

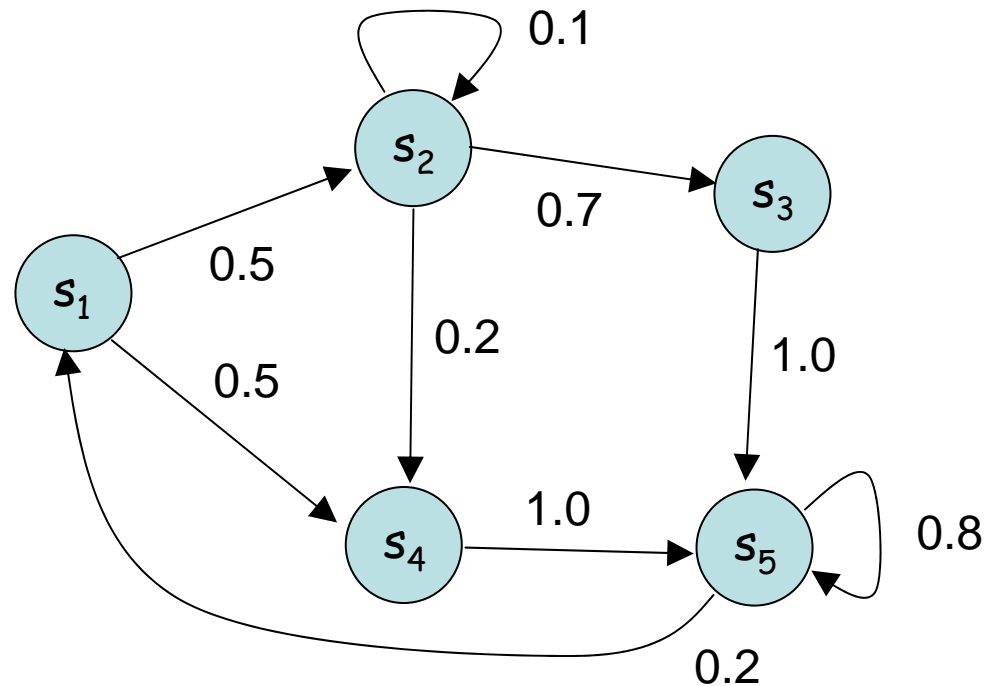
Sequential Prediction & Graphical Models

Introduction to Statistical Machine Learning
ANU COMP 6467/4670, Sem 1, 2008

Scott Sanner
NICTA / RSISE
First.Last@nicta.com.au

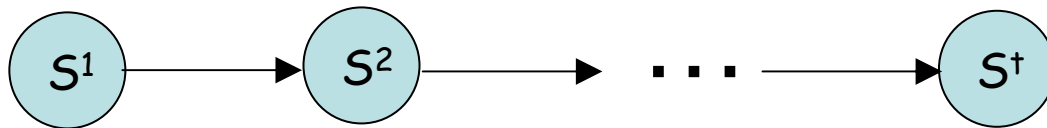
Markov Models (or Markov Chains)

- At each time step, probabilistically transition from current state to next state ($S = \{s_1, s_2, \dots, s_n\}$)
- Finite State Machine (FSM) view for $n=5$:



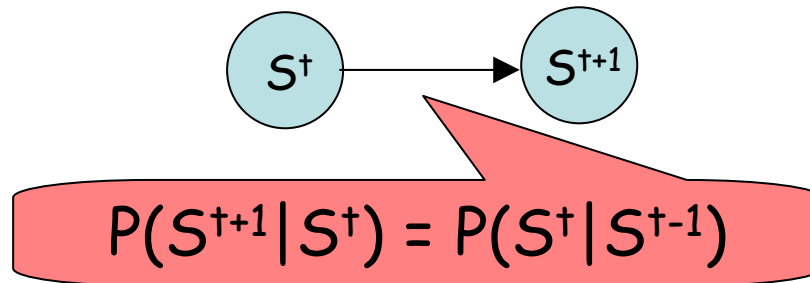
Markov Models

- The graphical model view for t steps:



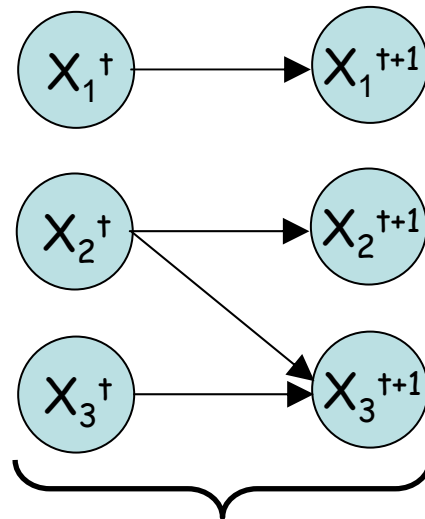
– Note: for $t = \infty$, an infinite graphical model!

