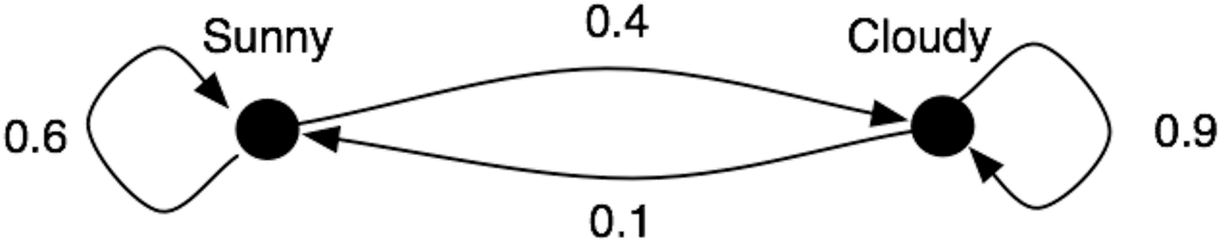
Hidden Markov Models

Sequence models

- A recurring theme in AI is modeling variables that change over time (or another single dimension)
  - E.g. weather across time, words across a sentence, etc.

Markov chains

- Stochastic process model
  - Due to Andrey Markov (1906)
  - e.g.,

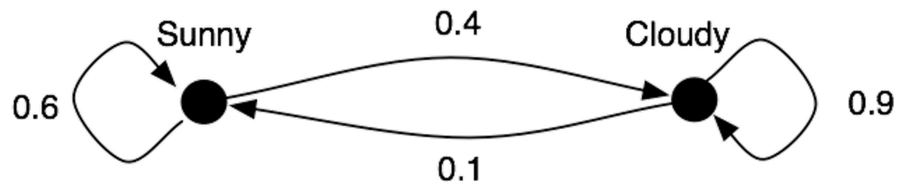


Markov chain

- Models a system which is in exactly one state at any time t, denoted by random variable Q_t
- A Markov chain model consists of:
  - A discrete set of states $S=\{s_1, \dots s_N\}$
  - *An initial probability distribution* $P(Q_0)$
  - Transition probability distribution, given by a conditional distribution $P(Q_{t+1} | Q_t)$
- The Markov assumption:
  - The probability of transitioning to each new state depends *only* on the current state (and not on the previous states)
  - More formally,

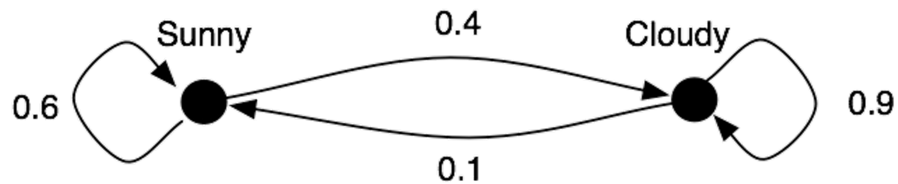
$$P(Q_{t+1} = q_{t+1} | Q_t = q_t, Q_{t-1} = q_{t-1}, \dots, Q_0 = q_0) = P(Q_{t+1} = q_{t+1} | Q_t = q_t)$$

Markov chains



- Suppose there's a 90% chance of sun on day 0. What is the probability of sun on day 3?

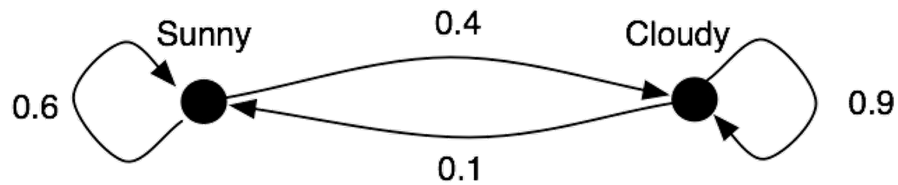
Markov chains



- Suppose there's a 90% chance of sun on day 0. What is the probability of sun on day 3?

$$P(Q_3 = \text{☼}) \quad = \quad P(Q_3 = \text{☼} | Q_2 = \text{☼}) P(Q_2 = \text{☼}) + P(Q_3 = \text{☼} | Q_2 = \text{🌧}) P(Q_2 = \text{🌧})$$

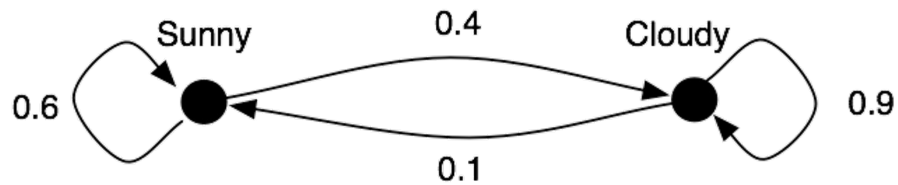
Markov chains



- Suppose there's a 90% chance of sun on day 0. What is the probability of sun on day 3?

$$
\begin{aligned}
P(Q_3 = \text{☼}) &= P(Q_3 = \text{☼} | Q_2 = \text{☼}) P(Q_2 = \text{☼}) + P(Q_3 = \text{☼} | Q_2 = \text{☂}) P(Q_2 = \text{☂}) \\
&= 0.6 P(Q_2 = \text{☼}) + 0.1 P(Q_2 = \text{☂})
\end{aligned}
$$

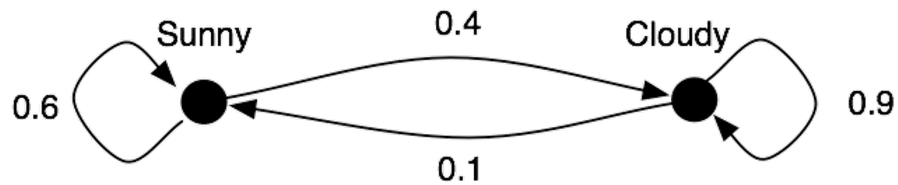
Markov chains



- Suppose there's a 90% chance of sun on day 0. What is the probability of sun on day 3?

$$
\begin{aligned}
P(Q_3 = \text{☀}) &= P(Q_3 = \text{☀} | Q_2 = \text{☀}) P(Q_2 = \text{☀}) + P(Q_3 = \text{☀} | Q_2 = \text{🌧}) P(Q_2 = \text{🌧}) \\
&= 0.6 P(Q_2 = \text{☀}) + 0.1 P(Q_2 = \text{🌧}) \\
&= 0.6(0.6 P(Q_1 = \text{☀}) + 0.1 P(Q_1 = \text{🌧})) + 0.1(0.4 P(Q_1 = \text{☀}) + 0.9 P(Q_1 = \text{🌧}))
\end{aligned}
$$

Markov chains

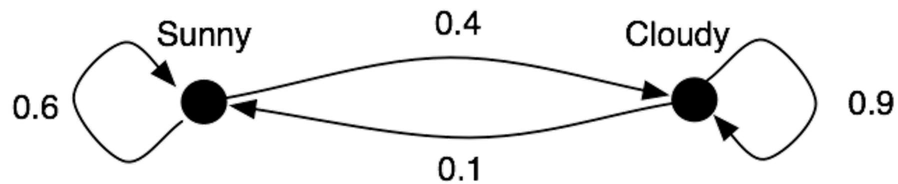


- Suppose there's a 90% chance of sun on day 0.

