Statistical Machine Learning from Data

Hidden Markov Models

Samy Bengio

IDIAP Research Institute, Martigny, Switzerland, and
Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland
bengio@idiap.ch
http://www.idiap.ch/~bengio

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

December 21, 2005

Markov Models

- Stochastic process of a temporal sequence: the probability distribution of the variable q at time t depends on the variable q at times $t-1$ to 1.

$$P(q_1, q_2, \ldots, q_T) = P(q_1^T) = P(q_1) \prod_{t=2}^{T} P(q_t | q_1^{t-1})$$

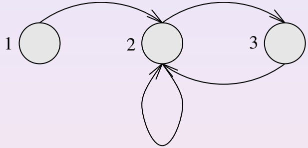
- First Order Markov Process:

$$P(q_t | q_1^{t-1}) = P(q_t | q_{t-1})$$

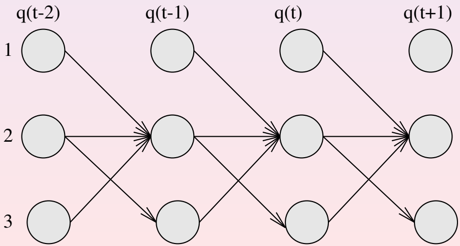
- Markov Model: model of a Markovian process with discrete states.

Markov Models (Graphical View)

- A Markov model:



- A Markov model unfolded in time:



Training Markov Models

- A Markov model is represented by all its transition probabilities:

$$P(q_t = i | q_{t-1} = j) \quad \forall i, j$$

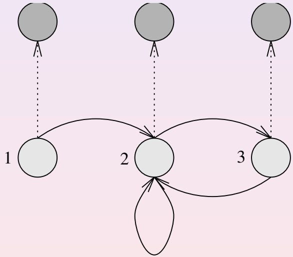
- Given a training set of sequences X, training means re-estimating these probabilities.
- Simply count them to obtain the maximum likelihood solution:

$$P(q_t = i | q_{t-1} = j) = \frac{\#(q_t = i \text{ and } q_{t-1} = j | X)}{\#(q_{t-1} = j | X)}$$

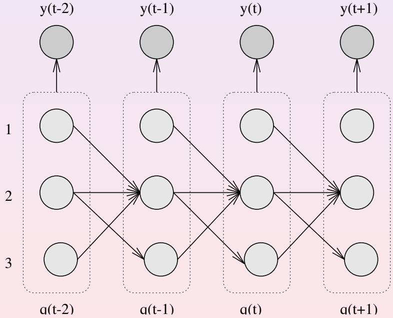
- Example: observe the weather today assuming it depends on the previous day.

Hidden Markov Models

- A hidden Markov model:



- A hidden Markov model unfolded in time:



Elements of an HMM

Hidden Markov Model: Markov Model whose state is not observed, but of which one can observe a manifestation (a variable x_t which depends only on q_t).

- A finite number of states N.
- Transition probabilities between states, which depend only on previous state: $P(q_t = i | q_{t-1} = j, \theta)$.
- Emission probabilities, which depend only on the current state: $p(x_t | q_t = i, \theta)$ (where x_t is observed).
- Initial state probabilities: $P(q_0 = i | \theta)$.
- Each of these 3 sets of probabilities have parameters θ to estimate.

The 3 Problems of HMMs

- The HMM model gives rise to 3 different problems:
  - Given an HMM parameterized by θ, can we compute the likelihood of a sequence $X = x_1^T = \{x_1, x_2, \ldots, x_T\}$:

$$p(x_1^T|\theta)$$

  - Given an HMM parameterized by θ and a set of sequences D_n, can we select the parameters θ^* such that:

$$\theta^* = \arg\max_\theta \prod_{p=1}^{n} p(X(p)|\theta)$$

  - Given an HMM parameterized by θ, can we compute the optimal path Q through the state space given a sequence X:

$$Q^* = \arg\max_Q p(X, Q|\theta)$$

HMMs as Generative Processes

HMMs can be use to generate sequences:

- Let us define a set of starting states with initial probabilities $P(q_0 = i)$.
- Let us also define a set of final states.
- Then for each sequence to generate:
  1. Select an initial state j according to $P(q_0)$.
  2. Select the next state i according to $P(q_t = i | q_{t-1} = j)$.
  3. Emit an output according to the emission distribution $P(x_t | q_t = i)$.
  4. If i is a final state, then stop, otherwise loop to step 2.

