16.410/413
Principles of Autonomy and Decision Making

Lecture 21: Intro to Hidden Markov Models
the Baum-Welch algorithm

Emilio Frazzoli

Aeronautics and Astronautics
Massachusetts Institute of Technology

November 24, 2010

Assignments

Readings

- Lecture notes
- [AIMA] Ch. 15.1-3, 20.3.
- Paper on Stellar: L. Rabiner, "A tutorial on Hidden Markov Models..."

Outline

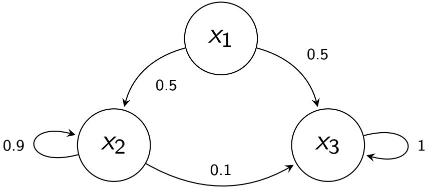
1. Decoding and Viterbi's algorithm
2. Learning and the Baum-Welch algorithm

Decoding

- Filtering and smoothing produce distributions of states at each time step.
- Maximum likelihood estimation chooses the state with the highest probability at the “best” estimate at each time step.
- However, these are pointwise best estimate: the sequence of maximum likelihood estimates is not necessarily a good (or feasible) trajectory for the HMM!
- How do we find the most likely state history, or state trajectory? (As opposed to the sequence of point-wise most likely states?)

- Three states:
  $\mathcal{X} = \{x_1, x_2, x_3\}$.
- Three possible observations:
  $\mathcal{Z} = \{2, 3\}$.
- Initial distribution:
  $\pi = (1, 0, 0)$.
- Transition probabilities:

$$T = \begin{bmatrix} 0 & 0.5 & 0.5 \\ 0 & 0.9 & 0.1 \\ 0 & 0 & 1 \end{bmatrix}$$



- Observation probabilities:

$$M = \begin{bmatrix} 0.5 & 0.5 \\ 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}$$

Observation sequence:

$$Z = (2, 3, 3, 2, 2, 2, 3, 2, 3).$$

Example: filtering/smoothing vs. decoding 2/4

- Using filtering:

t	x_1	x_2	x_3
1	**1.0000**	0	0
2	0	0.1000	**0.9000**
3	0	0.0109	**0.9891**
4	0	0.0817	**0.9183**
5	0	0.4165	**0.5835**
6	0	**0.8437**	0.1563
7	0	0.2595	**0.7405**
8	0	**0.7328**	0.2672
9	0	0.1771	**0.8229**

- The sequence of *point-wise* most likely states is:

$$(1, 3, 3, 3, 3, 2, 3, 2, 3).$$

- The above sequence is not feasible for the HMM model!

Example: filtering vs. smoothing vs. decoding 3/4

- Using smoothing:

t	x_1	x_2	x_3
1	**1.0000**	0	0
2	0	**0.6297**	0.3703
3	0	**0.6255**	0.3745
4	0	**0.6251**	0.3749
5	0	**0.6218**	0.3782
6	0	**0.5948**	0.4052
7	0	0.3761	**0.6239**
8	0	0.3543	**0.6457**
9	0	0.1771	**0.8229**

- The sequence of *point-wise* most likely states is:

$$(1, 2, 2, 2, 2, 2, 3, 3, 3).$$

Viterbi's algorithm

- As before, let us use the Markov property of the HMM.
- Define

$$\delta_k(s) = \max_{X_{1:(k-1)}} \Pr\left[X_{1:k} = (X_{1:(k-1)}, s), Z_{1:k} | \lambda\right]$$

(i.e., $\delta_k(s)$ is the joint probability of the most likely path that ends at state s at time k, generating observations $Z_{1:k}$.)

- Clearly,

$$\delta_{k+1}(s) = \max_q \left(\delta_k(q)\, T_{q,s}\right) M_{s,z_{k+1}}$$

- This can be iterated to find the probability of the most likely path that ends at each possible state s at the final time. Among these, the highest probability path is the desired solution.
- We need to keep track of the path...

Viterbi's algorithm 2/3

- Initialization, for all $s \in \mathcal{X}$:
  - $\delta_1(s) = \pi_s M_{s,z_1}$
  - $\mathrm{Pre}_1(s) = \texttt{null}$.
- Repeat, for $k = 1, \ldots, t-1$, and for all $s \in \mathcal{X}$:
  - $\delta_{k+1}(s) = \max_q \left(\delta_k(q) T_{q,s}\right) M_{s,z_{k+1}}$
  - $\mathrm{Pre}_{k+1}(s) = \arg\max_q \left(\delta_k(q) T_{q,s}\right)$
- Select most likely terminal state: $s_t^* = \arg\max_s \delta_t(s)$
- Backtrack to find most likely path. For $k = t-1, \ldots, 1$
  - $q_k^* = \mathrm{Pre}_{k+1}(q_{k+1}^*)$
- The joint probability of the most likely path + observations is found as $p^* = \delta_t(s_t^*)$.