  What is the probability of sun on day 3?

$$
\begin{aligned}
P(Q_3 = \text{☼}) &= P(Q_3 = \text{☼} | Q_2 = \text{☼}) P(Q_2 = \text{☼}) + P(Q_3 = \text{☼} | Q_2 = \text{🌧}) P(Q_2 = \text{🌧}) \\
&= 0.6 P(Q_2 = \text{☼}) + 0.1 P(Q_2 = \text{🌧}) \\
&= 0.6(0.6 P(Q_1 = \text{☼}) + 0.1 P(Q_1 = \text{🌧})) + 0.1(0.4 P(Q_1 = \text{☼}) + 0.9 P(Q_1 = \text{🌧})) \\
&= 0.6(0.6(0.6 P(Q_0 = \text{☼}) + 0.1 P(Q_0 = \text{🌧})) + 0.1(0.4 P(Q_0 = \text{☼}) + 0.9 P(Q_0 = \text{🌧}))) \\
&+ 0.1(0.4(0.4 P(Q_0 = \text{☼}) + 0.1 P(Q_0 = \text{🌧})) + 0.9(0.4 P(Q_0 = \text{☼}) + 0.9 P(Q_0 = \text{🌧})))
\end{aligned}
$$

Markov chains

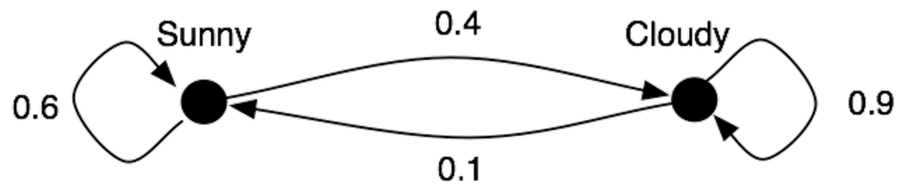


- Suppose there's a 90% chance of sun on day 0.

  What is the probability of sun on day 3?

$$
\begin{aligned}
P(Q_3 = \text{☀}) &= P(Q_3 = \text{☀}|Q_2 = \text{☀})P(Q_2 = \text{☀}) + P(Q_3 = \text{☀}|Q_2 = \text{🌧})P(Q_2 = \text{🌧}) \\
&= 0.6P(Q_2 = \text{☀}) + 0.1P(Q_2 = \text{🌧}) \\
&= 0.6(0.6P(Q_1 = \text{☀}) + 0.1P(Q_1 = \text{🌧})) + 0.1(0.4P(Q_1 = \text{☀}) + 0.9P(Q_1 = \text{🌧})) \\
&= 0.6(0.6(0.6P(Q_0 = \text{☀}) + 0.1P(Q_0 = \text{🌧})) + 0.1(0.4P(Q_0 = \text{☀}) + 0.9P(Q_0 = \text{🌧}))) \\
&+ 0.1(0.4(0.4P(Q_0 = \text{☀}) + 0.1P(Q_0 = \text{🌧})) + 0.9(0.4P(Q_0 = \text{☀}) + 0.9P(Q_0 = \text{🌧}))) \\
&= 0.6(0.6(0.6(0.8) + 0.1(0.2)) + 0.1(0.4(0.8) + 0.9(0.2))) \\
&+ 0.1(0.4(0.6(0.8) + 0.1(0.2)) + 0.9(0.4(0.8) + 0.9(0.2)))
\end{aligned}
$$

Markov chains



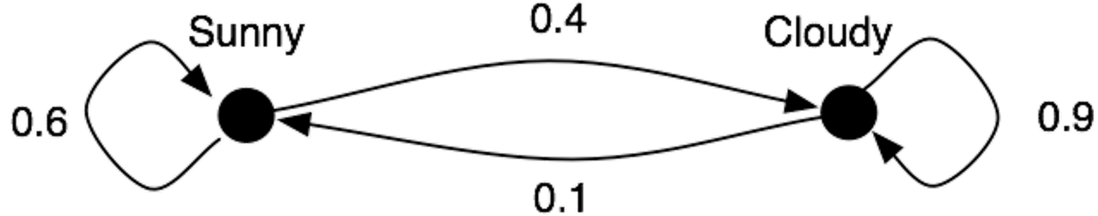
- Suppose there's a 80% chance of sun on day 0.

  What is the probability of sun on day 3?

$$
\begin{aligned}
P(Q_3 = \text{☼}) &= P(Q_3 = \text{☼} | Q_2 = \text{☼}) P(Q_2 = \text{☼}) + P(Q_3 = \text{☼} | Q_2 = \text{☁}) P(Q_2 = \text{☁}) \\
&= 0.6 P(Q_2 = \text{☼}) + 0.1 P(Q_2 = \text{☁}) \\
&= 0.6(0.6 P(Q_1 = \text{☼}) + 0.1 P(Q_1 = \text{☁})) + 0.1(0.4 P(Q_1 = \text{☼}) + 0.9 P(Q_1 = \text{☁})) \\
&= 0.6(0.6(0.6 P(Q_0 = \text{☼}) + 0.1 P(Q_0 = \text{☁})) + 0.1(0.4 P(Q_0 = \text{☼}) + 0.9 P(Q_0 = \text{☁}))) \\
&+ 0.1(0.4(0.4 P(Q_0 = \text{☼}) + 0.1 P(Q_0 = \text{☁})) + 0.9(0.4 P(Q_0 = \text{☼}) + 0.9 P(Q_0 = \text{☁}))) \\
&= 0.6(0.6(0.6(0.8) + 0.1(0.2)) + 0.1(0.4(0.8) + 0.9(0.2))) \\
&+ 0.1(0.4(0.6(0.8) + 0.1(0.2)) + 0.9(0.4(0.8) + 0.9(0.2))) \\
&= 0.275
\end{aligned}
$$