- Or assuming transition stationarity, just:



Markov Models

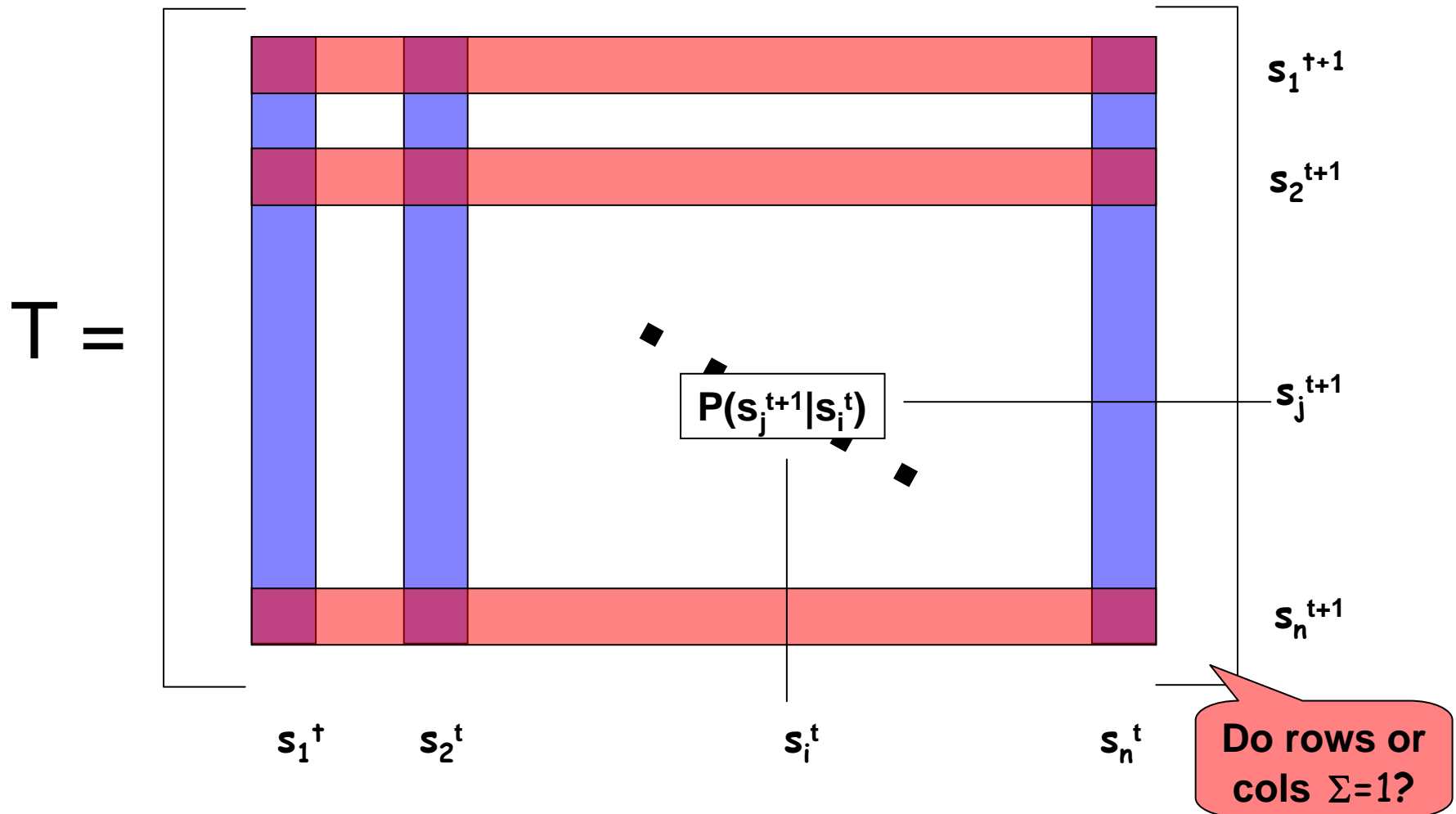
- The Dynamic Bayes Net (DBN) view:
 - State factors into variables: X_1, X_2, \dots, X_k
 - Capture transition independences



$$P(x_1^{t+1}, x_2^{t+1}, \dots, x_k^{t+1} \mid x_1^t, x_2^t, \dots, x_k^t) = \Pi \dots$$

Transition Matrix

- Represent $P(s^{t+1}|s^t)$ as transition matrix:



Transition Probabilities

- Formally
 - Define state set $S^t = \{s_1, s_2, \dots, s_n\} ; \forall t$
 - Define transition matrix $T_{ij}^t = P(S_i^{t+1} | S_j^t) ; \forall t$
- Properties of T_{ij}
 - *Stationary*: $T_{ij}^t = T_{ij}^{t-1}$ OR $P(S^{t+1} | S^t) = P(S^t | S^{t-1}) ; \forall t$
 - *Irreducible*: Possible to get from any s_i to s_j
 - *Aperiodic*: Time to return has periodicity = 1
 - *Transient*: Positive probability of not returning to state
 - *Recurrent*: Not transient
 - *Ergodic*: Aperiodic and (positive) recurrent

Examples
of each?

Distribution at Time t

- Given $P(s^0)$, what is $P(s^t)$?
- Use var. elim. to marginalize over intermediate time steps

- $P(s^t) = \sum_{s_1, \dots, s_{t-1}} P(s^0) \prod_{i=0 \dots t-1} P(s^{i+1} | s^i)$

If no evidence after time t, all factors for t+1 and after marginalize out

- Or let P_s^0 & P_s^t be column vectors...
 - Then simply: $P_s^t = (T^t) P_s^0$
 - Note: Intimate connection between matrix ops and var. elim.
 - When $P(s^{i+1} | s^i)$ factors as a DBN...
capture many efficiencies of var. elim. via sparse matrix ops

Stationary Distribution

- Stationary Distribution π at $t=\infty$
 - $\pi = (T^{\infty}) P s^0$
 - If T ergodic & irreducible, $P s^0$ irrelevant
 - Reaches *unique* steady-state distribution: $\pi = T \pi$
 - So $\pi =$ any row of T^{∞}
 - Can solve via eigenvector analysis (note: $\lambda=1$)
 - Related to (Krylov) iterated eigenvector computation
 - Or use fixed point to solve linear system
 - $T \pi - \pi = 0 \rightarrow \pi' T' - \pi' = 0 \rightarrow \pi' (T' - I) = 0$
s.t. constraints on π
 - Can solve linear system via matrix inversion

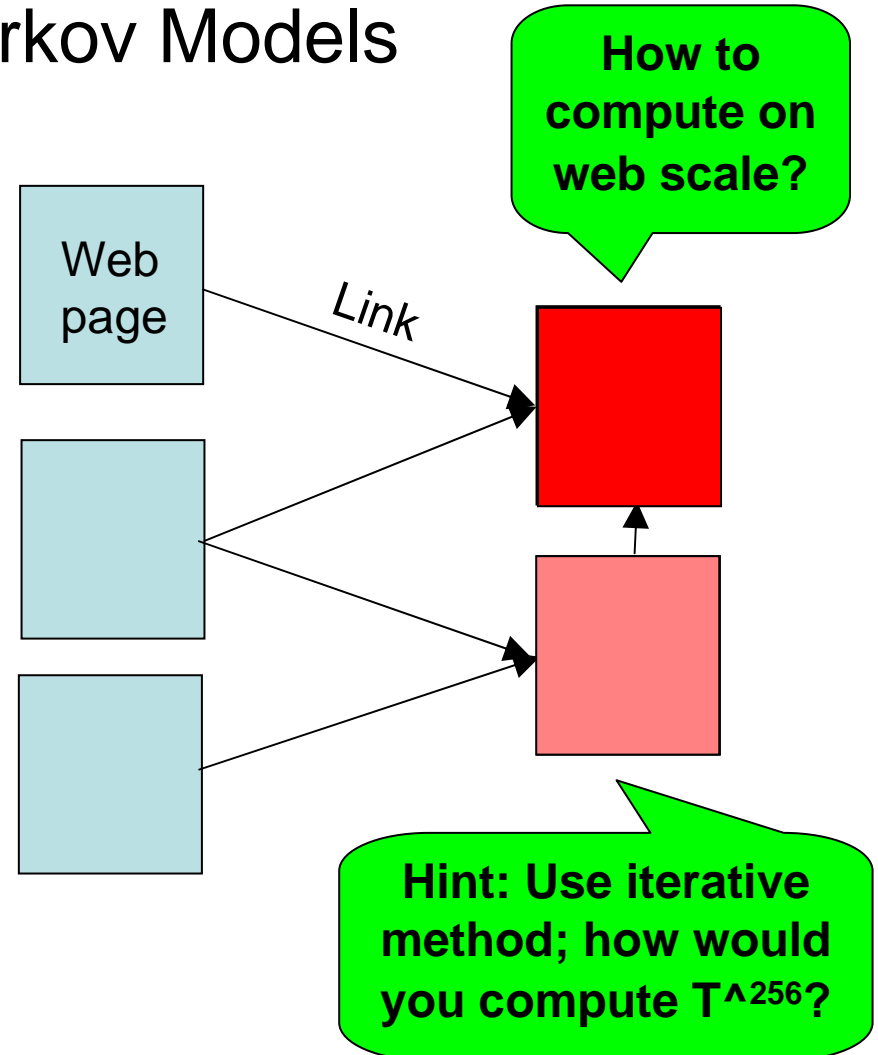
Why? What are they?

