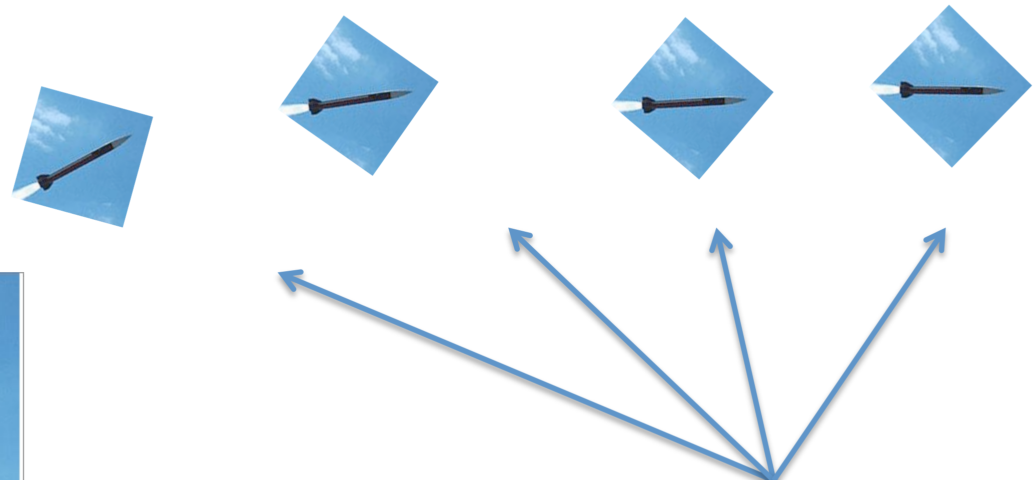
Hidden Markov models
Lecture 22

David Sontag
New York University

Example application: Tracking



Radar

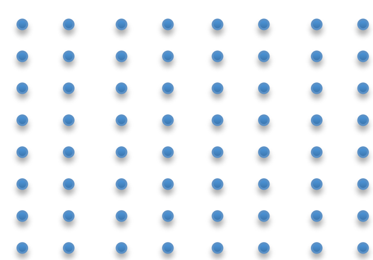
Observe noisy measurements of missile location: Y_1, Y_2, …

Where is the missile **now**? Where will it be in 10 seconds?

Probabilistic approach

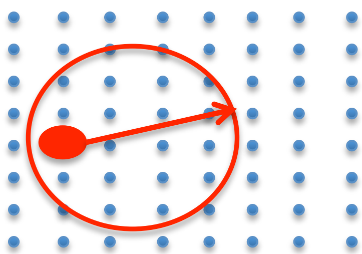
- Our measurements of the missile location were $Y_1, Y_2, \ldots, Y_n$
- Let X_t be the *true* <missile location, velocity> at time t
- To keep this simple, suppose that everything is discrete, i.e. X_t takes the values 1, …, k

Grid the space:



Probabilistic approach

- First, we specify the *conditional* distribution Pr(X_t | X_{t-1}):



From basic physics, we can bound the distance that the missile can have traveled

- Then, we specify Pr(Y_t | X_t=<(10,20), 200 mph toward the northeast>):

With probability ½, Y_t = X_t (ignoring the velocity). Otherwise, Y_t is a uniformly chosen grid location

Hidden Markov models

1960's

- Assume that the **joint** distribution on $X_{1,}$ X_2, …, X_n and Y_1, Y_2, …, Y_n factors as follows:

$$\Pr(x_1, \ldots x_n, y_1, \ldots, y_n) = \Pr(x_1) \Pr(y_1 \mid x_1) \prod_{t=2}^{n} \Pr(x_t \mid x_{t-1}) \Pr(y_t \mid x_t)$$

- To find out where the missile is *now*, we do **marginal inference**:

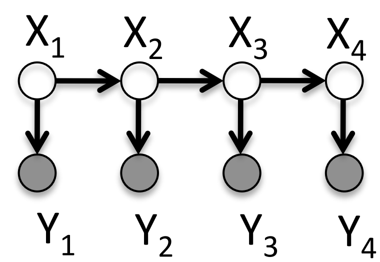
$$\Pr(x_n \mid y_1, \ldots, y_n)$$

- To find the most likely *trajectory*, we do **MAP (maximum a posteriori) inference**:

$$\arg\max_{\mathbf{x}} \Pr(x_1, \ldots, x_n \mid y_1, \ldots, y_n)$$

Inference

- Recall, to find out where the missile is now, we do marginal inference: $\Pr(x_n \mid y_1, \ldots, y_n)$



- How does one **compute** this?

- Applying rule of conditional probability, we have:

$$\Pr(x_n \mid y_1, \ldots, y_n) = \frac{\Pr(x_n, y_1, \ldots, y_n)}{\Pr(y_1, \ldots, y_n)}$$

- Naively, would seem to require k^{n-1} summations,

$$\Pr(x_n, y_1, \ldots, y_n) = \sum_{x_1, \ldots, x_{n-1}} \Pr(x_1, \ldots, x_n, y_1, \ldots, y_n)$$

Is there a more efficient algorithm?