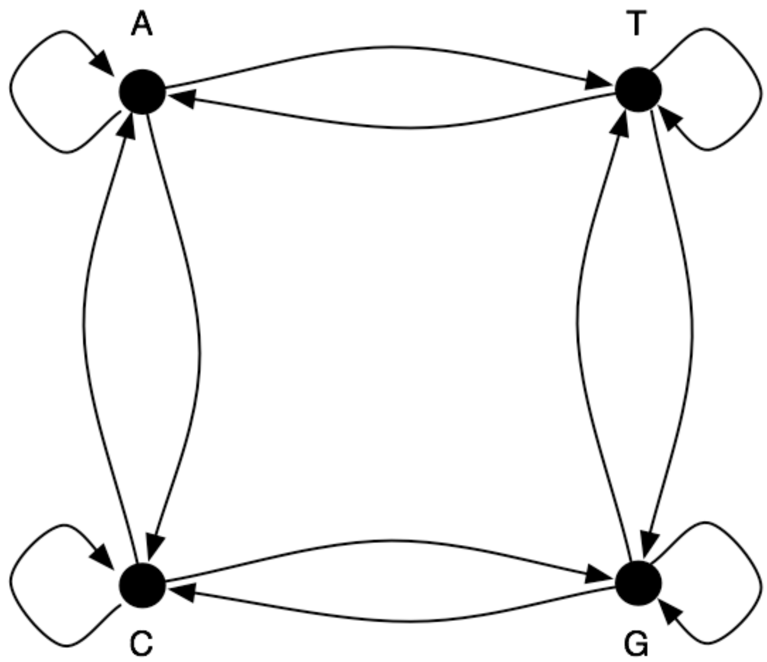
Learning a Markov chain

- Suppose the transition probabilities weren't given. How would you estimate them?



Application: bioinformatics

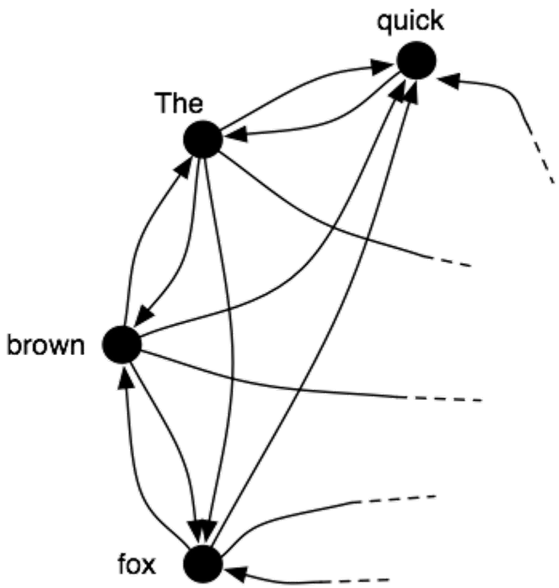
- Markov chains are used to model biological sequences
  - e.g. peptide



Application: language modeling

- A sentence is just a sequence of words
  - which we can model as a sequence of states.
- Sentence generation can be modeled as a Markov chain

```
The quick brown fox jumps
over the lazy dog.
```



Automatic sentence generation

- Random walks on the Markov chain produce sentences!
  - e.g. using a model trained on an essay of Jean Baudrillard, "The Precession of Simulacra"

> If we were to revive the fable is useless. Perhaps only the allegory of simulation is unendurable--more cruel than Artaud's Theatre of Cruelty, which was the first to practice deterrence, abstraction, disconnection, deterritorialisation, etc.; and if it were our own past. We are witnessing the end of the negative form. But nothing separates one pole from the very swing of voting "rights" to electoral "duties" that the disinvestment of the revolutionary and total strike collapses at the real and its object, as Castaneda does, etc., and to escape the spectre raised by simulation--namely that truth, reference and objective causes have ceased to exist.

By "Mark V. Shaney" and Rob Pike, 1989

Automatic sentence generation

- Random walks on the Markov chain produce sentences!
  - e.g. using a model trained on poetry

He was a light, slow, and there is a small Saturn -- away from a high flame lying in the life within it, a new dune, we are formations of caterpillars, we are formations of craziness to innocent, and as it moves it is complete different than the rising face, the cold water, even we can't see infinity is an ocean of downy treasure the welldeep pleasure of caterpillars, we are formations of the world, and what it with the ecstasy of the day is an iceberg we find ourselves on a caress mingled with sleep kill me its lights bands of subjective experience, and wonder why I had dirt a star-crystal-flower plants, made the dragon. Its neck was a novel entitled "Kaleidoscope Vision," which is hat crinkle were like fresh glass domain key - you become someone mentioned them and build in. We see the white my own rising and thunder clapping in the singularity of it, evaporating into a tree, like a long before shade.

By “Mark V. Shaney” and Justin McHale, 1994

Automatic sentence generation

- Random walks on the Markov chain produce sentences!
  - e.g. using a model trained on postings from alt.singles

When I meet someone on a professional basis, I want them to shave their arms. While at a conference a few weeks back, I spent an interesting evening with a grain of salt. I wouldn't dare take them seriously! This brings me back to the brash people who dare others to do so or not. I love a good flame argument, probably more than anyone....

By "Mark V. Shaney" and Rob Pike et al, 1984

Practical uses?

- Generating spam
- SCIgen: Generating scientific papers?!
  - “Rooter: A Methodology for the Typical Unification of Access Points and Redundancy,” *11th World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI),* 2005.
  - "I/O Automata No Longer Considered Harmful," *3rd International Symposium of Interactive Media Design*, 2005.
  - Cooperative, Compact Algorithms for Randomized Algorithms, *Applied Mathematics and Computation* (accepted but eventually rescinded)

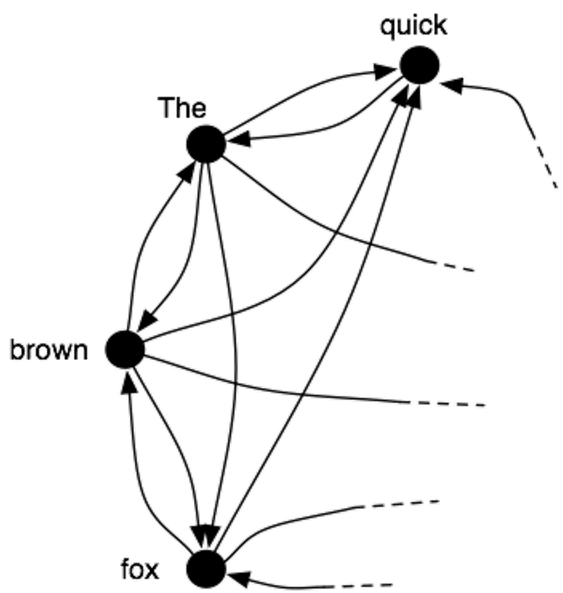
See http://pdos.csail.mit.edu/scigen/

Grammar checking

- Using a Markov model, we can compute the probability of sentences, marking low-probability ones
  - e.g.

