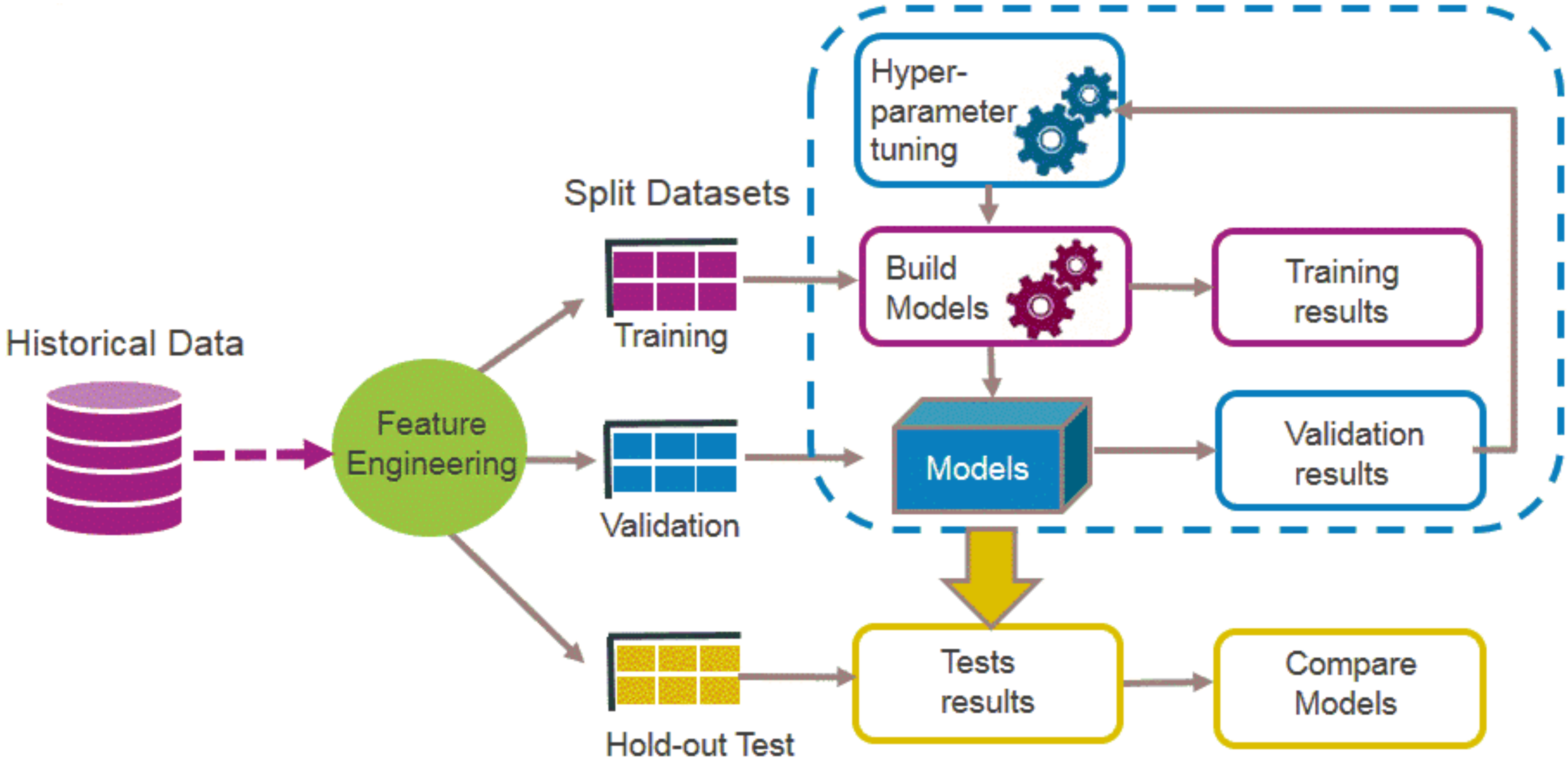
Hidden Markov Models

CS 534: Machine Learning

Slides adapted from David Sontag, Geoffrey Hinton, Eric Xing, and Nicholas Ruozzi

