

Administrivia

- ◆ Mini-project 2 (**really**) due today
 - Turn in a printout of your work at the end of the class
- ◆ **Project presentations**
 - **April 23** (Thursday next week) and **28** (Tuesday the week after)
 - Order will be randomized so be prepared to present on either day
 - **8 mins** total for each group (set aside 1-2 mins for questions)
 - Problem statement, approach, preliminary results, future work
 - How to give a good talk: <http://www.cs.berkeley.edu/~jrs/speaking.html>
- ◆ **Final report**
 - Maximum of 8 pages in NIPS paper format (word + latex style files)
 - <https://nips.cc/Conferences/2014/PaperInformation/StyleFiles>
 - Writeup due on **May 3 6** (submit pdf via Moodle)
 - Hard deadline — I've to submit your grades to the University

Hidden Markov Models

Subhansu Maji

CMPSCI 689: Machine Learning

16 April 2015

Overview of ML so far ...

- ◆ **Supervised learning**
 - decision trees
 - k nearest neighbor
 - perceptrons (+ kernels)
 - neural networks (+ convolution)
- ◆ **Learning is:**
 - optimization
 - density estimation
 - ... with known labels
- ◆ **Learning is hard**
 - bias-variance tradeoff
 - ensembles reduce variance
- ◆ **Learning is possible**
 - boosting weak learners
- ◆ **Unsupervised learning**
 - k-means
 - PCA (+ kernels)
 - spectral methods
 - mean shift
- ◆ **Learning is:**
 - optimization
 - density estimation
 - ... with hidden "labels"
- ◆ **EM:** a general technique to solve hidden variable problem
- ◆ **Reinforcement learning**
 - labels come from experience
 - guest lecture by Kevin Spiteri next Tuesday

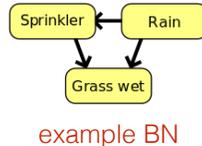
Reasoning over time

- ◆ Often, we want to reason about a **sequence** of **observations**
 - Speech recognition
 - Robot localization
 - User attention
 - Medical monitoring
 - Weather forecasting
- ◆ Need to introduce time in our models
- ◆ Basic approach: **Hidden Markov models (HMMs)**
- ◆ More general approach: **Dynamic Bayes Nets (DBNs)**

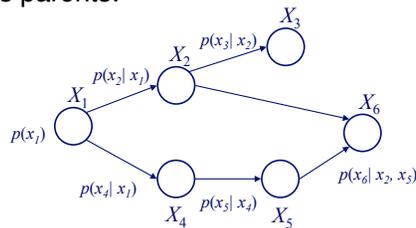
Bayes network — a quick intro

- ◆ A way of specifying **conditional independences**
- ◆ A **Bayes Network (BN)** is a **directed acyclic graph (DAG)**
- ◆ Nodes are random variables
- ◆ A node's distribution only depends on its parents
- ◆ Joint distribution decomposes:

$$p(x) = \prod_i p(x_i | \text{Parents}_i)$$

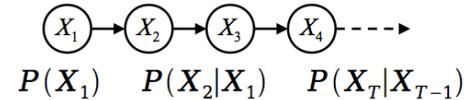


- ◆ A node's value is **conditionally independent** of everything else given the value of its parents:



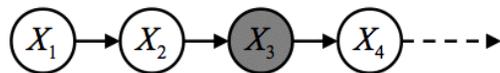
Markov models

- ◆ A **Markov model** is a chain-structured BN
 - Each node is identically distributed (**stationarity**)
 - Value of X at a given time t is called the **state**
 - As a BN:



- ◆ Parameters of the model
 - **Transition probabilities** or **dynamics** that specify how the state evolves over time
 - The **initial probabilities** of each state

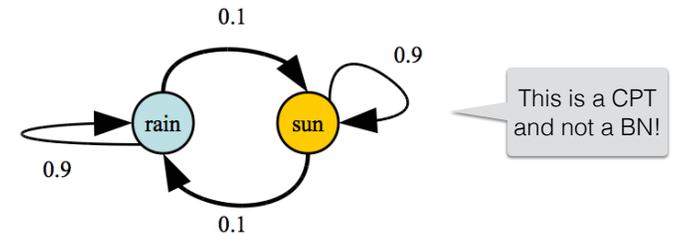
Conditional independence



- ◆ Basic **conditional independence**:
 - Past and future independent of the present
 - Each time step only depends on the previous
 - This is called the (**first order**) **Markov property**
- ◆ Note that the chain is just a (growing) BN
 - We can always use generic BN reasoning on it (if we truncate the chain)