Markovian Assumptions

- Emissions: the probability to emit x_t at time t in state $q_t = i$ does not depend on anything else:

$$p(x_t | q_t = i, q_1^{t-1}, x_1^{t-1}) = p(x_t | q_t = i)$$

- Transitions: the probability to go from state j to state i at time t does not depend on anything else:

$$P(q_t = i | q_{t-1} = j, q_1^{t-2}, x_1^{t-1}) = P(q_t = i | q_{t-1} = j)$$

- Moreover, this probability does not depend on time t:

$$P(q_t = i | q_{t-1} = j) \text{ is the same for all } t$$

we say that such Markov models are homogeneous.

Derivation of the Forward Variable α

the probability of having generated the sequence x_1^t and being in state i at time t:

$$
\begin{aligned}
\alpha(i,t) &\stackrel{\text{def}}{=} p(x_1^t, q_t = i) \\
&= p(x_t | x_1^{t-1}, q_t = i) p(x_1^{t-1}, q_t = i) \\
&= p(x_t | q_t = i) \sum_j p(x_1^{t-1}, q_t = i, q_{t-1} = j) \\
&= p(x_t | q_t = i) \sum_j P(q_t = i | x_1^{t-1}, q_{t-1} = j) p(x_1^{t-1}, q_{t-1} = j) \\
&= p(x_t | q_t = i) \sum_j P(q_t = i | q_{t-1} = j) p(x_1^{t-1}, q_{t-1} = j) \\
&= p(x_t | q_t = i) \sum_j P(q_t = i | q_{t-1} = j) \alpha(j, t-1)
\end{aligned}
$$

From α to the Likelihood

- Reminder: $\alpha(i,t) \stackrel{\text{def}}{=} p(x_1^t, q_t = i)$
- Initial condition:

$$\alpha(i,0) = P(q_0 = i) \rightarrow \text{ prior probabilities of each state } i$$

- Then let us compute $\alpha(i,t)$ for each state i and each time t of a given sequence x_1^T
- Afterward, we can compute the likelihood as follows:

$$\begin{aligned} p(x_1^T) &= \sum_i p(x_1^T, q_T = i) \\ &= \sum_i \alpha(i, T) \end{aligned}$$

- Hence, to compute the likelihood $p(x_1^T)$, we need $\mathcal{O}(N^2 \cdot T)$ operations, where N is the number of states

EM Training for HMM

- For HMM, the hidden variable Q will describe in which state the HMM was for each observation x_t of a sequence X.
- The joint likelihood of all sequences $X(l)$ and the hidden variable Q is then:

$$p(X, Q|\theta) = \prod_{l=1}^{n} p(X(l), Q|\theta)$$

- Let us introduce the following indicator variable:

$$q_{i,t} = \begin{cases} 1 & \text{if } q_t = i \\ 0 & \text{otherwise} \end{cases}$$

Joint Likelihood

Let us now use our indicator variables q to instanciate Q:

$$
\begin{aligned}
p(X, Q|\theta) &= \prod_{l=1}^{n} p(X(l), Q|\theta) \\
&= \prod_{l=1}^{n} \left(\prod_{i=1}^{N} P(q_0 = i)^{q_{i,0}} \right) \cdot \\
&\quad \prod_{t=1}^{T_l} \prod_{i=1}^{N} p(x_t(l)|q_t = i)^{q_{i,t}} \prod_{j=1}^{N} P(q_t = i|q_{t-1} = j)^{q_{i,t} \cdot q_{j,t-1}}
\end{aligned}
$$

Joint Log Likelihood

$$
\begin{aligned}
\log p(X, Q|\theta) \quad = \quad & \sum_{l=1}^{n} \sum_{i=1}^{N} q_{i,0} \log P(q_0 = i) + \\
& \sum_{l=1}^{n} \sum_{t=1}^{T_l} \sum_{i=1}^{N} q_{i,t} \log p(x_t(l)|q_t = i) + \\
& \sum_{l=1}^{n} \sum_{t=1}^{T_l} \sum_{i=1}^{N} \sum_{j=1}^{N} q_{i,t} \cdot q_{j,t-1} \log P(q_t = i|q_{t-1} = j)
\end{aligned}
$$