Marginal inference in HMMs

- Use **dynamic programming**

$$\Pr(A=a) = \sum_b \Pr(B=b, A=a)$$

$$\Pr(x_n, y_1, \ldots, y_n) = \sum_{x_{n-1}} \Pr(x_{n-1}, x_n, y_1, \ldots, y_n)$$

$$\Pr(\vec{A}=\vec{a}, \vec{B}=\vec{b}) = \Pr(\vec{A}=\vec{a}) \Pr(\vec{B}=\vec{b} \mid \vec{A}=\vec{a})$$

$$= \sum_{x_{n-1}} \Pr(x_{n-1}, y_1, \ldots, y_{n-1}) \Pr(x_n, y_n \mid x_{n-1}, y_1, \ldots, y_{n-1})$$

Conditional independence in HMMs

$$= \sum_{x_{n-1}} \Pr(x_{n-1}, y_1, \ldots, y_{n-1}) \Pr(x_n, y_n \mid x_{n-1})$$

$$\Pr(A=a, B=b) = \Pr(A=a) \Pr(B=b \mid A=a)$$

$$= \sum_{x_{n-1}} \Pr(x_{n-1}, y_1, \ldots, y_{n-1}) \Pr(x_n \mid x_{n-1}) \Pr(y_n \mid x_n, x_{n-1})$$

Conditional independence in HMMs

$$= \sum_{x_{n-1}} \Pr(x_{n-1}, y_1, \ldots, y_{n-1}) \Pr(x_n \mid x_{n-1}) \Pr(y_n \mid x_n)$$

- For n=1, initialize $\Pr(x_1, y_1) = \Pr(x_1) \Pr(y_1 \mid x_1)$
- Total running time is O(nk) – linear time! Easy to do **filtering**

MAP inference in HMMs

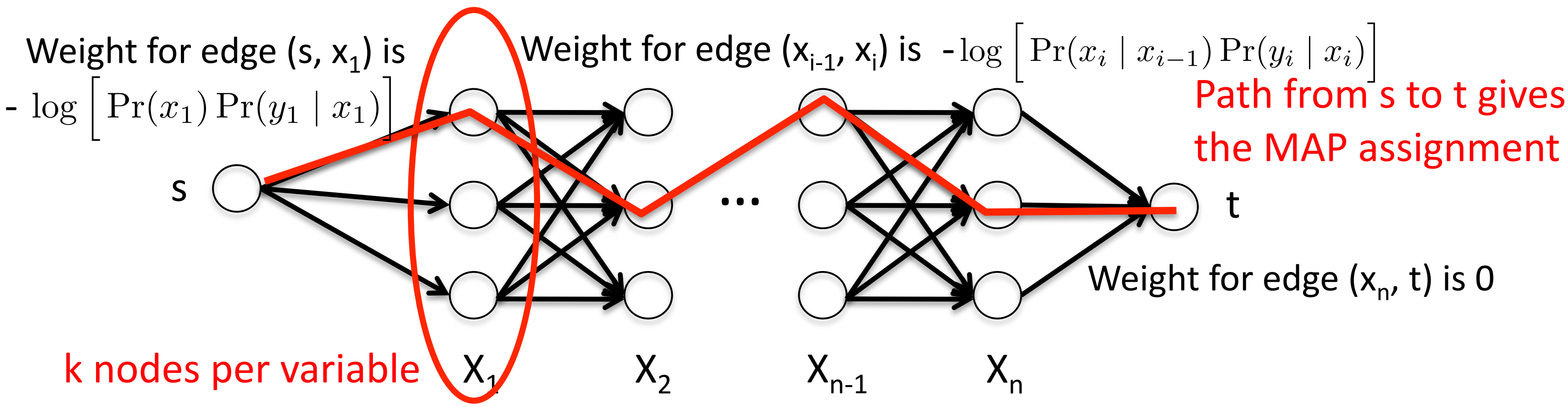
- MAP inference in HMMs can *also* be solved in linear time!

$$\arg\max_{\mathbf{x}} \Pr(x_1, \dots x_n \mid y_1, \dots, y_n) = \arg\max_{\mathbf{x}} \Pr(x_1, \dots x_n, y_1, \dots, y_n)$$

$$= \arg\max_{\mathbf{x}} \log \Pr(x_1, \dots x_n, y_1, \dots, y_n)$$

$$= \arg\max_{\mathbf{x}} \quad \log \Big[\Pr(x_1) \Pr(y_1 \mid x_1) \Big] + \sum_{i=2}^{n} \log \Big[\Pr(x_i \mid x_{i-1}) \Pr(y_i \mid x_i) \Big]$$

- Formulate as a shortest paths problem



Called the Viterbi algorithm

Applications of HMMs

- Speech recognition
  - Predict phonemes from the sounds forming words (i.e., the actual signals)
- Natural language processing
  - Predict parts of speech (verb, noun, determiner, etc.) from the words in a sentence
- Computational biology
  - Predict intron/exon regions from DNA
  - Predict protein structure from DNA (locally)
- And many many more!