Markov model: example

- ◆ Weather:
 - **States:** X = {rain, sun}
 - **Transitions:**



- **Initial distribution:** 1.0 sun
- **Question:** What is the **probability distribution** after one step?

$$P(X_2=\text{sun}) = P(X_2=\text{sun}|X_1=\text{sun})P(X_1=\text{sun}) + P(X_2=\text{sun}|X_1=\text{rain})P(X_1=\text{rain})$$

$$= 0.9 \times 1.0 + 0.1 \times 0.0$$

$$= 0.9$$

Markov model: example

- ◆ **Text synthesis** — create plausible looking poetry, love letters, term papers, etc.
- ◆ Typically a **higher order** Markov model
- ◆ Sample **word** w_t based on the **previous n words** i.e.:
 - $w_t \sim P(w_t | w_{t-1}, w_{t-2}, \dots, w_{t-n})$
 - These **probability tables** can be computed from lots of text
- ◆ Examples of **text synthesis** [A.K. Dewdney, Scientific American 1989]
 - “As I’ve commented before, really relating to someone involves standing next to impossible.”
 - “One morning I shot an elephant in my arms and kissed him.”
 - “I spent an interesting evening recently with a grain of salt”

Slide from Alyosha Efros, ICCV 1999

Mini-forward algorithm

- ◆ **Question:** probability of being in a state x at a time t ?
- ◆ **Slow answer:**
 - Enumerate all sequences of length t with end in s
 - Add up their probabilities:

$$P(X_t = \text{sun}) = \sum_{x_1, \dots, x_{t-1}} P(x_1, \dots, x_{t-1}, \text{sun})$$

$$P(X_1 = \text{sun})P(X_2 = \text{sun}|X_1 = \text{sun}) \dots P(X_t = \text{sun}|X_{t-1} = \text{sun})$$

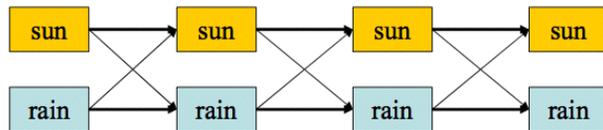
$$P(X_1 = \text{sun})P(X_2 = \text{rain}|X_1 = \text{sun}) \dots P(X_t = \text{sun}|X_{t-1} = \text{sun})$$

⋮

$$O(2^{t-1})$$

Mini-forward algorithm

- ◆ **Better way:** cached incremental belief updates
 - (GM folks: this is an instance of variable elimination)



$$P(x_1) = \text{known}$$

$$P(x_t) = \sum_{x_{t-1}} P(x_{t-1})P(x_t|x_{t-1})$$

forward simulation

Example

- ◆ From initial observation of **sun**

$$\begin{matrix} \langle \begin{matrix} 1.0 \\ 0.0 \end{matrix} \rangle & \langle \begin{matrix} 0.9 \\ 0.1 \end{matrix} \rangle & \langle \begin{matrix} 0.82 \\ 0.18 \end{matrix} \rangle & \longrightarrow & \langle \begin{matrix} 0.5 \\ 0.5 \end{matrix} \rangle \\ P(X_1) & P(X_2) & P(X_3) & & P(X_\infty) \end{matrix}$$

- ◆ From initial observation of **rain**

$$\begin{matrix} \langle \begin{matrix} 0.0 \\ 1.0 \end{matrix} \rangle & \langle \begin{matrix} 0.1 \\ 0.9 \end{matrix} \rangle & \langle \begin{matrix} 0.18 \\ 0.82 \end{matrix} \rangle & \longrightarrow & \langle \begin{matrix} 0.5 \\ 0.5 \end{matrix} \rangle \\ P(X_1) & P(X_2) & P(X_3) & & P(X_\infty) \end{matrix}$$

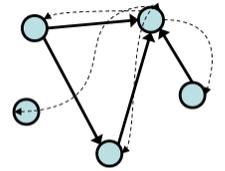
Stationary distribution

- ◆ If we simulate the chain long enough
 - ▶ What happens?
 - ▶ Uncertainty accumulates
 - ▶ Eventually, we have no idea what the state is!
- ◆ **Stationary distributions:**
 - ▶ For most chains, the distribution we end up in is independent of the initial distribution (but not always uniform!)
 - ▶ This distribution is called the **stationary distribution** of the chain
 - ▶ Usually, can only predict a short time out

