm} (5.39)$$

$$= \sum_m \mathcal{N}(x|\mu_m,\Sigma_m) c_m$$  \hspace{1cm} (5.40)$$

and where

$$\mathcal{N}(x|\mu,\Sigma) = \frac{1}{|2\pi\Sigma|^{d/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right)$$  \hspace{1cm} (5.41)$$

- The HMM BN becomes

![HMM BN diagram](image)

Correlated & Covariance

- Correlation between two real random vectors $X$ and $Y$

$$\text{cor}(X, Y) = E[XY^T]$$  \hspace{1cm} (5.42)$$

- Covariance between two real random vectors $X$ and $Y$

$$\text{cov}(X, Y) = E[(X - E(X))(Y - E(Y))^T] = E[XY^T] - E[X]E[Y^T]$$  \hspace{1cm} (5.43)$$

- $X$ and $Y$ are uncorrelated if $\text{cov}(X, Y) = 0$.

- $\text{cov}(X, Y) = \text{cor}(X, Y)$ if either the means are zero.

- If $X \perp \! \! \! \! \! \! \! \! \! \! \! \! \! \! \! \! \! \! \! \! \! \! \! \! Y$ then $\text{cov}(X, Y) = 0$ but vice versa only if $X, Y$ are jointly Gaussian.

- $\text{cov}(X, Y) = 0$ is indication of lack of linear dependence, and hint there might not be strong dependence at all.
Consider single-component Gaussian HMM (i.e., for each state, the observation distribution is a single multivariate Gaussian).

Assume that the Markov chain is currently stationary, with stationary distribution $\pi$.

What about $\text{cov}(X_t, X_{t+h})$? Is it zero? How does it decay?

Computing $E[X_t]$

$$EX_t = \int xp(X_t = x)dx = \int x \sum_i p(X_t = x|Q_t = i)\pi_i dx$$

$$= \sum_i E[X_t|Q_t = i]\pi_i = \sum_i \mu_i \pi_i$$

so this is just the weighted sums of means, where the weights are the stationary distribution probabilities.

Similarly, let's compute the correlation:

$$E[X_tX_{t+h}^\top]$$

$$= \int xy^\top p(X_t = x, X_{t+h} = y)dxdy$$

$$= \int xy^\top \sum_{ij} p(X_t = x, X_{t+h} = y|Q_t = i, Q_{t+h} = j)$$

$$p(Q_{t+h} = j|Q_t = i)\pi_i dxdy$$

$$= \sum_{ij} E[X_tX_{t+h}^\top|Q_t = i, Q_{t+h} = j](A^h)_{ij}\pi_i$$

The above equations follow from $p(Q_{t+h} = j|Q_t = i) = (A^h)_{ij}$ (i.e., the Chapman-Kolmogorov equations) where $(A^h)_{ij}$ is the $i,j^{th}$ element of the matrix $A$ raised to the $h$ power.
Because of the HMM conditional independence properties, we have

\[ E[X_t X_{t+h}^\top | Q_t = i, Q_{t+h} = j] = E[X_t | Q_t = i] E[X_{t+h}^\top | Q_{t+h} = j] = \mu_i \mu_j^\top \] (5.51)

yielding

\[ E[X_t X_{t+h}^\top] = \sum_{ij} \mu_i \mu_j^\top (A^h)_{ij} \pi_i \] (5.53)

Covariance is therefore:

\[ \text{cov}(X_t, X_{t+h}) = \sum_{ij} \mu_i \mu_j^\top (A^h)_{ij} \pi_i - \left( \sum_i \mu_i \pi_i \right) \left( \sum_i \mu_i \pi_i \right)^\top \] (5.54)

Therefore, we have

\[ \text{cov}(X_t, X_{t+h}) \overset{h}{\rightarrow} \sum_{ij} \mu_i \mu_j^\top (A^h)_{ij} \pi_i - \left( \sum_i \mu_i \pi_i \right) \left( \sum_i \mu_i \pi_i \right)^\top = 0 \] (5.56)

Thus, while the covariance between to observations is not necessarily zero in an HMM, once we are at a stationary distribution, this covariance goes to zero exponentially fast.

**Exercise:** For this property to hold, must we assume a unique stationary distribution here?
Correlation over time of simple HMM

- Example of the decay in the mutual-information correlation from a real-world HMM. I.e., we see \( f(\tau) = I(X_t; X_{t+\tau}) \), where \( I() \) is the mutual information function.
- Mutual information is stronger than correlation.
- This is compared against i.i.d. samples (high peak at \( \tau = 0 \) is expected).