Very low probability: `Their is the quick brown fox.`

Relatively high probability: `There is the quick brown fox.`



Hidden Markov Models (HMMs)

- A Markov Chain, but the system state is *not observable*
  - Instead there is an observable random variable, *O*, whose value probabilistically depends on the current state

- More formally, an HMM consists of:
  - Transition probabilities

$$p_{ij} = P(Q_{t+1} = j | Q_t = i)$$

  - Initial state distribution

$$w_i = P(Q_0 = i)$$

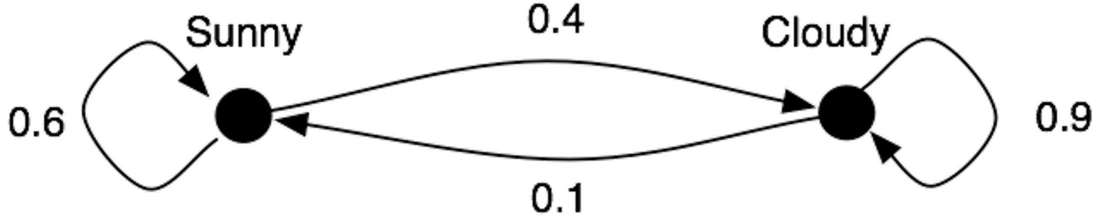
  - Emission probabilities

$$e_i(a) = P(O_t = a | Q_t = i)$$

Example

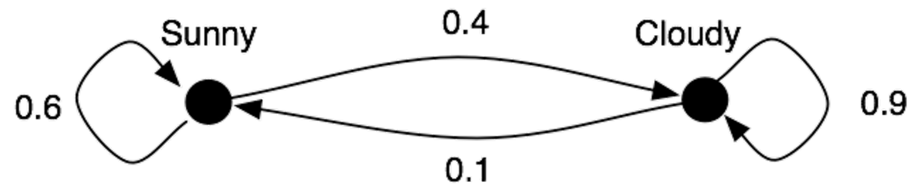
- Mary lives in Seattle, and tells the truth 80% of the time. Every day, she calls you to report the weather in Seattle.
  - It's either Sunny (S) or Cloudy (C)

- You know (based on historical data) that the weather in Seattle follows a Markov chain,



  - Also, the probability of sun on any given day is 0.2

- Mary reports that the following sequence over a 5 day period: SCSCC



- Transition probabilities $p_{SS} = P(Q_{t+1} = S|Q_t = S) = 0.6$

$p_{CS} = 0.1 \qquad p_{CC} = 0.9 \qquad p_{SC} = 0.4$

- Emission probabilities

$e_C(S) = P(O_t = S|Q_t = C) = 0.2 \qquad e_S(C) = P(O_t = C|Q_t = S) = 0.2$

$e_C(C) = P(O_t = C|Q_t = C) = 0.8 \qquad e_S(S) = P(O_t = S|Q_t = S) = 0.8$

- Initial state distribution

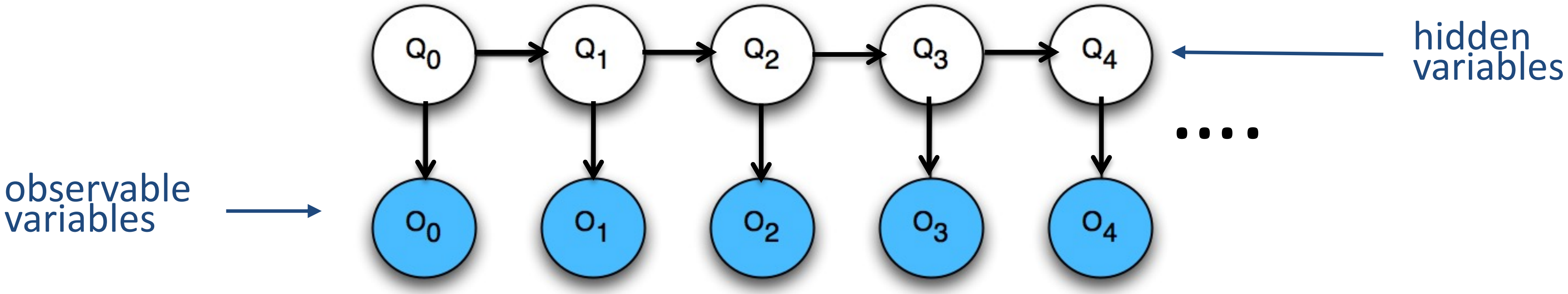
$w_S = P(Q_0 = S) = 0.2 \qquad w_C = P(Q_0 = C) = 0.8$

- Observation sequence

$$O_0 = S, O_1 = C, O_2 = S, O_3 = C, O_4 = C$$

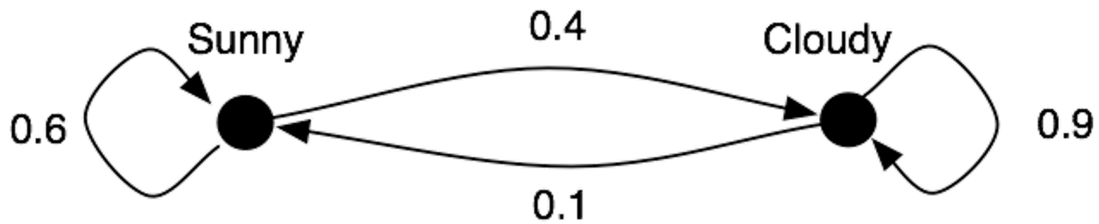
Inference on HMMs

- HMMs are just special cases of Bayes Nets!



- Intuitively, the HMM is balancing two goals:
  - maximizing emission probabilities -- finding a state sequence that agrees with the observations
  - maximizing transition probabilities -- finding a state sequence that has high likelihood according to the Markov chain

HMM inference



- Two important types of questions:
  - Given a particular observation (e.g. SCSCC), what is the distribution over the weather *on a particular day*? (Marginal inference)
  - Given a particular observation (e.g. SCSCC), what is the *most likely sequence of weather across all days*? (Maximum a posterior (MAP) inference)

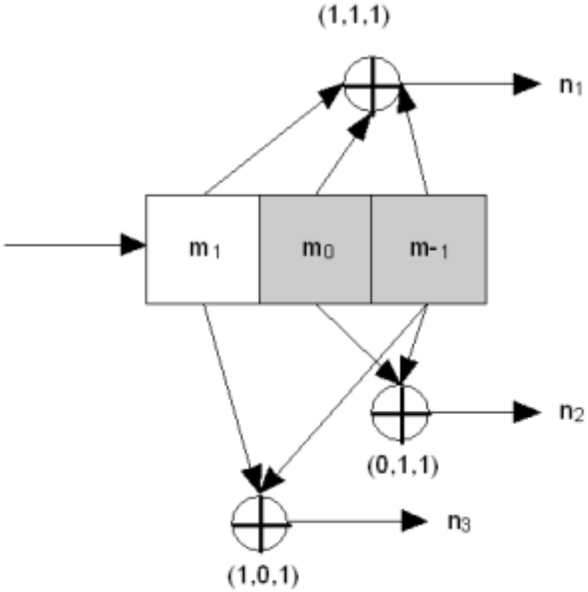
HMM inference

- How do we find the most likely state sequence, given a sequence of observations?
  - Brute force approach: Try all possible state sequences. Find the one that maximizes P(Q|O).
  - Viterbi decoding: Efficient algorithm based on dynamic programming.