Web link analysis

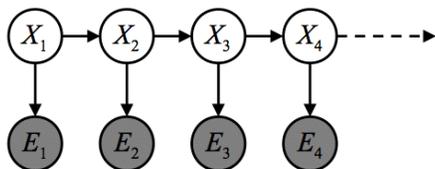
PageRank over a web graph

- ◆ Each web page is a state
- ◆ Initial distribution: uniform over pages
- ◆ Transitions:
 - ▶ With **probability** c , uniform jump to a random page (dotted lines)
 - ▶ With **probability** $1-c$, follow a random outlink (solid lines)
- ◆ **Stationary distribution**
 - ▶ Will spend more time on highly reachable pages
 - ▶ E.g. many ways to get to the Acrobat Reader download page
 - ▶ Somewhat robust to link spam (but not immune)
 - ▶ Google 1.0 returned the set of pages containing all your keywords in decreasing rank, now all search engines use link analysis along with many other factors

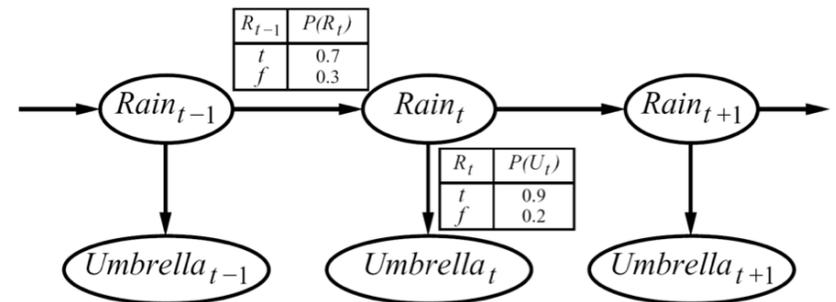


Hidden Markov Models

- ◆ **Markov chains** not so useful for most agents
 - ▶ Eventually you don't know anything anymore
 - ▶ Need observations to update your beliefs
- ◆ **Hidden Markov Models (HMMs)**
 - ▶ Underlying **Markov chain** over states S
 - ▶ You observe outputs (effects) at each time step
 - ▶ As a Bayes net:



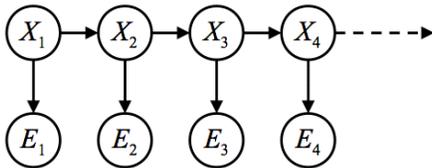
Example



- ◆ An **HMM** is defined by:
 - ▶ **Initial distribution:** $P(X_1)$
 - ▶ **Transitions:** $O(X_t | X_{t-1})$
 - ▶ **Emissions:** $P(E | X)$

Conditional independence

- ◆ HMMs have two important **independence properties**:
 - Markov hidden process, future depends on past via the present
 - Current observations independent of all else given the current state



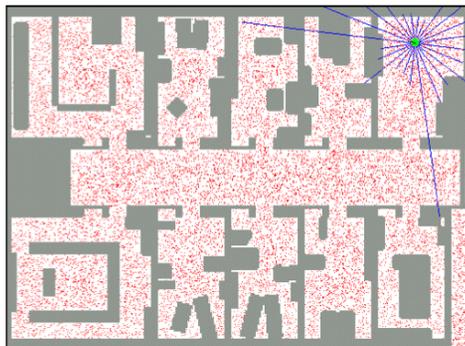
- ◆ Quiz: does this mean that the observations are independent?
 - No, correlated by the hidden state

Real HMM examples

- ◆ Speech recognition HMMs:
 - Observations are acoustic signals (continuous valued)
 - States are specific positions in specific words (so, tens of thousands)
- ◆ Machine translation HMMs:
 - Observations are words (tens of thousands)
 - States are translation options
- ◆ Robot tracking HMMs:
 - Observations are range readings (continuous)
 - States are positions on a map (continuous)

Filtering states

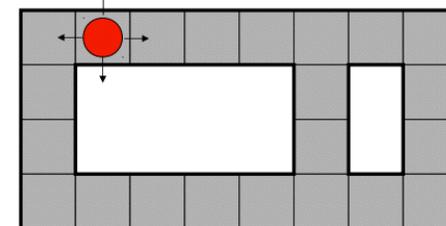
- ◆ Filtering is the task of tracking the distribution $B(X)$ (the belief state)
- ◆ We start with $B(X)$ in the initial setting, usually uniform
- ◆ As time passes, or we get observations we update $B(X)$



Example: Robot localization

Sensor model: can sense if each side has a wall or not (never more than 1 mistake)