Auxiliary Function

Let us now write the corresponding auxiliary function:

$$
\begin{aligned}
A(\theta, \theta^s) &= E_Q[\log p(X, Q|\theta)|X, \theta^s] \\
&= \sum_{l=1}^{n} \sum_{i=1}^{N} E_Q[q_{i,0}|X, \theta^s] \log P(q_0 = i) + \\
&\quad \sum_{l=1}^{n} \sum_{t=1}^{T_l} \sum_{i=1}^{N} E_Q[q_{i,t}|X, \theta^s] \log p(x_t(l)|q_t = i) + \\
&\quad \sum_{l=1}^{n} \sum_{t=1}^{T_l} \sum_{i=1}^{N} \sum_{j=1}^{N} E_Q[q_{i,t} \cdot q_{j,t-1}|X, \theta^s] \log P(q_t = i|q_{t-1} = j)
\end{aligned}
$$

From now on, let us forget about index l for simplification.

Derivation of the Backward Variable β

the probability to generate the rest of the sequence x_{t+1}^T given that we are in state i at time t

$$
\begin{aligned}
\beta(i,t) &\stackrel{\text{def}}{=} p(x_{t+1}^T | q_t{=}i) \\
&= \sum_j p(x_{t+1}^T, q_{t+1}{=}j | q_t{=}i) \\
&= \sum_j p(x_{t+1} | x_{t+2}^T, q_{t+1}{=}j, q_t{=}i) p(x_{t+2}^T, q_{t+1}{=}j | q_t{=}i) \\
&= \sum_j p(x_{t+1} | q_{t+1}{=}j) p(x_{t+2}^T | q_{t+1}{=}j, q_t{=}i) P(q_{t+1}{=}j | q_t{=}i) \\
&= \sum_j p(x_{t+1} | q_{t+1}{=}j) p(x_{t+2}^T | q_{t+1}{=}j) P(q_{t+1}{=}j | q_t{=}i) \\
&= \sum_j p(x_{t+1} | q_{t+1}{=}j) \beta(j, t+1) P(q_{t+1}{=}j | q_t{=}i)
\end{aligned}
$$

Final Details About β

- Reminder: $\beta(i,t) = p(x_{t+1}^T | q_t{=}i)$
- Final condition:

$$\beta(i,T) = \begin{cases} 1 & \text{if } i \text{ is a final state} \\ 0 & \text{otherwise} \end{cases}$$

- Hence, to compute all the β variables, we need $\mathcal{O}(N^2 \cdot T)$ operations, where N is the number of states

E-Step for HMMs

- Posterior on emission distributions:

$$
\begin{aligned}
E_Q[q_{i,t}|X,\theta^s] &= P(q_t = i|x_1^T, \theta^s) = P(q_t = i|x_1^T) \\
&= \frac{p(x_1^T, q_t = i)}{p(x_1^T)} \\
&= \frac{p(x_{t+1}^T|q_t = i, x_1^t)p(x_1^t, q_t = i)}{p(x_1^T)} \\
&= \frac{p(x_{t+1}^T|q_t = i)p(x_1^t, q_t = i)}{p(x_1^T)} \\
&= \frac{\beta(i,t) \cdot \alpha(i,t)}{\displaystyle\sum_j \alpha(j,T)}
\end{aligned}
$$

E-Step for HMMs

- Posterior on transition distributions:

$$E_Q[q_{i,t} \cdot q_{j,t-1} | X, \theta^s] = P(q_t = i, q_{t-1} = j | x_1^T, \theta^s)$$

$$
\begin{aligned}
&= \frac{p(x_1^T, q_t = i, q_{t-1} = j)}{p(x_1^T)} \\
&= \frac{p(x_{t+1}^T | q_t{=}i) P(q_t{=}i | q_{t-1}{=}j) p(x_t | q_t{=}i) p(x_1^{t-1}, q_{t-1}{=}j)}{p(x_1^T)} \\
&= \frac{\beta(i,t) P(q_t = i | q_{t-1} = j) p(x_t | q_t = i) \alpha(j, t-1)}{\sum_j \alpha(j, T)}
\end{aligned}
$$

E-Step for HMMs

- Posterior on initial state distribution:

$$\begin{aligned} E_Q[q_{i,0}|X,\theta^p] &= P(q_0 = i|x_1^T, \theta^s) = P(q_0 = i|x_1^T) \\ &= \frac{p(x_1^T, q_0 = i)}{p(x_1^T)} \\ &= \frac{p(x_1^T|q_0 = i)P(q_0 = i)}{p(x_1^T)} \\ &= \frac{\beta(i,0) \cdot P(q_0 = i)}{\sum_j \alpha(j, T)} \end{aligned}$$