Convolution and turbo codes

Andrew Viterbi

- Used extensively in wireless communications
  - Adds redundancy to a signal, so that transmission errors can be detected and corrected
    - Transmitted bits are a combination of the last *k* input bits
    - Transmitted bits are possibly corrupted by interference
    - The decoder uses Viterbi to infer the (hidden) state of the transmitter, from the (noisy) received bits



Natural Language Processing

- Statistical techniques like HMMs are very popular

“Every time I fire a linguist, my performance goes up.”

--- attributed to Fred Jelinek, 1980’s, IBM Watson

- For example: Part-of-speech (POS) tagging

She promised to back the bill

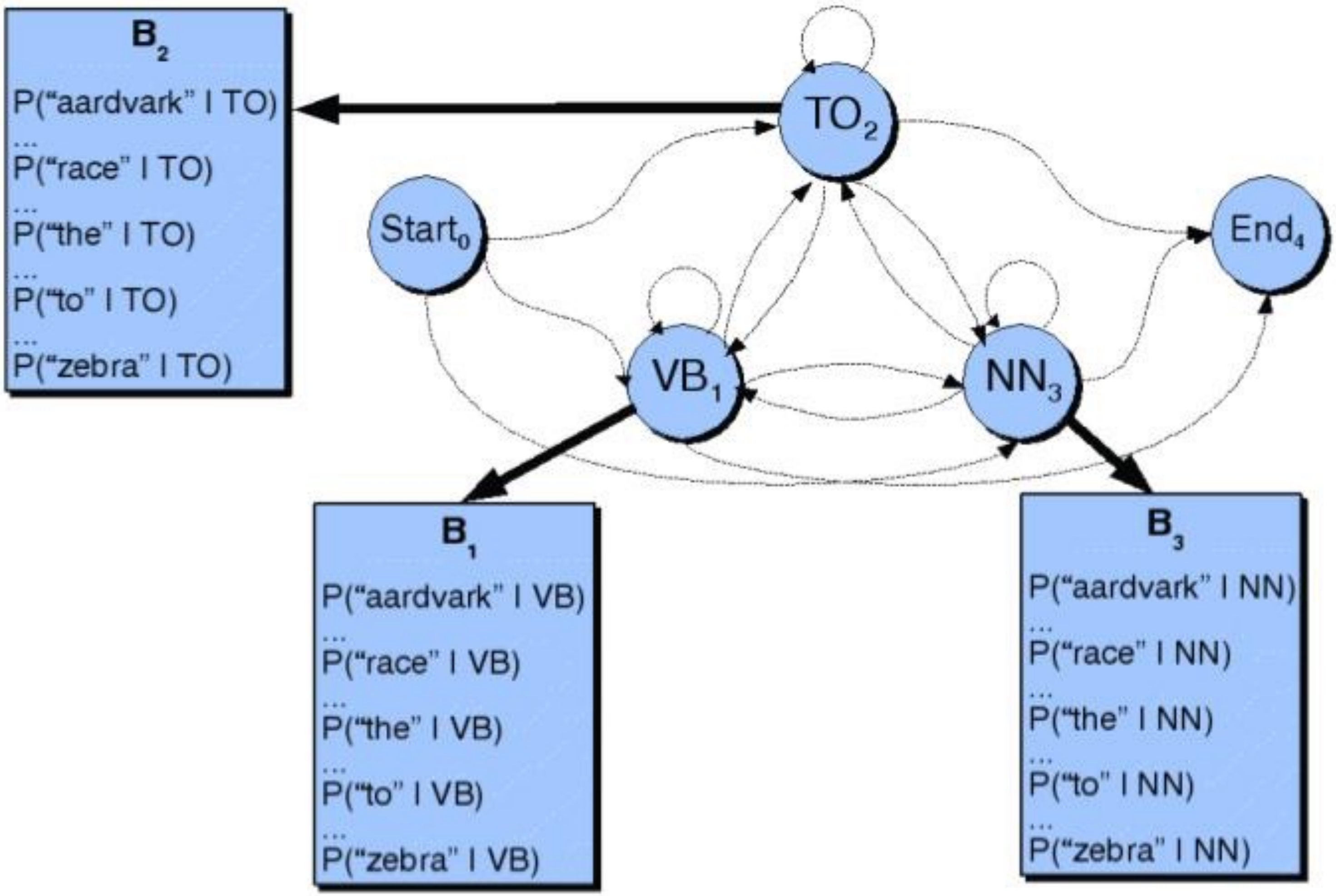
POS tagging as an HMM problem

- Observation sequence
  - sequence of words
- States
  - parts of speech (noun, verb, adjective, etc.)

- Emission probabilities
  - probability that word w is produced by state s
  - related (by Baye's law) to the probability that w has POS s

$$P(V \mid race) = \frac{Count(race\ \ is\ \ verb)}{total\ \ Count(race)} \quad \sim 0.95$$

HMM-based POS tagging



HMM-based POS tagging

- Transition probabilities

	VB	TO	NN	PPSS
<s>	.019	.0043	.041	.067
VB	.0038	.035	.047	.0070
TO	.83	0	.00047	0
NN	.0040	.016	.087	.0045
PPSS	.23	.00079	.0012	.00014

Figure 4.15 Tag transition probabilities (the a array, $p(t_i|t_{i-1})$) computed from the 87-tag Brown corpus without smoothing. The rows are labeled with the conditioning event; thus $P(PPSS|VB)$ is .0070. The symbol <s> is the start-of-sentence symbol.