Whack-the-mole example

- Viterbi's algorithm
  - $\delta_1 = (0.6, 0, 0)$ $\quad \text{Pre}_1 = (\texttt{null}, \texttt{null}, \texttt{null})$
  - $\delta_2 = (0.012, 0.048, 0.18)$ $\quad \text{Pre}_2 = (1, 1, 1).$
  - $\delta_3 = (0.0038, 0.0216, 0.0432)$ $\quad \text{Pre}_3 = (2, 3, 3).$
- Joint probability of the most likely path + observations: 0.0432
- End state of the most likely path: 3
- Most likely path: $3 \leftarrow 3 \leftarrow 1$.

- Using Viterbi's algorithm:

t	x_1	x_2	x_3
1	0.5/0	0	0
2	0/1	0.025/1	0.225/1
3	0/1	0.00225/2	0.2025/3
4	0/1	0.0018225/2	0.02025/3
5	0/1	0.0014762/2	0.002025/3
6	0/1	0.0011957/2	0.0002025/3
7	0/1	0.00010762/2	0.00018225/3
8	0/1	8.717e-05/2	1.8225e-05/3
9	0/1	7.8453e-06/2	1.6403e-05/3

- The most likely sequence is:

$$(1, 3, 3, 3, 3, 3, 3, 3, 3).$$

- *Note: Based on the first 8 observations, the most likely sequence would have been*

$$(1, 2, 2, 2, 2, 2, 2, 2)!$$

Viterbi's algorithm 3/3

- Viterbi's algorithm is similar to the forward algorithm, with the difference that the summation over the states at time step k becomes a maximization.
- The time complexity is, as for the forward algorithm, linear in t (and quadratic in $\mathrm{card}(\mathcal{X})$).
- The space complexity is also linear in t (unlike the forward algorithm), since we need to keep track of the "pointers" Pre_k.
- Viterbi's algorithm is used in most communication devices (e.g., cell phones, wireless network cards, etc.) to decode messages in noisy channels; it also has widespread applications in speech recognition.

Outline

Problem 3: Learning

The learning problem

Given a HMM λ, and an observation history $Z = (z_1, z_2, \ldots, z_t)$, find a new HMM λ' that explains the observations at least as well, or possibly better, i.e., such that $\Pr[Z|\lambda'] \geq \Pr[Z|\lambda]$.

- Ideally, we would like to find the model that maximizes $\Pr[Z|\lambda]$; however, this is in general an intractable problem.
- We will be satisfied with an algorithm that converges to local maxima of such probability.
- Notice that in order for learning to be effective, we need lots of data, i.e., many, long observation histories!

Let us consider the following problem.

- The elusive leader of a dangerous criminal organization (e.g., Keyser Söze, from the movie *"The Usual Suspects"*) is known to travel between two cities (say, Los Angeles and New York City)
- The FBI has no clue about his whereabouts at the initial time (e.g., uniform probability being at any one of the cities).
- The FBI has no clue about the probability that he would stay or move to the other city at each time period:

from\to	LA	NY
LA	0.5	0.5
NY	0.5	0.5

- At each time period the FBI could get sighting reports (or evidence of his presence in a city), including a non-sighting `null` report. An estimate of the probability of getting such reports is

where \ report	LA	NY	`null`
LA	0.4	0.1	0.5
NY	0.1	0.5	0.4

- Let us assume that the FBI has been tracking sighting reports for, say, 20 periods, with observation sequence Z

$$Z = (-, LA, LA, -, NY, -, NY, NY, NY, -, NY, NY, NY, NY, NY, -, -, LA, LA, NY).$$

- We can compute, using the algorithms already discussed:
  - the current probability distribution (after the 20 observations):

$$\gamma_{20} = (0.1667, 0.8333)$$

  - the probability distribution at the next period (so that we can catch him):

$$\gamma_{21} = T' \gamma_{20} = (0.5, 0.5)$$

  - the probability of getting that particular observation sequence given the model:

$$\Pr[Z|\lambda] = 1.9 \cdot 10^{-10}$$

- Using smoothing:

t	LA	NY
1	0.5556	0.4444
2	0.8000	0.2000
3	0.8000	0.2000
...	...	...
18	0.8000	0.2000
19	0.8000	0.2000
20	0.1667	0.8333

- The sequence of *point-wise* most likely states is:

$$(LA, LA, LA, LA, NY, LA, NY, NY, NY, LA, NY, NY, NY, NY, NY, LA, LA, LA)$$

- The new question is: given all the data, can we improve on our model, in such a way that the observations are more consistent with it?