M-Step for HMMs

- Find the parameters θ that maximizes A, hence search for

$$\frac{\partial A}{\partial \theta} = 0$$

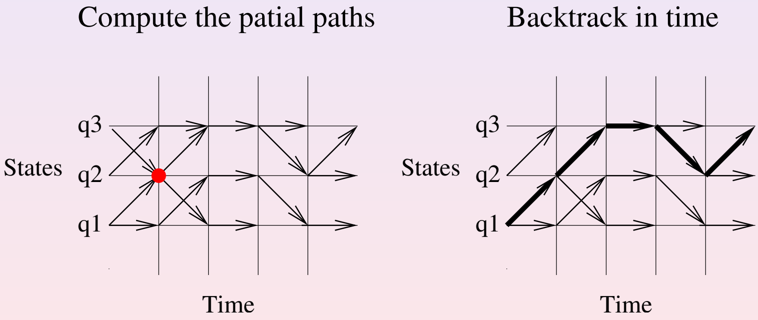
- When transition distributions are represented as tables, using a Lagrange multiplier, we obtain:

$$P(q_t = i | q_{t-1} = j) = \frac{\displaystyle\sum_{t=1}^{T} P(q_t = i, q_{t-1} = j | x_1^T, \theta^s)}{\displaystyle\sum_{t=1}^{T} P(q_{t-1} = j | x_1^T, \theta^s)}$$

- When emission distributions are implemented as GMMs, use already given equations, weighted by the posterior on emissions $P(q_t = i | x_1^T, \theta^s)$.

The Most Likely Path (Graphical View)

- The Viterbi algorithm finds the best state sequence.



The Viterbi Algorithm for HMMs

The Viterbi algorithm finds the best state sequence.

$$
\begin{aligned}
V(i,t) &\stackrel{\text{def}}{=} \max_{q_1^{t-1}} p(x_1^t, q_1^{t-1}, q_t{=}i) \\
&= \max_{q_1^{t-1}} p(x_t|x_1^{t-1}, q_1^{t-1}, q_t{=}i)\, p(x_1^{t-1}, q_1^{t-1}, q_t{=}i) \\
&= p(x_t|q_t{=}i) \max_{q_1^{t-2}} \max_j p(x_1^{t-1}, q_1^{t-2}, q_t{=}i, q_{t-1}{=}j) \\
&= p(x_t|q_t{=}i) \max_{q_1^{t-2}} \max_j p(q_t{=}i|q_{t-1}{=}j)\, p(x_1^{t-1}, q_1^{t-2}, q_{t-1}{=}j) \\
&= p(x_t|q_t{=}i) \max_j p(q_t{=}i|q_{t-1}{=}j) \max_{q_1^{t-2}} p(x_1^{t-1}, q_1^{t-2}, q_{t-1}{=}j) \\
&= p(x_t|q_t{=}i) \max_j p(q_t{=}i|q_{t-1}{=}j)\, V(j, t-1)
\end{aligned}
$$

From Viterbi to the State Sequence

- Reminder: $V(i,t) = \max_{q_1^{t-1}} p(x_1^t, q_1^{t-1}, q_t{=}i)$
- Let us compute $V(i,t)$ for each state i and each time t of a given sequence x_1^T
- Moreover, let us also keep for each $V(i,t)$ the associated argmax previous state j
- Then, starting from the state $i = \arg\max_j V(j,T)$ backtrack to decode the most probable state sequence.
- Hence, to compute all the $V(i,t)$ variables, we need $\mathcal{O}(N^2 \cdot T)$ operations, where N is the number of states

Applications of HMMs

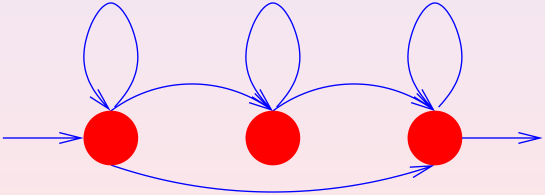
- Classifying sequences such as...
  - DNA sequences (which family)
  - gesture sequences
  - video sequences
  - phoneme sequences
  - etc.
- Decoding sequences such as...
  - continuous speech recognition
  - handwriting recognition
  - sequence of events (meeting, surveilance, games, etc)