Review: Applied ML Process



Review: ML Models

	Unsupervised	Supervised

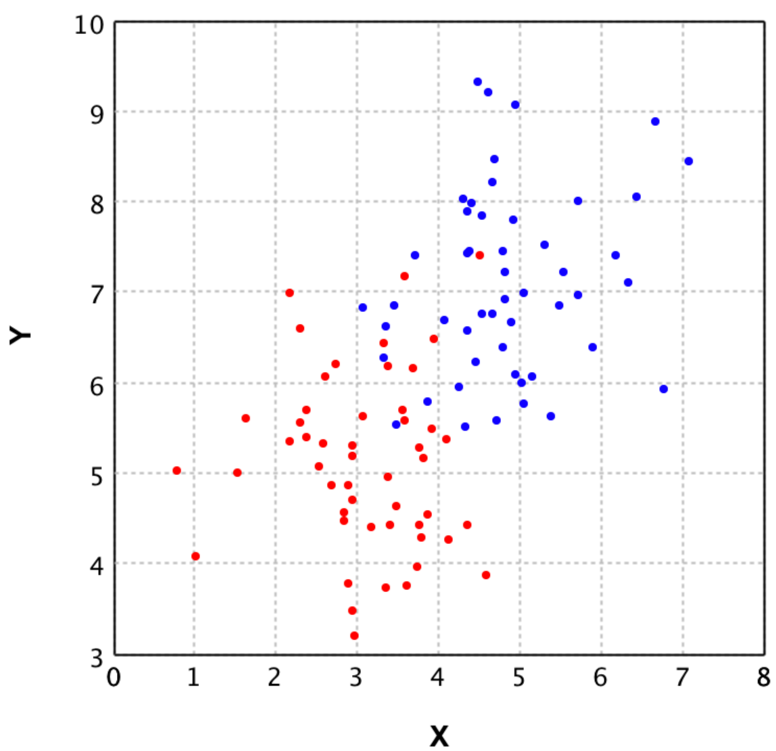
Continuous Data	• Clustering
	  • k-Means
	  • Agglomerative clustering
	  • Gaussian mixture models
	• Dimensionality reduction
	  • PCA / SVD
	  • NMF
		• Multivariate regression
		• Decision trees
		• SVR
		• Boosting, bagging, ensembles

Categorical Data		• k-NN • Decision trees • Logistic regression • SVM • Neural networks • Boosting, bagging, ensembles

Review: ML Models

~~Each point is independent and identically distributed~~

What if this isn't true?



