Hidden Markov Models

Lyle Ungar

Slides mostly from Mitch Marcus and Eric Fosler

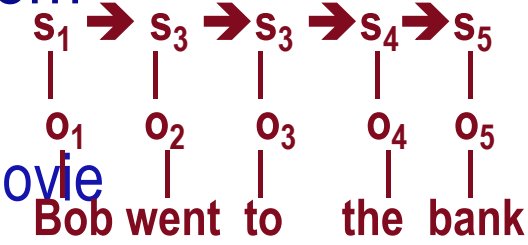
(with lots of modifications)

Markov matrix
Hidden Markov Model
Later: Recurrent Neural Nets

HMMs – models in the style of Kalman filters, linear dynamical systems, and LSTMs

HMMs are dynamic latent variable models

- Given a sequence of *sounds*, find the sequence of *words* most likely to have produced them
- Given a sequence of *images* find the sequence of *locations* most likely to have produced them.
- Given a sequence of *words*, find the sequence of "*meanings*" most likely to have generated them
  - Or *parts of speech*: Noun, verb, adverb, …
  - Or *entity type*: Person, place, company, date, movie
    - E.g. *river bank* vs. *money bank*



Conditional Independence

- If we want the joint probability of an entire sequence, the *Markov* assumption lets us treat it as a product of "bigram" conditional probabilities:

$$p(w1,w2,w3,w4) =$$

$$p(w1)\ p(w2|w1)\ p(w3|w2,w1)\ p(w4|w3,w2,w1) \sim$$

$$p(w1)\ p(w2|w1)\ p(w3|w2) \qquad p(w4|w3)$$

A Markovian weather model:

- **Tomorrow is like today, except when it isn't.**

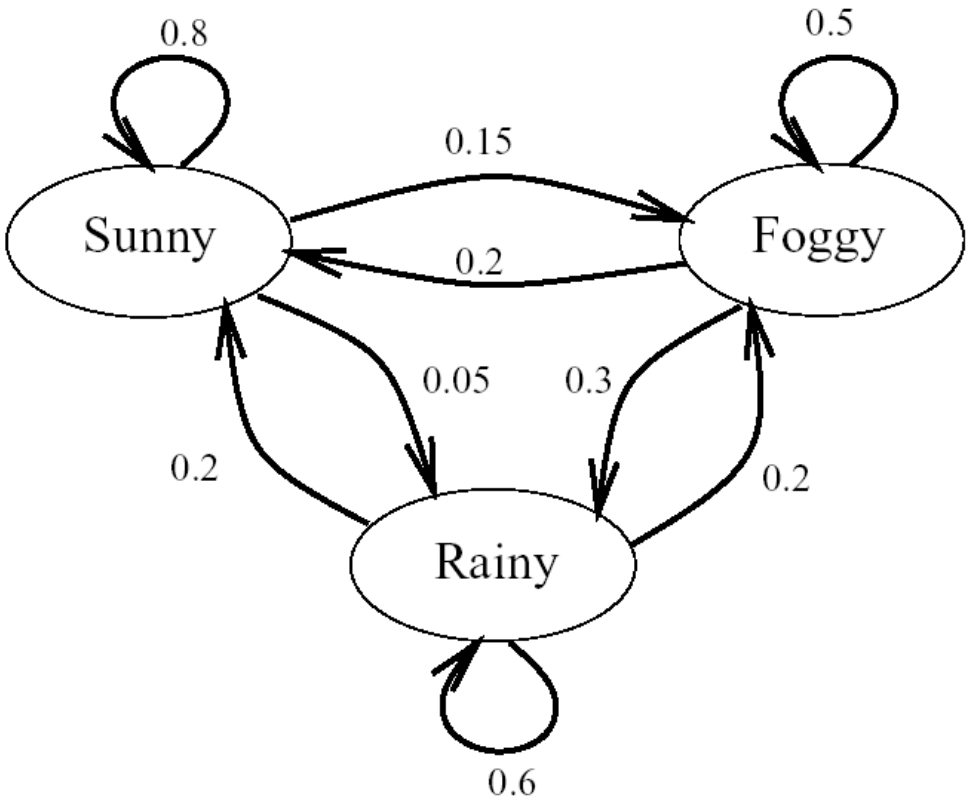
		Tomorrow's Weather		
		Sunny	Rainy	Foggy
Today's Weather	Sunny	0.8	0.05	0.15
	Rainy	0.2	0.6	0.2
	Foggy	0.2	0.3	0.5

If you start from any prior distribution over states (weathers) and run long enough you converge to a stationary distribution.