HMMs for Speech Recognition

- Application: continuous speech recognition:

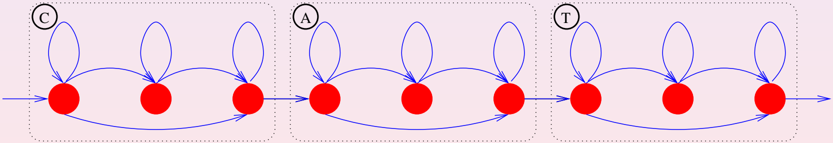
  Find a sequence of phonemes (or words) given an acoustic sequence

- Idea: use a phoneme model



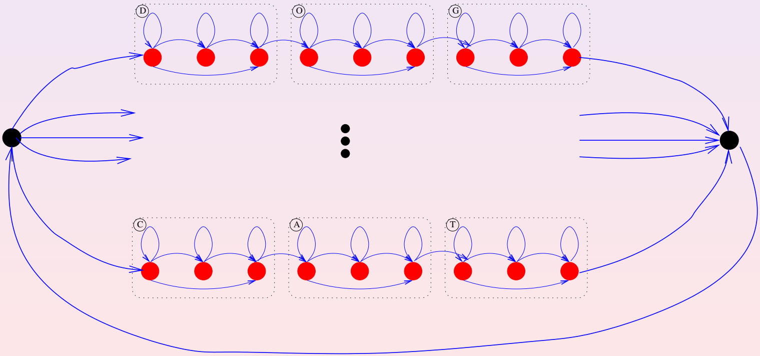
Embbeded Training of HMMs

- For each acoustic sequence in the training set, create a new HMM as the concatenation of the HMMs representing the underlying sequence of phonemes.
- Maximize the likelihood of the training sentences.



HMMs: Decoding a Sentence

- Decide what is the accepted vocabulary.
- Optionally add a language model: $P(\text{word sequence})$
- Efficient algorithm to find the optimal path in the decoding HMM:



Measuring Error

- How do we measure the quality of a speech recognizer?
- Problem: the target solution is a sentence, the obtained solution is also a sentence, but they might have different size!
- Proposed solution: the Edit Distance:
  - assume you have access to the operators insert, delete, and substitute,
  - what is the smallest number of such operators we need to go from the obtained to the desired sentence?
  - An efficient algorithm exists to compute this.
- At the end, we measure the error as follows:

$$\text{WER} = \frac{\#\text{ins} + \#\text{del} + \#\text{subst}}{\#\text{words}}$$

- Note that the word error rate (WER) can be greater than 1...

Maximum Mutual Information

- Using the Maximum Likelihood criterion for a classification task might sometimes be worse than using a discriminative approach
- What about changing the criterion to be more discriminative?
- Maximum Mutual Information (MMI) between word (W) and accoustic (A) sequences:

$$\begin{aligned} I(A,W) &= \log \frac{P(A,W)}{P(A)P(W)} \\ &= \log P(A|W)P(W) - \log P(A) - \log P(W) \\ &= \log P(A|W) - \log P(A) \\ &= \log P(A|W) - \sum_w \log P(A|w)P(w) \end{aligned}$$

- Apply gradient ascent: $\frac{\partial I(A,W)}{\partial \theta}$.

Practical Aspects

- Capacity tuned by the following hyper-parameters:
  - Number of states (or values the hidden variable can take)
  - Non-zero transitions (full-connect, left-to-right, etc)
  - Capacity of underlying emission models
  - Number of training iterations
- Initialization:
  - If the training set is aligned, use this information
  - Otherwise, uniform for transitions, K-Means for GMM-based emissions
- Computational contraint:
  - Work in the logarithmic domain!

Imbalance between Transitions and Emissions

- A problem often seen in speech recognition...
- Decoding with Viterbi:

$$V(i,t) = p(x_t|q_t = i) \max_j P(q_t = i|q_{t-1} = j) V(j, t-1)$$

- Emissions represented by GMMs: densities depend on the number of dimensions of x_t.
- Practical estimates on Numbers'95 database (39 dimensions):

	Variance
$\log P(q_t\|q_{t-1})$	9.8
$\log p(x_t\|q_t)$	11486.0

Comparison of variances of log distributions during decoding