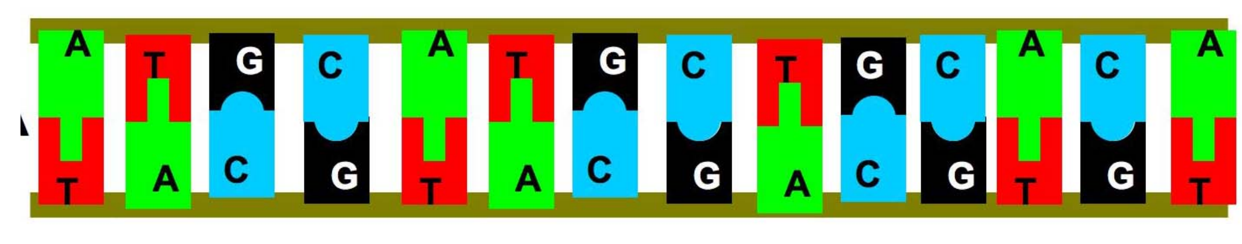
Favorite ML Algorithms

Classifier / output

Construct feature matrix **X**

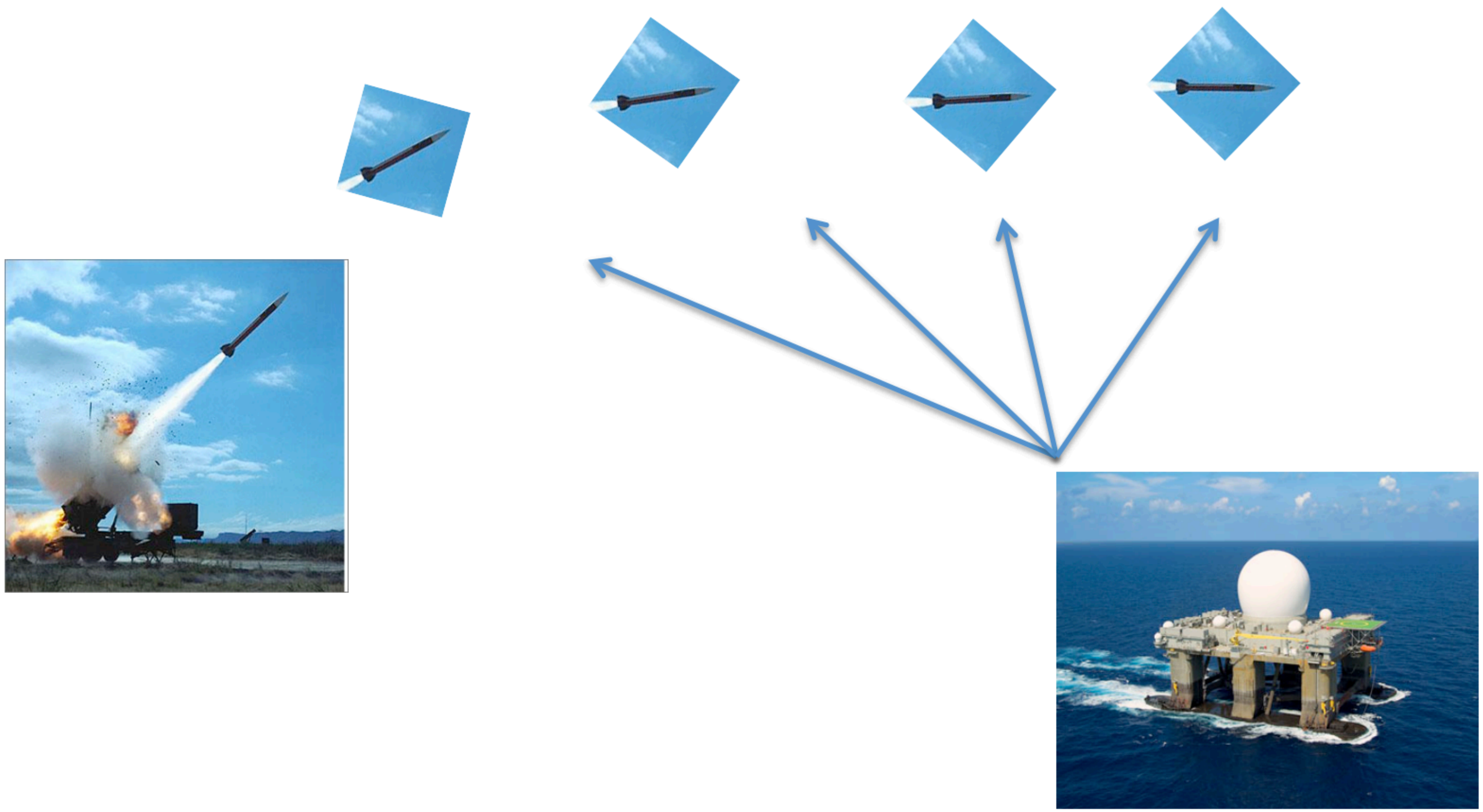
Sequential Data

- What about sequential data?
  - Time-series: Stock market, weather, speech, video
  - Ordered: Text, genes



Sequential Data: Tracking

Observe noisy measurements of missile location



Where is the missile now? Where will it be in 1 minute?

Sequential Data: Weather

- Predict the weather tomorrow using previous information
- If it rained yesterday, and the previous day and historically it has rained 7 times in the past 10 years on this date — does this affect my prediction?

The Weather Channel

Atlanta, GA 10 Day Weather

1:09 pm EDT

DAY	DESCRIPTION	HIGH / LOW	PRECIP	WIND	HUMIDITY
TODAY APR 5	Strong Storms	75°/52°	100%	SSW 17 mph	74%
THU APR 6	Cloudy/Wind	59°/42°	10%	W 25 mph	52%
FRI APR 7	Sunny/Wind	63°/41°	0%	WNW 23 mph	41%
SAT APR 8	Sunny	69°/43°	0%	NW 12 mph	38%
SUN APR 9	Sunny	76°/50°	0%	SSE 6 mph	42%
MON APR 10	Sunny	79°/54°	10%	SSE 9 mph	45%
TUE APR 11	Mostly Sunny	80°/58°	10%	SSE 8 mph	51%
WED APR 12	Partly Cloudy	77°/53°	20%	W 9 mph	55%
THU APR 13	Sunny	74°/52°	20%	NW 9 mph	50%
FRI APR 14	Sunny	72°/50°	10%	NW 14 mph	49%
SAT APR 15	Sunny	76°/52°	20%	WNW 8 mph	52%
SUN APR 16	Mostly Sunny	81°/57°	10%	WNW 9 mph	50%
MON APR 17	Mostly Sunny	82°/57°	20%	W 8 mph	52%
TUE APR 18	Isolated Thunderstorms	82°/57°	30%	WSW 8 mph	52%
WED APR 19	Partly Cloudy	82°/58°	20%	SSW 9 mph	53%

Sequential Data: Weather

- Use product rule for joint distribution of a sequence

$$P(X_1, X_2, \cdots, X_T) = \prod_{t=1}^{T} P(X_t | X_{t-1}, \cdots X_1)$$

- How do I solve this?
  - Model how weather changes over time
  - Model how observations are produced
  - Reason about the model

Markov Chain

- Markov chain: Sequence of random variables $X_1, X_2, \cdots, X_T \in S$ such that

$$p(x_{t+1}|x_1, \cdots, x_t) = p(x_{t+1}|x_t)$$

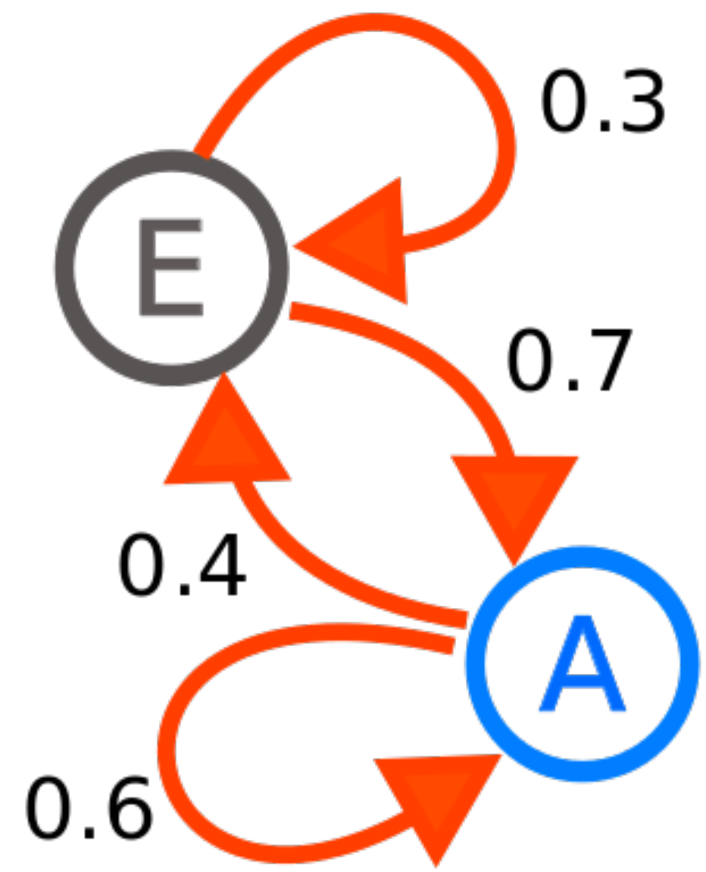
- Set S is called the state space

- Transition probability (probability of transitioning from state i to state j at time t)