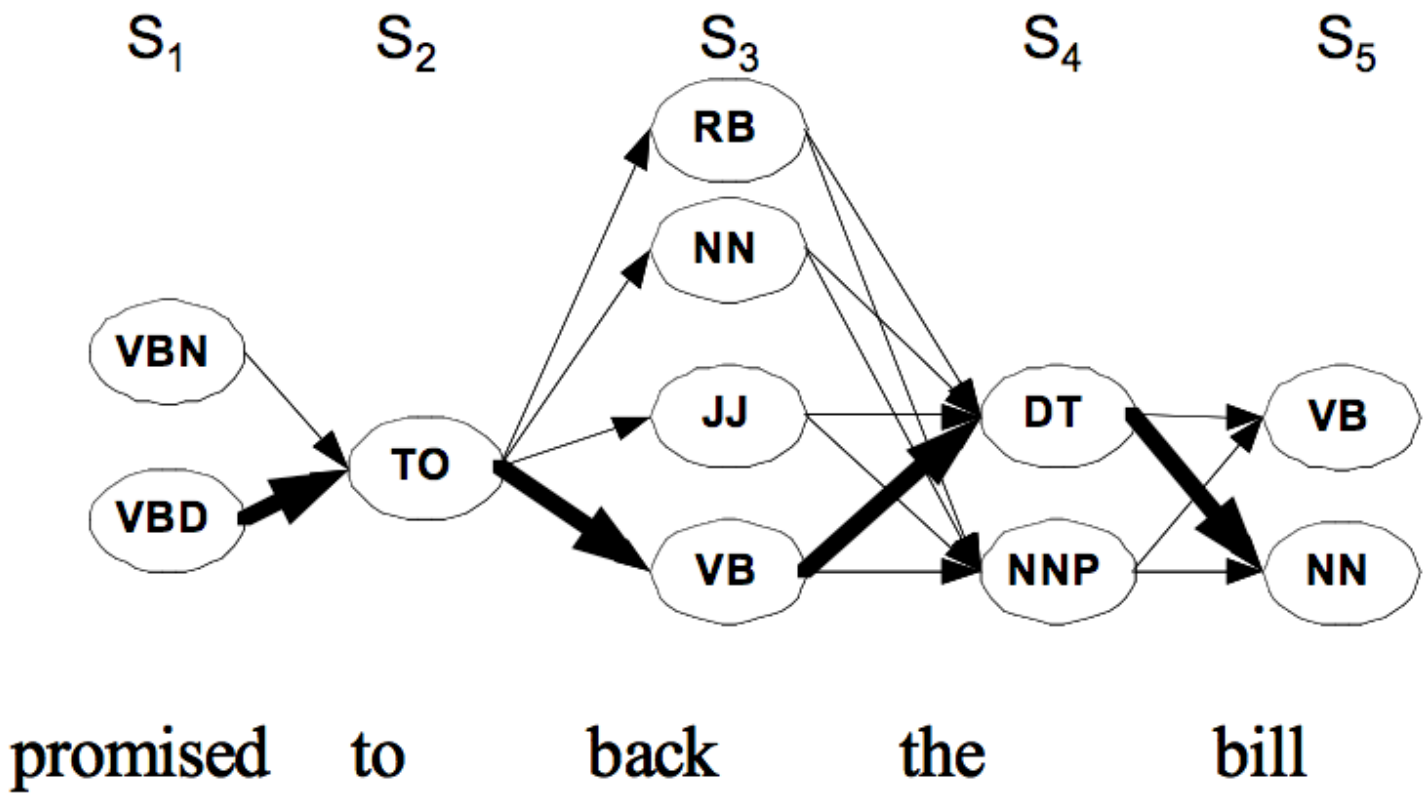
HMM-based POS tagging

- Emission probabilities

	I	want	to	race
VB	0	.0093	0	.00012
TO	0	0	.99	0
NN	0	.000054	0	.00057
PPSS	.37	0	0	0

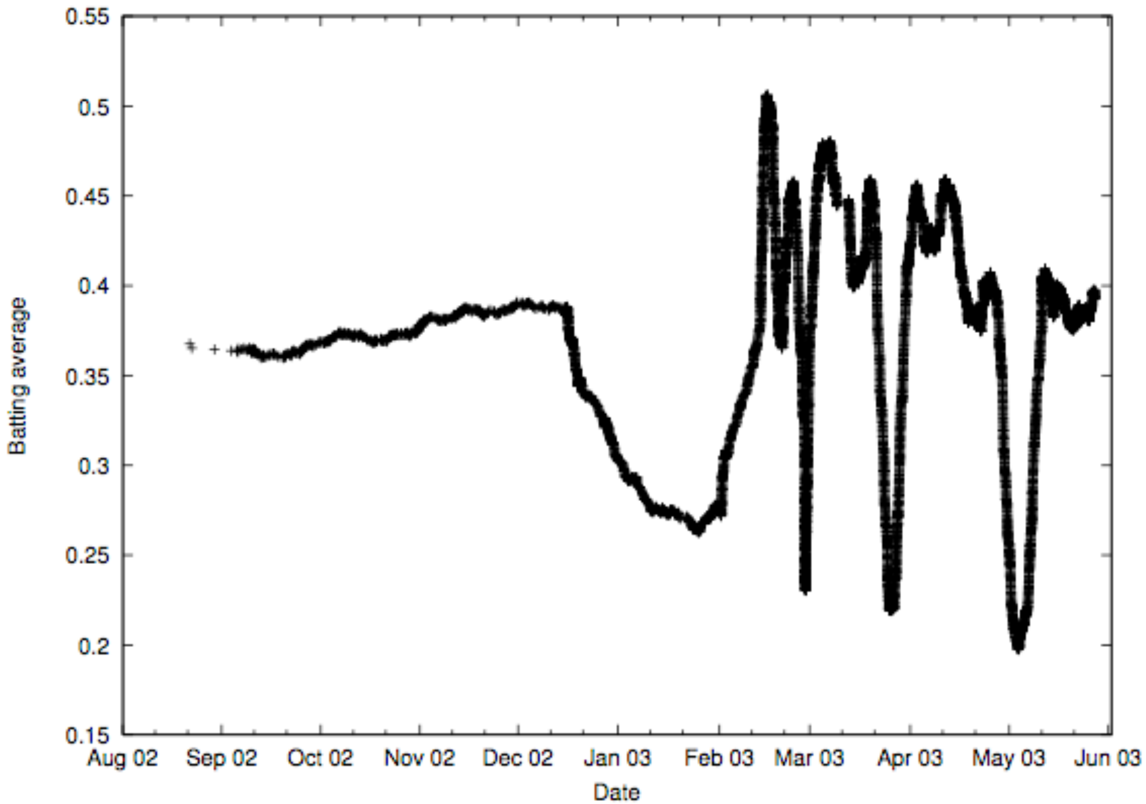
Figure 4.16 Observation likelihoods (the *b* array) computed from the 87-tag Brown corpus without smoothing.