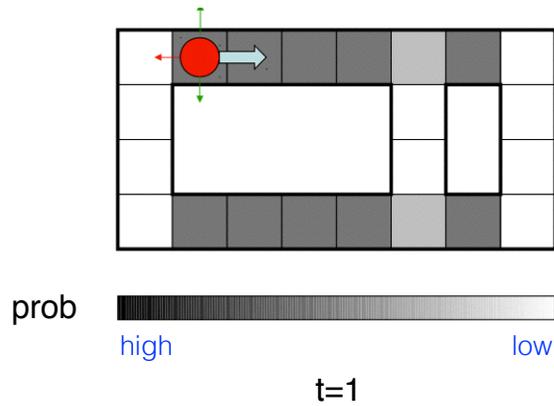
Motion model: may not execute action with a small probability



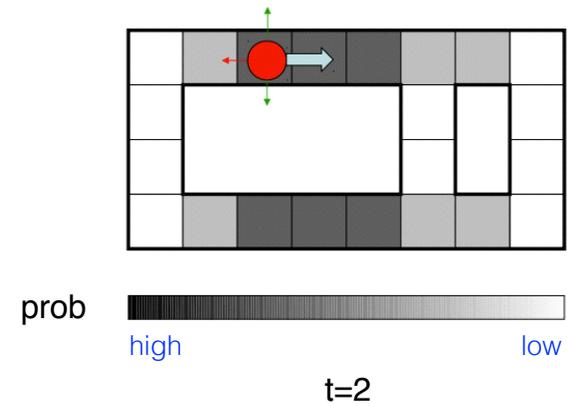
t=0

Example from Michael Pfeiffer

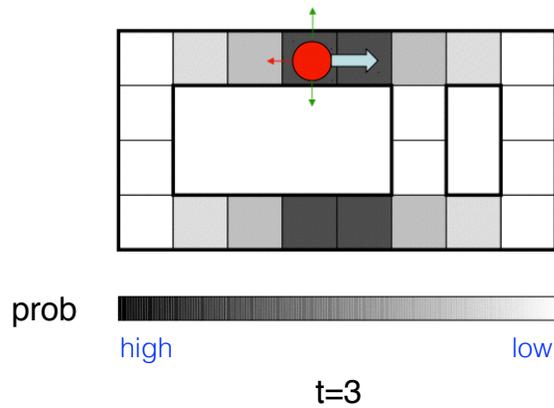
Example: Robot localization



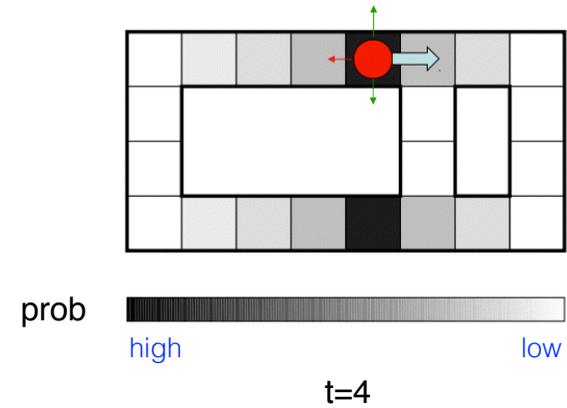
Example: Robot localization



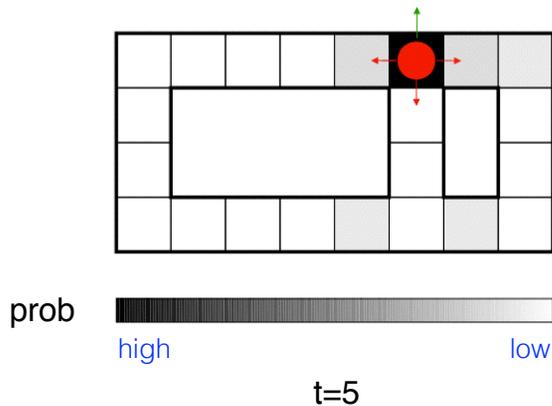
Example: Robot localization



Example: Robot localization



Example: Robot localization



Passage of time

- ◆ Assume we have a current belief state $P(X | \text{evidence to date})$

$$B(X_t) = P(X_t | e_{1:t})$$

- ◆ Then, after one time step passes:

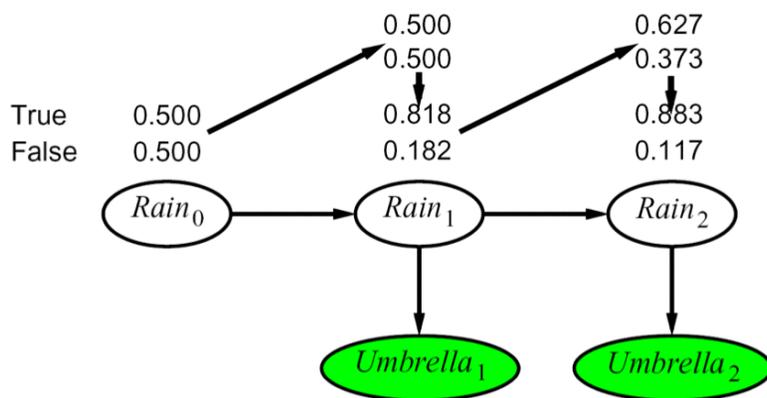
$$P(X_{t+1} | e_{1:t}) = \sum_{x_t} P(X_{t+1} | x_t) P(x_t | e_{1:t})$$

- ◆ Or, compactly:

$$B'(X_{t+1}) = \sum_{x_t} P(X' | x) B(x_t)$$

- ◆ Basic idea: beliefs get “pushed” through the transitions
 - With the “B” notation, we have to be careful about what time step t the belief is about, and what evidence it includes

Example HMM



Most likely explanation

- ◆ **Question:** most likely sequence ending in x at time t ?
 - E.g., if sun on day 4, what's the most likely sequence?
 - Intuitively: probably sun on all four days

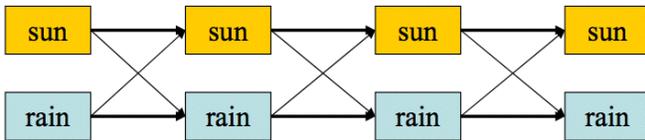
- ◆ **Slow answer:** enumerate and score

$$\text{most likely sequence} \leftarrow \arg \max_{x_1, \dots, x_{t-1}} P(x_1, \dots, x_{t-1}, \text{sun})$$

- **Complexity** $O(2^{t-1})$

Mini-Viterbi algorithm

- ◆ **Better answer:** cached incremental updates



- ◆ **Define:**

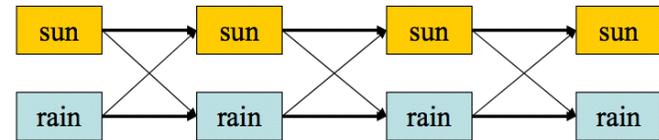
$$m_t[x] = \max_{x_{1:t-1}} P(x_{1:t-1}, x)$$

$$a_t[x] = \arg \max_{x_{1:t-1}} P(x_{1:t-1}, x)$$

- ◆ Read of the **best sequence** from the m and a vectors

Mini-Viterbi algorithm

- ◆ **Better answer:** cached incremental updates



$$m_t[x] = \max_{x_{1:t-1}} P(x_{1:t-1}, x)$$

$$= \max_{x_{1:t-1}} P(x_{1:t-1})P(x|x_{t-1})$$

$$= \max_{x_{t-1}} P(x|x_{t-1}) \max_{x_{1:t-2}} P(x_{1:t-2}, x_{t-1})$$

$$= \max_{x_{t-1}} P(x|x_{t-1})m_{t-1}[x_{t-1}]$$

$$m_1[x] = P(x)$$

Viterbi algorithm

- ◆ **Question:** what is the most likely state sequence given the observations?

- ▶ **Slow answer:** enumerate all possibilities
- ▶ **Better answer:** cached incremental version