$$p(X_{t+1} = j|X_t = i)$$

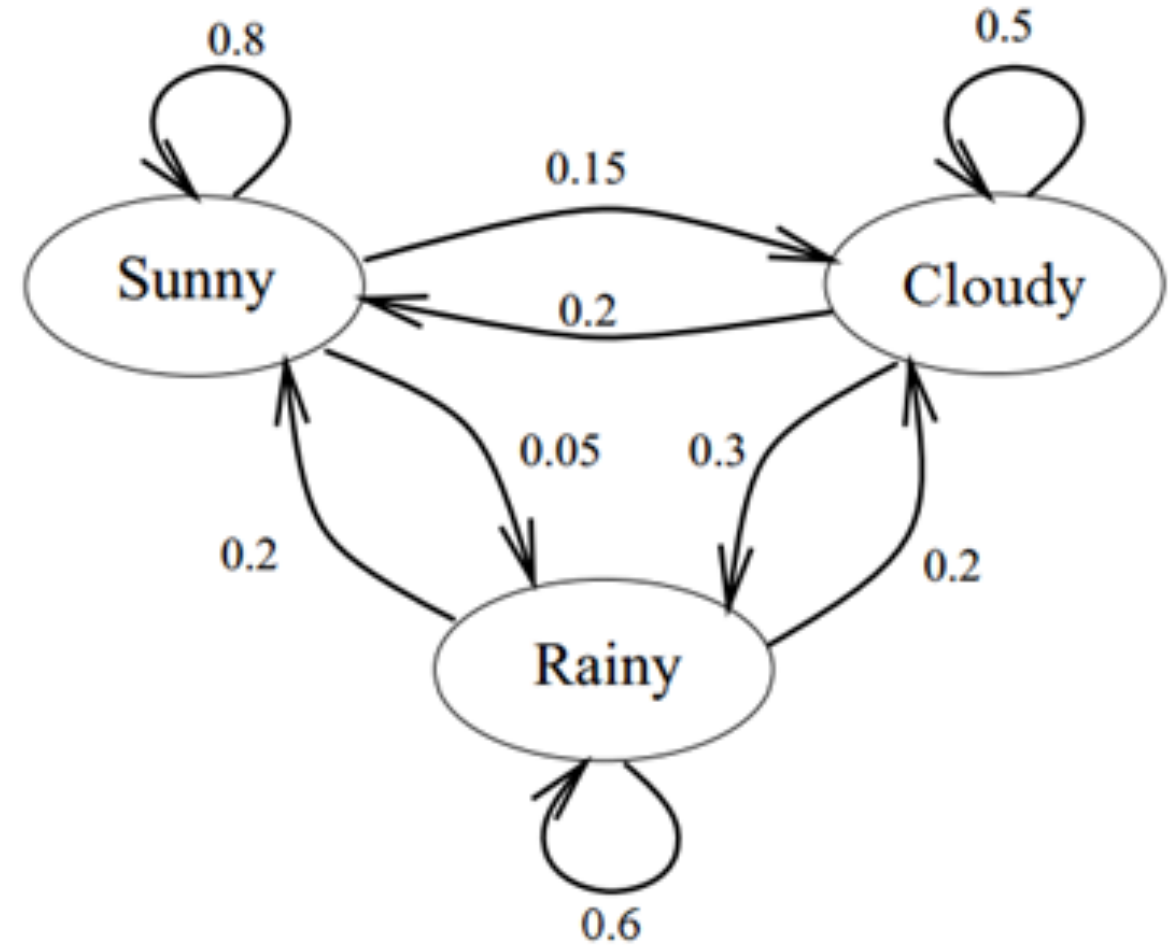
Markov Chain

- Time homogenous Markov chain: transition probability between two states does not depend on time
- Transition matrix A (|S| x |S|)
  - A is a stochastic matrix (all rows sum to one)
  - $A_{ij} = p(X_{t+1} = j | X_t = i)$



Example: Weather Prediction

- Markov chain to predict weather of tomorrow using previous information of the past days
- 3 states: Sunny, cloudy, rainy
- Establish transition probabilities by collecting data