- **Markov matrix gives**

  p(tomorrow's weather | today's weather)

The same model shown as a graph



Imperfect Knowledge

But sometimes we can only observe a process "as through a glass darkly." We don't get to see what is really happening, but only some clues that are more or less strongly indicative of what might be happening.

So you're bucking for partner in a windowless law office and you don't see the weather for days at a time … But you do see whether your office mate (who has an actual life) brings in an umbrella or not:

	Probability of Umbrella
Sunny	0.1
Rainy	0.8
Foggy	0.3

How to make predictions?

Now you're bored with researching briefs,
and you want to guess the weather
from a sequence of umbrella (non)sightings:

$P(w_1, w_2, \ldots w_n \mid u_1, \ldots, u_n)$

You observe u, but not w.

w is the "hidden" part of the "Hidden Markov Model"

How to do it?

In speech recognition, we observe the sounds,
but not the intended words

Bayes rule rules!

Bayes' Rule!

$$P(w_1, \ldots, w_n \mid u_1, \ldots, u_n) = \frac{P(u_1, \ldots, u_n \mid w_1, \ldots, w_n) P(w_1, \ldots, w_n)}{P(u_1, \ldots, u_n)}$$

A Rainy-Day Example

- **You go into the office Sunday morning and it's sunny.**
  - w_1 = Sunny
- **You work through the night on Sunday, and on Monday morning, your officemate comes in with an umbrella.**
  - u_2 = T
- **What's the probability that Monday is rainy?**
  - $P(w_2=\text{Rainy} \mid w_1=\text{Sunny}, u_2=T) =$
    $P(u_2=T|w_2=\text{Rainy})/P(u_2=T| w_1=\text{Sunny}) \times P(w_2=\text{Rainy}| w_1=\text{Sunny})$
    (likelihood of umbrella)/normalization x prior

Bayes rule for speech

- **To find the most likely word**
  - Start with a prior of how likely each word is
  - And the likelihood of each set of sounds given the word
- **The most likely word is the one most likely to have generated the sounds heard**

The "fundamental equation of speech recognition":

$$argmax_w\ P(w|u) = argmax_w\ P(u|w)\ P(w) / P(u)$$

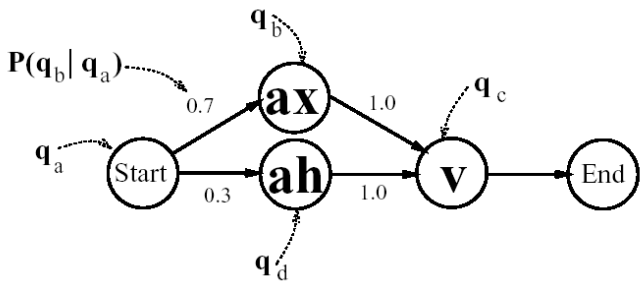
w = word, u = sound ("utterance")

Speech Recognition

- **Markov model for words in a sentence**

  P(I like cabbages) = P(I|START)P(like|I)P(cabbages|like)

- **Markov model for sounds in a word**
  - Model the relation of words to sounds by breaking words down into pieces

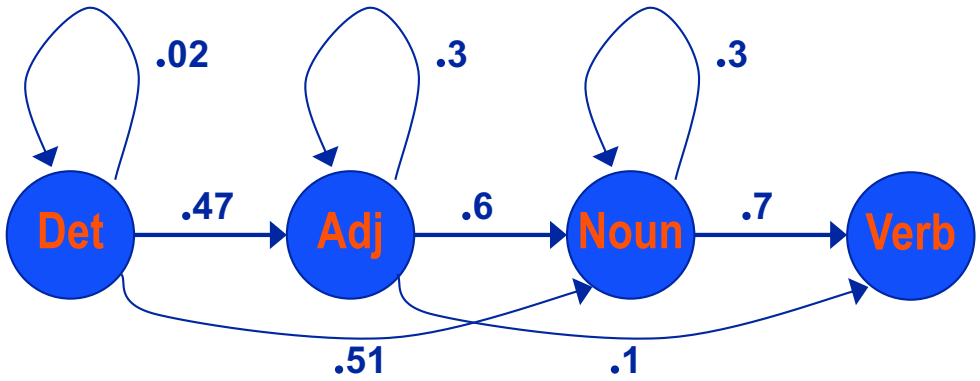


HMMs: Midpoint Summary

- **Language can be modeled by HMMs**
  - Predict words from sounds
  - Captures priors on words
  - Hierarchical
    - Phonemes to morphemes to words to phrases to sentences
  - Was used in all commercial speech recognition software
    - Now replaced by deep networks
- **Markov assumption, HMM definition**
  - “Fundamental equation of speech recognition”