$$x_{1:t}^* = \arg \max_{x_{1:t}} P(x_{1:t}|e_{1:t})$$

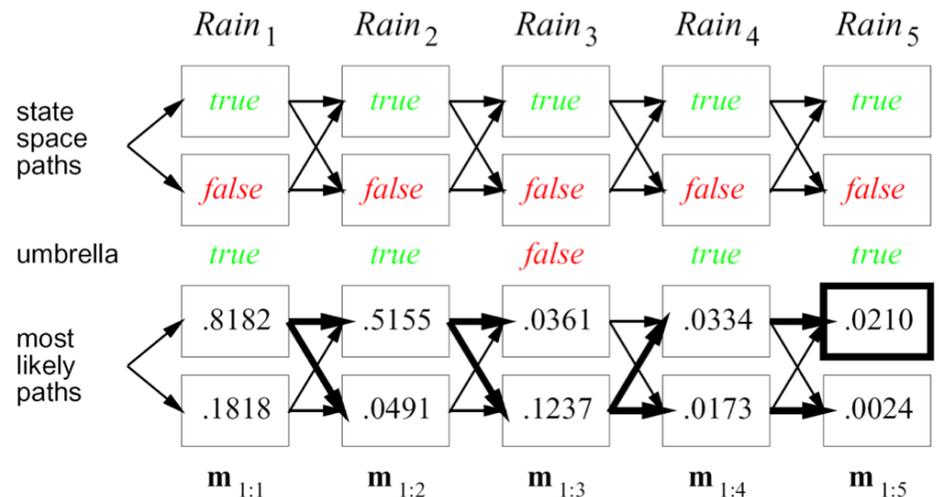
$$m_t[x_t] = \max_{x_{1:t-1}} P(x_{1:t-1}, x_t, e_{1:t})$$

$$= \max_{x_{1:t-1}} P(x_{1:t-1}, e_{1:t-1})P(x_t|x_{t-1})P(e_t|x_t)$$

$$= P(e_t|x_t) \max_{x_{t-1}} P(x_t|x_{t-1}) \max_{x_{1:t-2}} P(x_{1:t-2}, e_{1:t-2})$$

$$= P(e_t|x_t) \max_{x_{t-1}} P(x_t|x_{t-1})m_{t-1}[x_{t-1}]$$

Example



Learning discrete HMMs: I

- ◆ Given sequences of **training data** (X^1, Y^1) (X^2, Y^2) ... (X^N, Y^N) — the **hidden states** are known
- ◆ **Maximum-likelihood** parameters estimation is easy:
 - **Transition probabilities:**

$$P_t(a|b) = \frac{\sum_{n,t} [X_t^n = a, X_{t-1}^n = b]}{\sum_{n,t,a} [X_t^n = a, X_{t-1}^n = b]}$$

- **Emission probabilities:**

$$P_e(a|b) = \frac{\sum_{n,t} [Y_t^n = a, X_t^n = b]}{\sum_{n,t,a} [Y_t^n = a, X_t^n = b]}$$

- **Initial probabilities:**

$$\pi(a) = \frac{\sum_n [X_1^n = a]}{\sum_{n,a} [X_1^n = a]}$$

Learning discrete HMMs: II

- ◆ Given sequences of **training data** (Y^1, Y^2, \dots, Y^N) — no hidden states
- ◆ Use EM algorithm!

- Randomly initialize **parameters** of the HMM
- **E step:** Compute posterior probabilities

$$q(X_t^n = a, X_{t-1}^n = b) \leftarrow p(X_t^n = a, X_{t-1}^n = b | \mathcal{D}, \Theta)$$
$$q(X_t^n = a) \leftarrow p(X_t^n = a | \mathcal{D}, \Theta)$$

dynamic programming

- **M step:** Update parameters of the HMM

- **Transition probabilities:** $P_t(a|b) = \frac{\sum_{n,t} q(X_t^n = a, X_{t-1}^n = b)}{\sum_{n,t,a} q(X_t^n = a, X_{t-1}^n = b)}$

- **Emission probabilities:** $P_e(a|b) = \frac{\sum_{n,t} [Y_t^n = a] q(X_t^n = b)}{\sum_{n,t,a} [Y_t^n = a] q(X_t^n = b)}$

- **Initial probabilities:** $\pi(a) = \frac{\sum_n q(X_1^n = a)}{\sum_{n,a} q(X_1^n = a)}$

Summary

- ◆ Hidden Markov Models (HMMs) for modeling sequential data
 - Parameters for a discrete HMM: transition probabilities, emission probabilities, initial state probabilities
- ◆ Inference questions —
 - What is the **belief state** given observations?
 - What is the **most likely explanation** given the observations? (Viterbi)
 - All of these can be computed using **dynamic programming** in **O(S²T)** time compared to brute-force enumeration that needs **O(S^T)** time
- ◆ Learning HMMs
 - Known hidden states — ML estimates of parameters are proportional to the counts
 - Unknown hidden states — use EM (Baum-Welch 1960)

Slides credit

- ◆ Many of the slides are adapted from those by Hal Daume III, Dan Klein, Stuart Russell or Andrew Moore
- ◆ The “sprinkler, rain, grass” figure is from Wikipedia’s discussion on Bayes networks http://en.wikipedia.org/wiki/Bayesian_network
- ◆ The robot navigation example is from Michael Pfeiffer