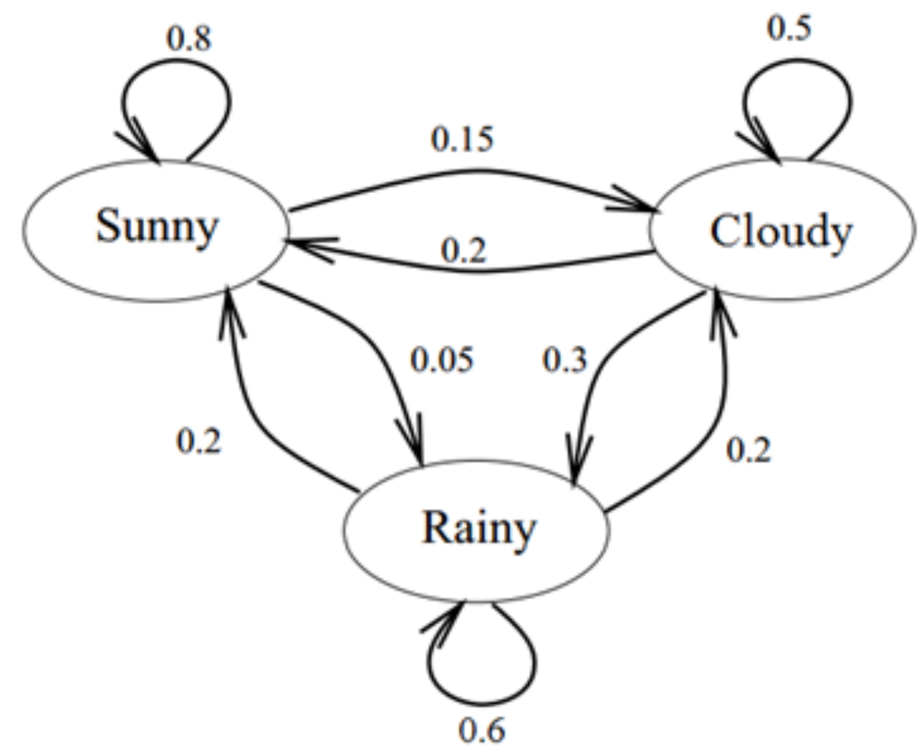
Example: Weather Prediction

- Compute probability of tomorrow’s weather using Markov property

$$p(X_1, \cdots, X_n) = \prod_{i=1}^{n} p(X_i | X_{i-1})$$

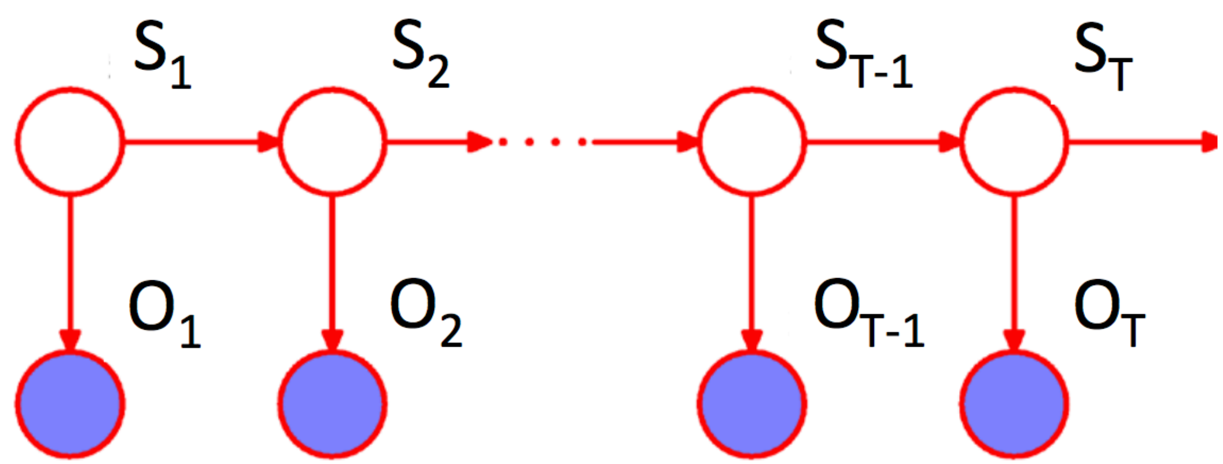
- Given today is sunny, what’s the probability that tomorrow is sunny and the next day is rainy?

- Given yesterday’s weather was rainy, and today is cloudy, what is the probability tomorrow will be sunny?



Hidden Markov Model (HMM)

- Stochastic model where the states of the model are hidden
- Each state can emit an output which is observed



HMM: Parameters

- State transition matrix **A**

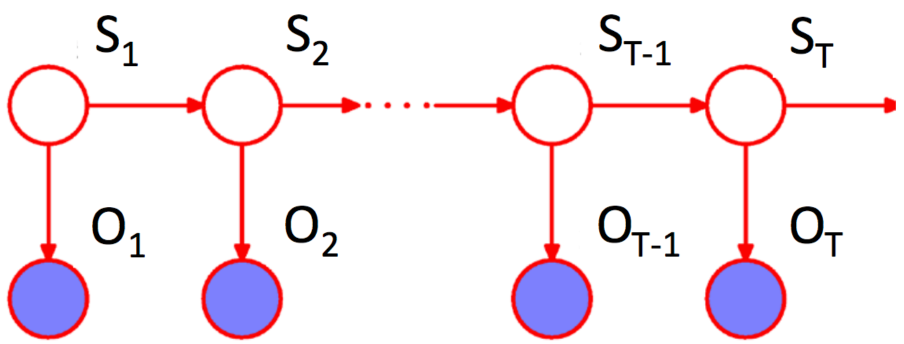
$$A_{jk} = p(S_t = s_k | S_{t-1} = s_j)$$