![Graph showing decay in mutual information correlation](image)

State Duration Modeling

- Markov chain, state duration distribution is geometric
- Let \( D \) be such a random variable, then
  \[
  P(D = d) = p^{d-1}(1 - p)
  \]
  \[\text{(5.57)}\]
  where \( d \geq 1 \) is an integer and \( p = a_{ii} \), if \( D \) is the duration random variable for state \( i \) of the chain, giving:

![Graph showing geometric distribution for state duration](image)

- Many sequential tasks have sub-segments that do not follow this distribution.
State Duration Modeling

- Many “tricks” to using an HMM, can alleviate such problems.
- “state-tying”, where multiple states have the same observation distribution (parameters are shared).
- That is, state $q$ and $q'$ are tied if it is the case that
  \[ p(x|Q_t = q) = p(x|Q_t = q') \quad \forall x \in D_X \]  \hspace{1cm} (5.58)
- If $n$ states in a series are strung together, all of which share the same observation distribution, that observation distribution will be active for as long as we are in that state. For example:

\[
\begin{array}{ccccccc}
0.75 & 0.25 & 0.75 & 0.25 & 0.75 & 0.25 & 0.75
\end{array}
\]

This corresponds to the sum of random variables. Let \( \{D_i\}_i \) be a collection of independent geometrically distributed random variables with parameter $p$, and let $W_r = \sum_{i=1}^r D_i$, then

\[
p(W_r = k) = \binom{k-1}{r-1} p^r (1-p)^{k-r}, \quad k = r, r+1, \ldots \]  \hspace{1cm} (5.59)
- This is a “negative binomial distribution”:

\[
\begin{array}{ccccccc}
0.08 & 0.06 & 0.04 & 0.02 & 0.00 & 0.00 & 0.00
\end{array}
\]
State Duration Modeling

- If we have multiple parallel states in series, all of which share the same observation distribution, we can construct much more interesting (multimodal) distributions.
- For example, the following left chain will have distribution as shown on the right (a mixture of negative binomial distributions).

Other examples: very long chains, ladders, fixed-length distributions (histograms), and so on.
- HMMs can have flexible distributions, cost of extra states.

What HMMs can do - summary so far

- Observations are not i.i.d., but conditioned on state variables, they are independent.
- Observations are not “Viterbi i.i.d.”
- HMMs are a stationary process over $p(x_{1:n})$ whenever the underlying hidden Markov chain is a stationary process.
- Single Gaussian per state HMM: Covariance decays as:

$$\begin{align*}
\text{cov}(X_t, X_{t+h}) &= \sum_{i,j} \mu_i \mu_j^\top (A^h)_{ij} \pi_i - \left( \sum_i \mu_i \pi_i \right) \left( \sum_i \mu_i \pi_i \right)^\top \\
&\overset{h \to \infty}{\longrightarrow} \sum_{i,j} \mu_i \mu_j^\top \pi_j \pi_i - \left( \sum_i \mu_i \pi_i \right) \left( \sum_i \mu_i \pi_i \right)^\top = 0
\end{align*}$$
but mutual information (in practice) can apparently extend in time reasonably far (but also decays).

Parameter sharing means enormous flexibility in state duration models (e.g., negative binomial, mixtures thereof, fixed histograms).

---

HMMs Generative Accuracy

- We can view an HMM as an approximate generative distribution of the observation variables, as in \( p_h(x_{1:T}) \approx p(x_{1:T}) \)
- Given that \( p_h \) is an approximation, one that is a mixture

\[
p_h(x_{1:T}) = \sum_{q_{1:T}} p_h(x_{1:T}, q_{1:T})
\]

(5.60)

what can we say about \( p_h \) and its accuracy?

- Accuracy can be measured by KL-divergence

\[
D(p(x_{1:T}) || p_h(x_{1:T})) = \sum_{x_{1:T}} p(x_{1:T}) \log \frac{p(x_{1:T})}{p_h(x_{1:T})}
\]

(5.61)

and if \( D(p(x_{1:T}) || p_h(x_{1:T})) = 0 \), then the HMM is perfectly generatively accurate.
For an HMM to be generatively accurate, we can derive necessary conditions on the HMM, e.g., number of required states.

Recall, \( n^{th} \)-order Markov chain convertible to 1st-order one.