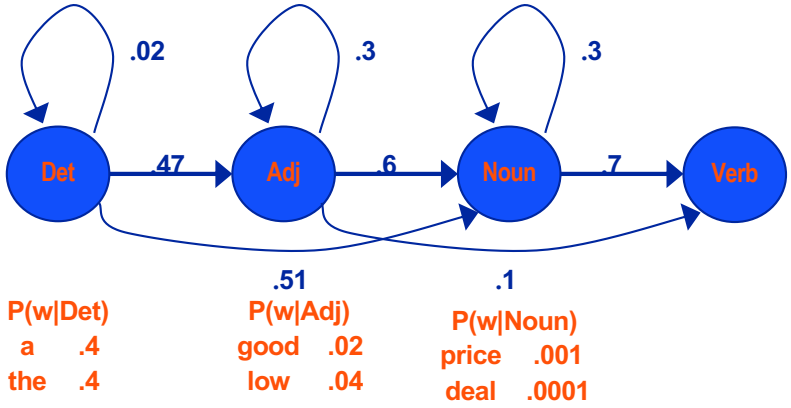
Hidden Markov Models (HMMs)

Part of Speech tagging was often done using HMMs



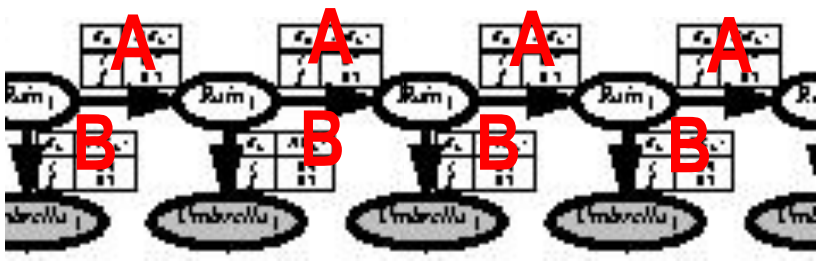
HMM: The Model

- Starts in some initial state s_i
- Moves to a new state s_j with probability $p(s_j \mid s_i) = a_{ij}$
  - The matrix A with elements a_{ij} is the *Markov transition matrix*
- Emits an observation o_v (e.g. a word, w) with probability $p(o_v \mid s_i) = b_{iv}$



HMMs are dynamic Bayesian Networks

There is a node for the hidden state and for the emission (observed state) at each time step, but the probability tables are the same at all times.



A: Markov transition matrix
B: Emission probabilities

Parameters of an HMM

- ***States***: $S = s_1, \ldots, s_k$
- ***Markov transition probabilities***: *Markov transition matrix* $A_{k,k}$ Each $a_{i,j} = p(s_j \mid s_i)$ represents the probability of transitioning from state s_i to s_j.
- ***Emission probabilities***: functions $b_i(o_t) = p(o|s_i)$ giving the probability of observation o_t being emitted by s_i
- ***Initial state distribution***: the probability that s_i is a start state π_i

The Three Basic HMM Problems

- ***Problem 1 (Evaluation):*** Given the observation sequence $O=o_1,\ldots,o_T$ and an HMM model $\lambda = (A,B,\pi)$, compute the probability of O given the model.
- ***Problem 2 (Decoding):*** Given the observation sequence $O=o_1,\ldots,o_T$ and an HMM model $\lambda = (A,B,\pi)$ find the state sequence that best explains the observations

Problem 3 (Learning): Pick the model parameters $\lambda = (A,B,\pi)$ to maximize $P(O \mid \lambda)$

What you should know

- **HMMs**
  - Markov assumption
  - Markov transition matrix, Emission probabilities
  - Solved by EM
- **Many other models generalize HMM**
  - Emission can be a real-valued (e.g. Gaussian) function of the hidden state
  - The hidden state can be a real valued vector instead of a “one hot” discrete state
    - Instead of moving with a Markov Transition Matrix between states, one moves with Gaussian noise between real states
  - Nonlinear: dynamical neural nets