- Emission / observation conditional output probabilities **B**

$$B_{ik} = p(O_t = v_k | S_t = s_i)$$

- Initial (prior) state probabilities

$$\pi_i = p(S_i)$$



HMM: Properties

- Useful for modeling sequential data with few parameters using discrete hidden states that satisfy Markov assumption
- State space representation: initial probability, transition probability, emission probability
- Can be learned fairly efficiently

Example: Dishonest Casino

- A casino has two dices that it switches between with 5% probability
  - Fair dice

$$P(1) = P(2) = P(3) = 1/6$$
$$P(4) = P(5) = P(6) = 1/6$$

  - Loaded dice

$$P(1) = P(2) = P(3) = 1/10$$
$$P(4) = P(5) = 1/10$$
$$P(6) = 1/2$$

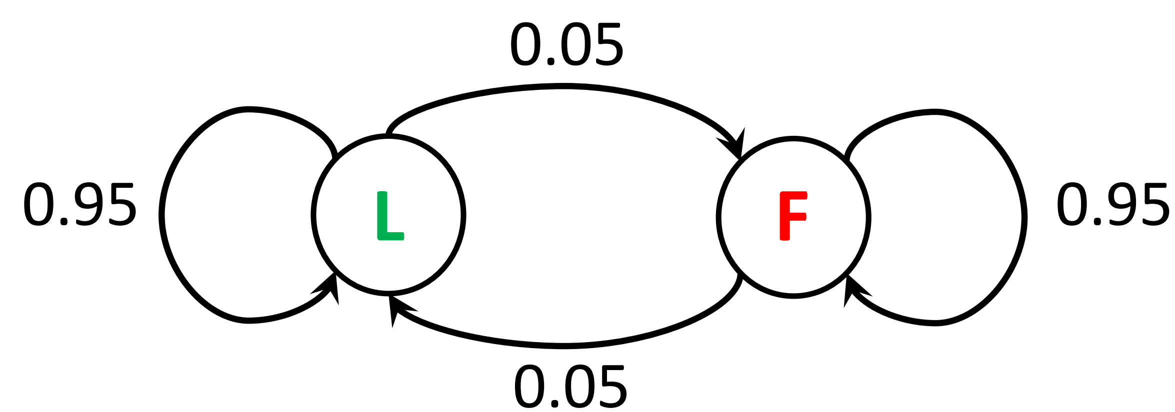
Example: Dishonest Casino

- Initial probabilities

$$P(S_1 = L) = P(S_1 = F) = 0.5$$

- State transition matrix

$$\mathbf{A} = \begin{bmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{bmatrix}$$



- Emission probabilities

$$\mathbf{B} = \begin{bmatrix} 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/10 & 1/10 & 1/10 & 1/10 & 1/10 & 1/2 \end{bmatrix}$$

Example: Dishonest Casino

- Given a sequence of rolls by the casino player

  1245526462146146136136661664661636616366163616515615115146123562344

- How likely is this sequence given our model of how the casino works?

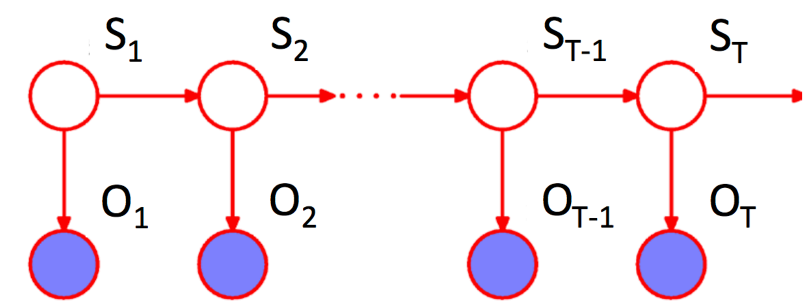
- What sequence portion was generated with the fair die?

- How “loaded” is the loaded die? How “fair” is the fair die? How often does the casino player change from fair to loaded and back?

HMM: Problems

- Evaluation: Given parameters and observation sequence, find probability of observed sequence
- Decoding: Given HMM parameters and observation sequence, find the most probable sequence of hidden states
- Learning: Given HMM with unknown parameters and observation sequence, find the parameters that maximizes likelihood of data

HMM: Evaluation Problem



- Given

$$p(S_1), p(S_t|S_{t-1}), p(O_t|S_t), \{O_t\}_{t=1}^T$$

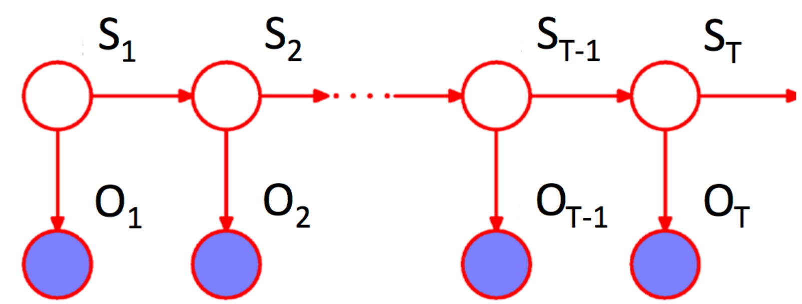
- Probability of observed sequence

$$p(\{O_t\}_{t=1}^T) = \sum_{S_1,\cdots,S_T} p(\{O_t\}_{t=1}^T, \{S_t\}_{t=1}^T)$$

$$= \sum_{S_1,\cdots,S_T} p(S_1) \prod_{t=2}^T p(S_t|S_{t-1}) \prod_{t=1}^T p(O_t|S_t)$$