If \( D(p(x_{1:T})||p_h(x_{1:T})) = 0 \), then the following mutual information quantities must be equal

\[
I(X_{S_1};X_{S_2}) = I_h(X_{S_1};X_{S_2})
\]

and where \( S_1, S_2 \subseteq 1 : T \), and where \( I(\cdot;\cdot) \) is true mutual information, and \( I_h(\cdot;\cdot) \) is the mutual information under the HMM

\[
I_h(X_{S_1};X_{S_2}) = \sum_{x_{S_1} \cup x_{S_2}} p_h(x_{S_1},x_{S_2}) \log \frac{p_h(x_{S_1},x_{S_2})}{p_h(x_{S_1})p_h(x_{S_2})}
\]

Define \( X_{-t} \triangleq \{ X_1, X_2, \ldots, X_{t-1}, X_{t+1}, \ldots, X_T \} \) (i.e., \( -t \) is set of all indices sans \( t \)).
Proof: HMMs Generative Accuracy

Proof.

Accurate HMM (i.e., zero KL-divergence from true distribution) implies
\[ I(X_{-t}; X_t) = I_h(X_{-t}; X_t). \]
We expand \( I_h(X_{-t}; Q_t, X_t) \) in two ways using the chain rule of mutual information:
\[
\begin{align*}
I_h(X_{-t}; Q_t, X_t) &= I(X_{-t}; Q_t, X_t) \\
&= I(X_{-t}; Q_t) + I(X_{-t}; X_t | Q_t) \\
&= I(X_{-t}; X_t) + I_h(X_{-t}; Q_t | X_t) \\
&= I(X_{-t}; X_t) + I_h(X_{-t}; Q_t | X_t)
\end{align*}
\]

The HMM conditional independence properties say that
\[ I_h(X_{-t}; X_t | Q_t) = 0, \]
implying
\[ I_h(X_{-t}; Q_t) = I(X_{-t}; X_t) + I_h(X_{-t}; Q_t | X_t) \]  \hspace{1cm} (5.68)

Reasoning as above, this leads to
\[ I_h(X_t; Q_t) \geq I(X_t; X_{-t}), \]  \hspace{1cm} (5.73)

or that
\[ I_h(X_{-t}; Q_t) = I(X_{-t}; X_t) \]  \hspace{1cm} (5.69)

since \( I_h(X_{-t}; Q_t | X_t) \geq 0 \). This is the first condition. Similarly, the quantity \( I_h(X_t; Q_t, X_{-t}) \) may be expanded as follows:
\[
\begin{align*}
I_h(X_t; Q_t, X_{-t}) &= I(X_t; Q_t, X_{-t}) \\
&= I(X_t; Q_t) + I_h(X_t; X_{-t} | Q_t) \\
&= I(X_t; X_{-t}) + I_h(X_t; Q_t | X_{-t})
\end{align*}
\]

Reasoning as above, this leads to
\[ I_h(X_t; Q_t) \geq I(X_t; X_{-t}), \]  \hspace{1cm} (5.73)
Proof cont.: HMMs Generative Accuracy

... cont.

A sequence of inequalities establishes the third condition:

\[
\log |D_Q| \geq H(Q_t) \geq H(Q_t | X_t)
\]

\[
= I_h(Q_t; X_t) \geq I(X_t; X_{\neg t})
\]

so \( |D_Q| \geq 2^{I(X_t; X_{\neg t})} \).

- This is a lower bound - the number of states must have enough capacity so that it is not a bottleneck, at the very least!
- This could be quite large, and grow with \( T \).
- r.h.s. \( I(X_t; X_{\neg t}) \) is upper bounded by \( H(X_{\neg t}) \) which could be as bad as \( \log |D_{X_{\neg t}}| \).

Nec. conditions for HMMs Generative Accuracy

- Insufficient states can lead to model inaccuracies (e.g., state duration distribution using a geometric rather than something more realistic, add states to improve duration distribution while sharing observation parameters)
- Observation density family must be rich enough (2nd inequality)
- Two bottlenecks: observation density (e.g., number of Components of a Gaussian mixture), and time-dependency (number of states).
Sufficient conditions for HMM accuracy. An HMM \( p_h(X_{1:T}) \) will accurately represent a true discrete distribution \( p(X_{1:T}) \) if the following conditions hold for all \( t \):

- \( H(Q_t | X_{<t}) = 0 \)
- \( p_h(X_t = x_t | q_{x_{<t}}) = p(X_t = x_t | X_{<t} = x_{<t}) \).

where \( q_{x_{<t}} = f(x_{<t}) \) is the unique state sub-sequence associated with \( x_{<t} \).

- Quite strong and unrealistic requirements, but they guarantee accuracy nonetheless.
- Note \( \{< t \} \triangleq \{1, 2, \ldots, t - 1\} \)

Proof: Suff conds for HMMs Generative Accuracy

Proof.