POS decoding



Application: Analyzing noisy data

- Click-through rate for a particular webpage:



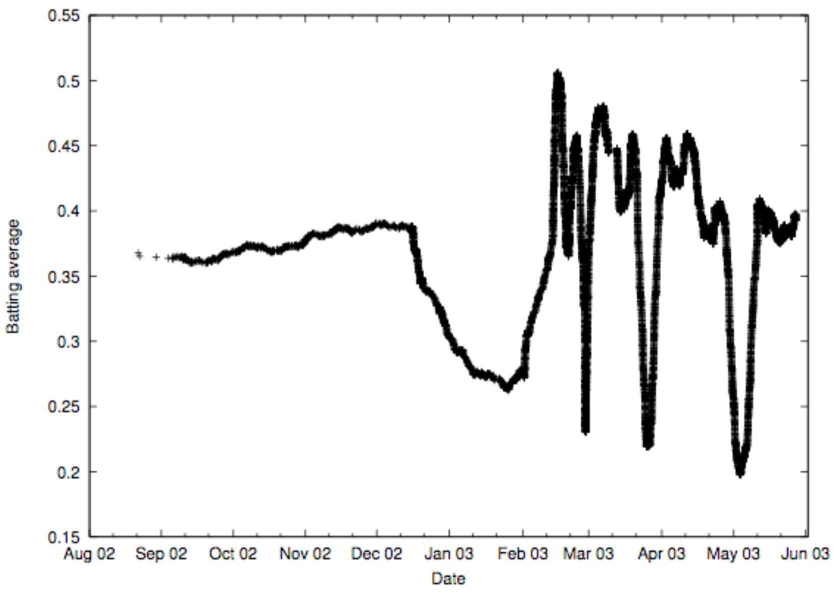
[Aizen04]

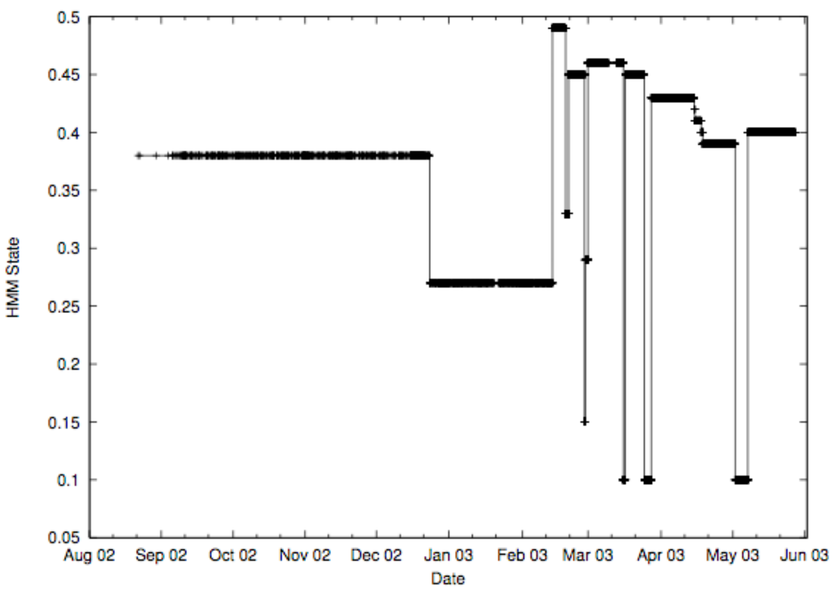
Application: Analyzing noisy data

- Model the world's interest in page as Markov chain
  - Changes with news events, etc.
  - But it is hidden -- we can't observe it directly
  - Instead we can observe raw click-through rates

- Use HMM inference to denoise the data
  - The states are the different interest levels
  - Observable variable is the empirical click-through rate
  - The Markov chain is learned from actual data
  - Viterbi gives the most likely interest level sequence

Application: Analyzing noisy data





Classifying photo streams

3:35pm

Alcatraz, SF bay?
Ellis Island, NYC?

8:03pm

Piazza San Marco, Venice?
Sather Tower, Berkeley?

9:27pm

Bay Bridge, SF bay?
Geo Wash Bridge, NYC?

Classifying photo streams

3:35pm

Alcatraz, SF bay?
~~Ellis Island, NYC?~~

8:03pm

~~Piazza San Marco, Venice?~~
Sather Tower, Berkeley?

9:27pm

Bay Bridge, SF bay?
~~Geo Wash Bridge, NYC?~~

Classifying photo streams

3:35pm

Alcatraz, SF bay?
~~Ellis Island, NYC?~~

8:03pm

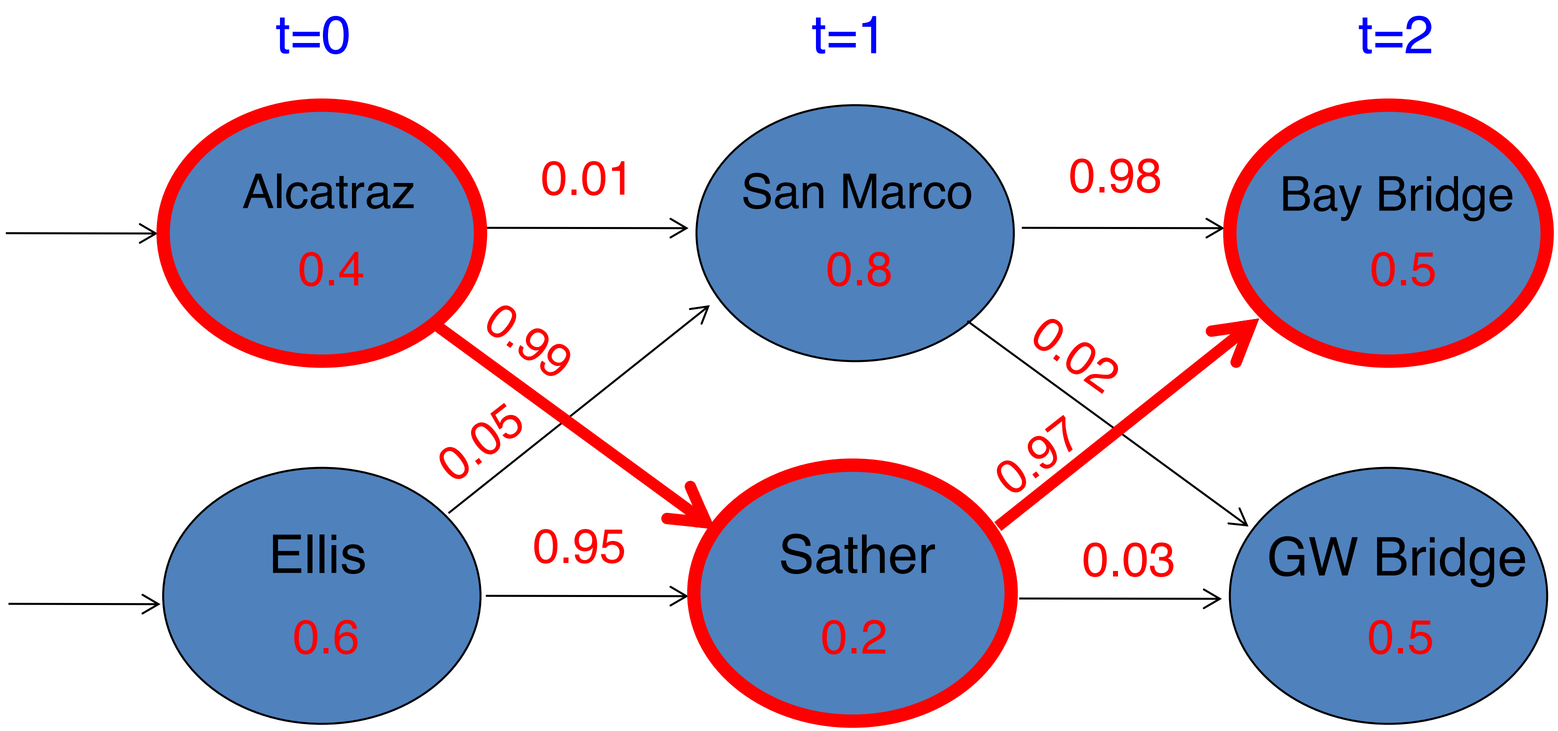
~~Piazza San Marco, Venice?~~
Sather Tower, Berkeley?