Expectation of (state) counts

- Let us define
$$\gamma_k(s) = \Pr[X_k = s | Z, \lambda],$$
i.e., $\gamma_k(s)$ is the probability that the system is at state s at the k-th time step, given the observation sequence Z and the model λ.
- We already know how to compute this, e.g., using smoothing:
$$\gamma_k(s) = \frac{\alpha_k(s)\beta_k(s)}{\Pr[Z|\lambda]} = \frac{\alpha_k(s)\beta_k(s)}{\sum_{s \in \mathcal{X}} \alpha_t(s)}.$$
- New concept: how many times is the state trajectory expected to *transition from* state s?
$$\mathrm{E}[\# \text{ of transitions from } s] = \sum_{k=1}^{t-1} \gamma_k(s)$$

Expectation of (transition) counts

- In much the same vein, let us define

$$\xi_k(q,s) = \Pr\left[X_k = q, X_{k+1} = s | Z, \lambda\right]$$

(i.e., $\xi_k(q,s)$ is the probability of being at state q at time k, and at state s at time $k+1$, given the observations and the current HMM model)

- We have that

$$\xi_k(q,s) = \eta_k \alpha_k(q) T_{q,s} M_{s,z_{k+1}} \beta_{k+1}(s),$$

where η_k is a normalization factor, such that $\sum_{q,s} \xi_k(q,s) = 1$.

- New concept: how many times it the state trajectory expected to *transition from* state q *to* state s?

$$\mathrm{E}[\# \text{ of transitions from } q \text{ to } s] = \sum_{k=1}^{t-1} \xi_k(q,s)$$

Baum-Welch algorithm 1

- Based on the probability estimates and expectations computed so far, using the original HMM model $\lambda = (T, M, \pi)$, we can construct a new model $\lambda' = (T', M', \pi')$ (notice that the two models share the states and observations):
- The new initial condition distribution is the one obtained by smoothing:

$$\pi'_s = \gamma_1(s)$$

- The entries of the new transition matrix can be obtained as follows:

$$T'_{qs} = \frac{\mathrm{E}[\#\text{ of transitions from state } q \text{ to state } s]}{\mathrm{E}[\#\text{ of transitions from state } q]} = \frac{\sum_{k=1}^{t-1} \xi_k(q,s)}{\sum_{k=1}^{t-1} \gamma_k(q)}$$

- The entries of the new observation matrix can be obtained as follows:

$$M'_{sm} = \frac{\mathrm{E}[\#\text{ of times in state } s\text{, when the observation was } m]}{\mathrm{E}[\#\text{ of times in state } s]} = \frac{\sum_{k=1}^{t} \gamma_k(s) \cdot \mathbf{1}(z_k = m)}{\sum_{k=1}^{t} \gamma_k(s)}$$

- It can be shown [Baum *et al.*, 1970] that the new model λ' is such that
  - $\Pr[Z|\lambda'] \geq \Pr[Z|\lambda]$, as desired.
  - $\Pr[Z|\lambda'] = \Pr[Z|\lambda]$ only if λ is a critical point of the likelihood function $f(\lambda) = \Pr[Z|\lambda]$

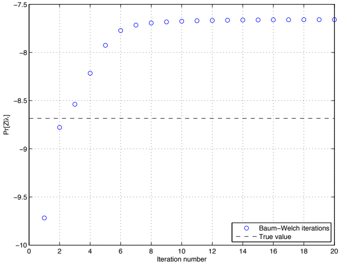
Let us apply the method to the example. We get

- Initial condition: $\pi = (1, 0)$.
- Transition matrix:

$$\begin{bmatrix} 0.6909 & 0.3091 \\ 0.0934 & 0.9066 \end{bmatrix}$$

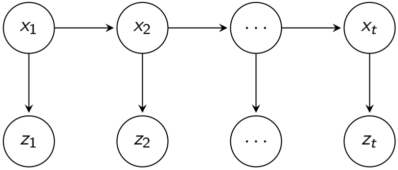
- Observation matrix:

$$\begin{bmatrix} 0.5807 & 0.0010 & 0.4183 \\ 0.0000 & 0.7621 & 0.2379 \end{bmatrix}$$



- Note that it is possible that $\Pr[Z|\lambda'] > \Pr[Z|\lambda_{\text{true}}]$! This is due to overfitting over one particular data set.

Recursive Bayesian estimation: HMMs and Kalman filters



- The idea of the filtering/smoothing techniques for HMM is in fact broader. In general it applies to any system where the state at a time step only depends on the state at the previous time step (Markov property), and the observation at a time step only depends on the state at that time step.
  - HMMs: discrete state (Markov chain), arbitrary transition and observation matrices.
  - Kalman filter: continuous state (Markov process), (Linear-)Gaussian transitions, Gaussian observations.

16.410 / 16.413 Principles of Autonomy and Decision Making
Fall 2010