We have for all \( t \):

\[
D(p(X_t | X_{<t}) || p_h(X_t | X_{<t}))
\]  
\[
= \sum_{x_{1:t}} p(x_{1:t}) \log \frac{p(x_t | x_{<t})}{p_h(x_t | x_{<t})} \]  
\[
= \sum_{x_{1:t}} p(x_{1:t}) \log \frac{p(x_t | x_{<t})}{\sum_{q_t} p_h(x_t | q_t) p_h(q_t | x_{<t})} \]  
\[
= \sum_{x_{1:t}} p(x_{1:t}) \log \frac{p(x_t | x_{<t}) p_h(x_t | q_{x_{<t}})}{p_h(x_t | q_{x_{<t}})} \]  
\[
= \sum_{x_{1:t}} p(x_{1:t}) \log \frac{p(x_t | x_{<t})}{p(x_t | x_{<t})} \]  
\[
= 0
\]
It then follows, using the above equation, that:

\[
0 = \sum_t D(p(X_t|X_{<t})||p_h(X_t|X_{<t})) \quad (5.80)
\]

\[
= \sum_t \sum_{x_{1:t}} p(x_{1:t}) \log \frac{p(x_t|x_{<t})}{p_h(x_t|x_{<t})} = \sum_t \sum_{x_{1:T}} p(x_{1:T}) \log \frac{p(x_t|x_{<t})}{p_h(x_{1:T})} \quad (5.81)
\]

\[
= \sum_{x_{1:T}} p(x_{1:T}) \log \frac{\prod_t p(x_t|x_{<t})}{\prod_t p_h(x_t|x_{<t})} = \sum_{x_{1:T}} p(x_{1:T}) \log \frac{p(x_{1:T})}{p_h(x_{1:T})} \quad (5.82)
\]

\[
= D(p(X_{1:T})||p_h(X_{1:T})) \quad (5.83)
\]

- Strong conditions \( H(Q_t|X_{<t}) = 0 \), not likely to happen in practice.
- Is this really what we need for an HMM, generative accuracy? We’ll address this again soon.

---

**Mealy vs. Moore Machines**

- **Mealy vs. Moore finite state automata:**

  Moore machine has only one possible output for each state — the output is a function only of the current state.

  Mealy machine has only one possible output for each state-input pair — given current state, the input determines both the next state and the current output.

  Mealy machine: given a current state, and a next state (determined by input), there is an output.

  Hence, we can think of a Mealy machine as one where the output is a function of the transition between two states.
Mealy vs. Moore HMMs

- Mealy (or Jelinek) HMMs are ones where the observation distribution is a function of the transition between states.
- Markov transition diagram often shown with symbols on the edges between state nodes (note, this is not a graphical model, rather a labeled state-transition diagram).

```
1 2
```

- Moore (or Rabiner) HMMs are ones where the observation is a function of the state itself.
- Markov transition diagram often shown with symbols on the nodes of the diagram.

```
all but
"(" or ")"
```

Mealy vs. Moore HMMs

- Two ways to show the pronunciation model for the word “and”
  ```
  æ
  n
  d
  a
  ```
  ```
  q0
  q1
  q2
  q4
  ```

Left: Moore machine, Right: Mealy machine.
- The output label on each Mealy arcs is shown as input:output and \( \epsilon \) means “null”
Mealy vs. Moore HMMs

- Graphical model for Moore-HMM: as we have seen it ($Q_t$ is a parent of $X_t$ for each $t$).

- Graphical model for Mealy-HMM: $X_t$ has two parents, either ($Q_{t-1}, Q_t$) or ($Q_t, Q_{t+1}$).

Mealy and Moore HMMs are equivalent in that they correspond to the same class of probability distributions.

- Mealy → Moore conversion: cluster adjacent variables in the graphical model.
- More → Mealy conversion: Use a separate arc for each possible output with observation randomness applied to the arc.
- It is sometimes more efficient, in terms of number of states, to specify a language in terms of a Mealy HMM (see parenthesis matching example in the readings).
Sources for Today’s Lecture

- “doc.pdf” sections 8.1 - 8.3
<table>
<thead>
<tr>
<th>HMMs</th>
<th>Trellis</th>
<th>Other HMM queries</th>
<th>MPE</th>
<th>Sampling</th>
<th>What HMMs can do</th>
<th>Summary</th>
<th>Scratch</th>
</tr>
</thead>
</table>

**Scratch Paper**