9:27pm

Bay Bridge, SF bay?
~~Geo Wash Bridge, NYC?~~

- Model as a Hidden Markov Model, do fast inference using the Viterbi algorithm

HMM decoding



Classifying photo streams with HMMs

Probabilities with individual photo classifiers:

Bathroom:	**0.931**	0.023	0.002	0.007	0.009	0.018	0.016**0**	0.073
Bedroom:	0.006	**0.734**	**0.461**	0.120	0.082	0.002	**.885**	0.018
Garage:	0.006	0.192	0.117	**0.744**	**0.746**	0.168	0.059	0.003
Living:	0.014	0.020	0.420	0.127	0.162	**0.811**	0.023	0.005
Office:	0.042	0.031	0.000	0.001	0.001	0.001	0.018	**0.901**

Probabilities after applying HMM:

Bathroom:	**0.896**	0.436	0.060	0.015	0.010	0.006	0.002	0.000
Bedroom:	0.010	0.052	0.026	0.004	0.002	0.002	0.002	0.000
Garage:	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Living:	0.079	**0.441**	**0.881**	**0.968**	**0.975**	**0.873**	0.125	0.005
Office:	0.006	0.027	0.009	0.009	0.012	0.116	**0.865**	**0.994**

HMM inference

- How do we find the most likely state sequence, given a sequence of observations?
  - Brute force approach: Try all possible state sequences. Find the one that maximizes P(Q|O).
  - Viterbi decoding: Efficient algorithm based on dynamic programming.

Viterbi decoding

- Key idea: the posterior probability of a state sequence, P(Q|O), factors nicely

$$P(Q_0 = q_0, ..., Q_T = q_T | O_0 ... O_T)$$

(Bayes' Law) $$= \frac{P(O_0 ... O_T | Q_0 = q_0 ... Q_T = q_T) P(Q_0 = q_0 ... Q_T = q_T)}{P(O_0 ... O_T)}$$

Viterbi decoding

- Key idea: the posterior probability of a state sequence, P(Q|O), factors nicely

$$P(Q_0 = q_0, ..., Q_T = q_T | O_0 ... O_T)$$

(Bayes' Law) $$= \frac{P(O_0 ... O_T | Q_0 = q_0 ... Q_T = q_T) P(Q_0 = q_0 ... Q_T = q_T)}{P(O_0 ... O_T)}$$

(denom depends only on O) $$\propto P(O_0 ... O_T | Q_0 = q_0 ... Q_T = q_T) P(Q_0 = q_0 ... Q_T = q_T)$$

Viterbi decoding

- Key idea: the posterior probability of a state sequence, P(Q|O), factors nicely

$$P(Q_0 = q_0, ..., Q_T = q_T | O_0 ... O_T)$$

(Bayes' Law) $$= \frac{P(O_0 ... O_T | Q_0 = q_0 ... Q_T = q_T) P(Q_0 = q_0 ... Q_T = q_T)}{P(O_0 ... O_T)}$$

(denom depends only on O) $$\propto P(O_0 ... O_T | Q_0 = q_0 ... Q_T = q_T) P(Q_0 = q_0 ... Q_T = q_T)$$

(O_t depends only on Q_t) $$= P(Q_0 = q_0 ... Q_T = q_T) \prod_{t=0}^{T} P(O_t | Q_t = q_t)$$

Viterbi decoding

- Key idea: the posterior probability of a state sequence, P(Q|O), factors nicely

$$P(Q_0 = q_0, ..., Q_T = q_T | O_0 ... O_T)$$

(Bayes' Law)
$$= \frac{P(O_0 ... O_T | Q_0 = q_0 ... Q_T = q_T) P(Q_0 = q_0 ... Q_T = q_T)}{P(O_0 ... O_T)}$$

(denom depends only on O)
$$\propto P(O_0 ... O_T | Q_0 = q_0 ... Q_T = q_T) P(Q_0 = q_0 ... Q_T = q_T)$$

(O_t depends only on Q_t)
$$= P(Q_0 = q_0 ... Q_T = q_T) \prod_{t=0}^{T} P(O_t | Q_t = q_t)$$

(Markov property: Q_{t+1} depends Only on Q_t)
$$= P(Q_0 = q_0) \prod_{t=0}^{T-1} P(Q_{t+1} = q_{t+1} | Q_t = q_t) \prod_{t=0}^{T} P(O_t | Q_t = q_t)$$

Viterbi decoding

- Based on dynamic programming
  - Let $v_i(t)$ be the probability of the most probable path ending at state *i* at time *t*,

$$v_i(t) = \max_{q_0 \cdots q_{t-1}} P(Q_0 = q_0, \ldots, Q_{t-1} = q_{t-1}, Q_t = i | O_0, O_1, \ldots, O_t)$$

  - Then we can recursively find the probability of the most probable path ending at state j at time t+1,

$$v_j(t+1) = e_j(O_{t+1}) \max_{1 \leq i \leq N} v_i(t) p_{ij}$$

Probability of transition from state i to j

Probability that system is in state j at time t+1 (Q_{t+1}=j)

Probability of observing O_{t+1} given that system is in state j at time t+1

Max over all possible states at time t

Probability that system in state i at time t

Next time

- Finish Viterbi, talk about Constraint Satisfaction Problems