Summing over all possible hidden state values at all times — K^T exponential # terms

HMM: Forward Algorithm



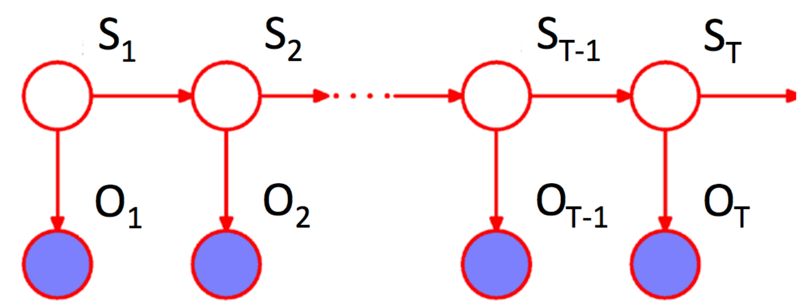
- Instead pose as recursive problem

$$p(\{O_t\}_{t=1}^{T}) = \sum_k \underbrace{p(\{O_t\}_{t=1}^{T}, S_T = k)}_{\alpha_T^k}$$

- Use dynamic programming to compute forward probability

$$\begin{aligned} \alpha_k^t = & p(O_1, \cdots, O_t, S_t = k) \\ & = p(O_t | S_t = k) \sum_i \alpha_{t-1}^i p(S_t = k | S_t - 1 = i) \end{aligned}$$

HMM: Decoding Problem 1



- Given

$$p(S_1), p(S_t|S_{t-1}), p(O_t|S_t), \{O_t\}_{t=1}^T$$

- Probability that hidden state at time t was k

$$\begin{aligned} p(S_t = k, \{O_t\}_{t=1}^T) &= p(O_1, \cdots, O_t, S_t = k, O_{t+1}, \cdots, O_T) \\ &= \underbrace{p(O_1, \cdots, O_t, S_t = k)}_{\alpha_t^k} \underbrace{p(O_{t+1}, \cdots, O_T | S_t = k)}_{\beta_t^k} \\ &= \alpha_t^k \beta_t^k \end{aligned}$$

We know how to compute the first part using forward algorithm

HMM: Backward Probability

- Similar to forward probability, we can express as a recursion problem

$$\begin{aligned}\beta_k^t =& p(O_{t+1}, \cdots, O_T | S_t = k) \\ =& \sum_i p(S_{t+1} = i | S_t = k) p(O_{t+1} | S_{t+1} = i) \beta_{t+1}^i\end{aligned}$$

- Dynamic program
  - Initialize $\beta_T^k = 1$
  - Iterate using recursion

HMM: Decoding Problem 1

- Probability that hidden state at time t was k

$$P(S_t = k | \{O_t\}_{t=1}^T) = \frac{p(S_t = k, \{O_t\}_{t=1}^T)}{p(\{O_t\}_{t=1}^T)}$$

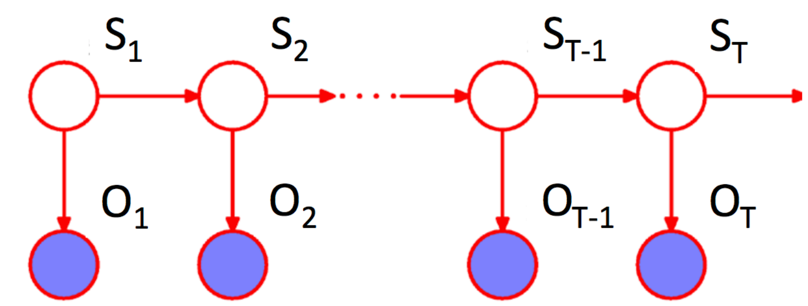
$$= \frac{\alpha_t^k \beta_t^k}{\sum_i \alpha_t^i \beta_t^i}$$

Forward-backward algorithm

- Most likely state assignment

$$\text{argmax}_k p(S_t = k | \{O_t\}_{t=1}^T) = \text{argmax}_k \alpha_t^k \beta_t^k$$

HMM: Decoding Problem 2



- Given
  $p(S_1), p(S_t|S_{t-1}), p(O_t|S_t), \{O_t\}_{t=1}^{T}$

- What is most likely state sequence?

$$\begin{aligned}\mathrm{argmax}_k p(\{S_t\}_{t=1}^{T}|\{O_t\}_{t=1}^{T}) \\ &= \mathrm{argmax}_k p(\{S_t\}_{t=1}^{T}, \{O_t\}_{t=1}^{T}) \\ &= \mathrm{argmax}_k \underbrace{\max_{\{S_t\}_{t=1}^{T-1}} p(S_T = k, \{S_t\}_{t=1}^{T-1}, \{O_t\}_{t=1}^{T})}_{v_T^k}\end{aligned}$$

probability of most likely sequence of states ending at state S_T=k

HMM: Viterbi Algorithm

- Compute probability recursively over t

$$
\begin{aligned}
v_t^k &= \max_{\{S_t\}_{t=1}^t} p(S_t = k, \{S_t\}_{t=1}^{t-1}, \{O_t\}_{t=1}^t) \\
&= p(O_t | S_t = k) \max_i p(S_t = k | S_{t-1} = i) v_{t-1}^i
\end{aligned}
$$

- Use dynamic programming again!

HMM: Viterbi Algorithm

- Initialize

$$v_1^k = p(O_1|S_1 = k)p(S_1 = k)$$

- Iterate

$$v_t^k = p(O_t|S_t = k)\max_i p(S_t = k|S_{t-1} = i)v_{t-1}^i$$

- Terminate

$$\max_{\{S_t\}_{t=1}^{T-1}} p(\{S_t\}_{t=1}^T, \{O_t\}_{t=1}^T) = \max_k v_T^k$$

$$S_T^* = \text{argmax}_k v_T^k$$

Traceback

$$S_{t-1}^* = \text{argmax}_i p(S_t^*|S_{t-1} = i)v_{t-1}^i$$

HMM: Computational Complexity

- What is the running time for the forward algorithm, backward algorithm, and Viterbi?

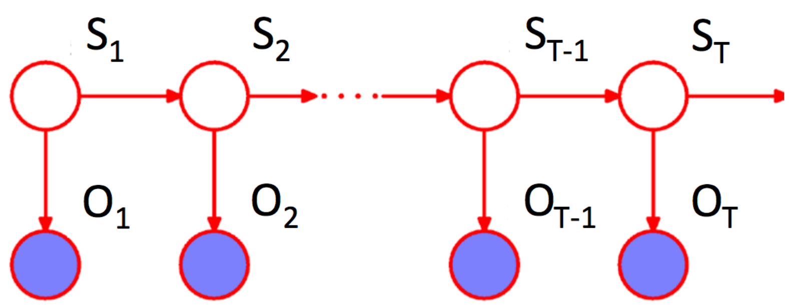
$$\alpha_k^t = p(O_t | S_t = k) \sum_i \alpha_{t-1}^i p(S_t = k | S_t - 1 = i)$$

$$\beta_k^t = \sum_i p(S_{t+1} = i | S_t = k) p(O_{t+1} | S_{t+1} = i) \beta_{t+1}^i$$

$$v_t^k = p(O_t | S_t = k) \max_i p(S_t = k | S_{t-1} = i) v_{t-1}^i$$

$O(K^2T)$ vs $O(K^T)$!

HMM: Learning Problem



- Given only observations

$$\{O_t\}_{t=1}^{T}$$

- Find parameters that maximize likelihood

$$\mathrm{argmax}_{\theta} p(\{O_t\}_{t=1}^{T}|\theta)$$

- Need to learn hidden state sequences as well

- Much harder problem than the others — use our friend, the EM algorithm

HMM: Baum-Welch (EM) Algorithm

- Randomly initialize parameters
- E-step: Fix parameters, find expected state assignment

$$\gamma_i(t) = p(S_t = i | \{O_t\}_{t=1}^T, \theta) = \frac{\alpha_t^k \beta_t^k}{\sum_i \alpha_t^i \beta_t^i}$$

Forward-backward algorithm

$$\begin{aligned}
\epsilon_{ij}(t) &= p(S_{t-1} = i, S_t = j | \{O_t\}_{t=1}^T, \theta) \\
&= \frac{p(S_{t-1} = i | \{O_t\}_{t=1}^T, \theta) p(S_t = j, O_t, \cdots, O_T | S_{t-1} = i, \theta)}{p(O_t, \cdots, O_T | S_{t-1} = i, \theta)} \\
&= \frac{\gamma_i(t-1) p(S_t = j | S_{t-1} = i) P(O_t | S_t = j) \beta_t^j}{\beta_{t-1}^i}
\end{aligned}$$

HMM: Baum-Welch (EM) Algorithm

- Expected number of times we will be in state i

$$\sum_{t=1}^{T} \gamma_i(t)$$

- Expected number of transitions from state i

$$\sum_{t=1}^{T-1} \gamma_i(t)$$

- Expected number of transitions from state i to j

$$\sum_{t=1}^{T-1} \epsilon_{ij}(t)$$

HMM: Baum-Welch (EM) Algorithm

- M-step: Fix expected state assignments, update parameters

$$\pi_i = \gamma_i(1)$$

$$A_{ij} = \frac{\sum_{t=1}^{T-1} \epsilon_{ij}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)}$$

$$B_{ik} = \frac{\sum_{t=1}^{T} \gamma_i(t)\delta_{O_t=k}}{\sum_{t=1}^{T} \gamma_i(t)}$$

HMM: Applications

- Classification
  - DNA sequences
  - Gesture sequences
  - Video sequences
  - Phoneme sequences
  - Etc.
- Decoding
  - Continuous speech recognition
  - Handwriting recognition
  - Sequence of events

HMM vs Linear Dynamical Systems

- HMM
  - States are discrete
  - Observations are discrete or continuous
- Linear dynamical systems
  - Observations and states are multivariate Gaussians
  - Can use Kalman Filters to solve