Markov Model Applications

- Simple theory, ingenious applications:
 - n^{th} -order Markov models
 - Relax Markovian assumption to previous n states
 - Used in text and speech processing
 - N-grams for predicting next word occurrence
 - Colocation identification
 - [Dasher](#) for text input, try it in your [web browser](#)
 - More generally
 - Physics (states of systems)
 - Queuing theory (random entries and exits)
 - Economics, Biology, Chemistry, etc...
 - Google!

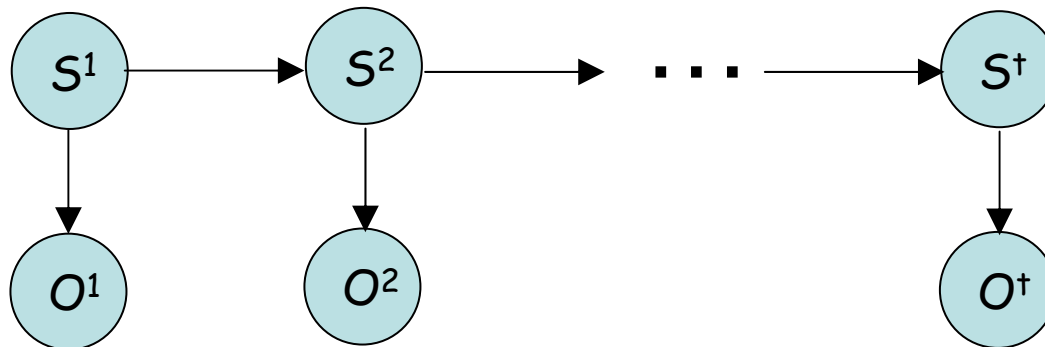
Google PageRank Example

- Very beautiful use of Markov Models
- Model of web browsing:
 - Probabilistically take link with $\sim 1/k$ chance if k links
 - Small chance of random transition
- Stationary distribution π gives PageRank!
 - Measure of “authority”



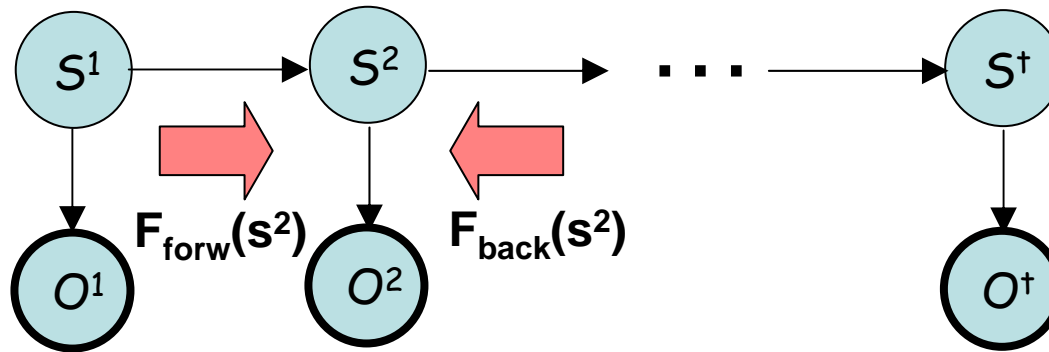
Hidden Markov Models

- Formally
 - Define state set $S^t = \{s_1, s_2, \dots, s_n\} ; \forall t$
 - Define observation set $O^t = \{o_1, o_2, \dots, o_m\} ; \forall t$
 - Define transition matrix $T_{ij}^t = P(S_i^{t+1} | S_j^t) ; \forall t$
- Graphical Model view:



Forward-Backward Algorithm

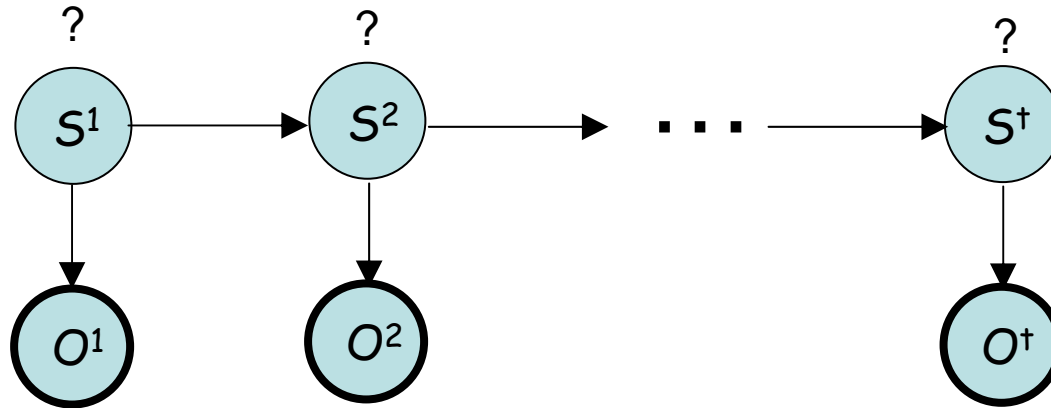
- Compute marginals at all S^t given observations
 - Can perform var. elim. (VE) \dagger times for each marginal
 - But many repeated computations



- Forward-backward algorithm:
 - Just cache forward and backward VE messages to each S^t
 - Marginal is then just appropriate product

Viterbi Algorithm

- Most probable state assignment given observations
 - $\operatorname{argmax}_{s^0, \dots, s^t} P(s^0) P(o^0 | s^0) \prod_{i=0 \dots t-1} P(s^{i+1} | s^i) P(o^{i+1} | s^{i+1})$



- Define (i.e., preallocate arrays and compute):
 - $v_j(0) = P(o^0 | S^0=j) P(S^0=j)$
 - $a_j(t) = \operatorname{argmax}_i P(o^t | S^t=j) P(S^t=j | S^{t-1}=i) v_j(t-1) ; t \geq 1$
 - $v_j(t) = \max_i P(o^t | S^t=j) P(S^t=j | S^{t-1}=i) v_j(t-1) ; t \geq 1$
- Extract maximal state assignment:
 - Best (most probable) state at time t is $b^t = \operatorname{argmax}_b v_b(t)$
 - Work backward using $b^{t-1} = a_{b^t}(t)$ until $t=1$

Assignment at $t-1$ that leads to best $v_j(t)$

Learning ML Parameters

- Also applies to just Markov Model
- Note: We have an infinite graphical model!
 - With non-stationary model, never have enough data!
 - With stationary model, much better...
 - Key trick is implicit parameter tying
 - Lots of data per parameter = low variance
 - Can always convert non-stationary to stationary by encoding extra state information!
- As we know in general for Bayes nets
 - To estimate $P(s^{t+1}|s^t)$ & $P(o^t|s^t)$
 - Just empirical estimates from frequency counts

Baum-Welch Training

(using Forward-backward Algorithm)

- Learning ML Parameters w/ data that has unobserved state
- Maximize expected joint log likelihood of observed & unobserved variables in data
- Just EM, iterate:
 - *E-step*: Use forward-backward to compute expectations for all s^t
 - *M-step*: Use generalized EM update to compute transition probabilities that increase likelihood
 - Just use the expected transition function
 - Improves expected log likelihood, or reaches fixed-point maxima... historically, this took some effort to prove

Generative vs. Discriminative HMMs

- Maximum Likelihood (ML)
 - $L(\theta) = \operatorname{argmax}_{\theta} \prod_{d \in \mathcal{D}} P(s^d, o^d | \theta)$
 - This trains a generative model
 - $L(\theta)$ convex in θ , at maxima:
 - expected sufficient statistics = empirical sufficient statistics
 - closed-form training!
- Maximum Conditional Likelihood (MCL)
 - $CL(\theta) = \operatorname{argmax}_{\theta} \prod_{d \in \mathcal{D}} P(s^d | o^d, \theta)$
 - This trains a discriminative model
 - If we always observe o^d , goal should be to maximize prediction of s^d
 - $CL(\theta)$ still convex in θ , at maxima:
 - conditional expected sufficient statistics = empirical sufficient statistics
 - no closed-form training, but can use gradient descent
- MCL version of HMM is “Chain-CRF (Conditional Random Field)”
 - See excellent [CRF tutorial](#)

Other Training Methods

- Max margin (MM) structured estimation
 - Connection to multiclass SVM-style training
 - Maximize some margin between true label and incorrect classification
- (Conditional) pseudolikelihood
 - An efficient *approximation* of (conditional) likelihood

When to use Each Training Method?

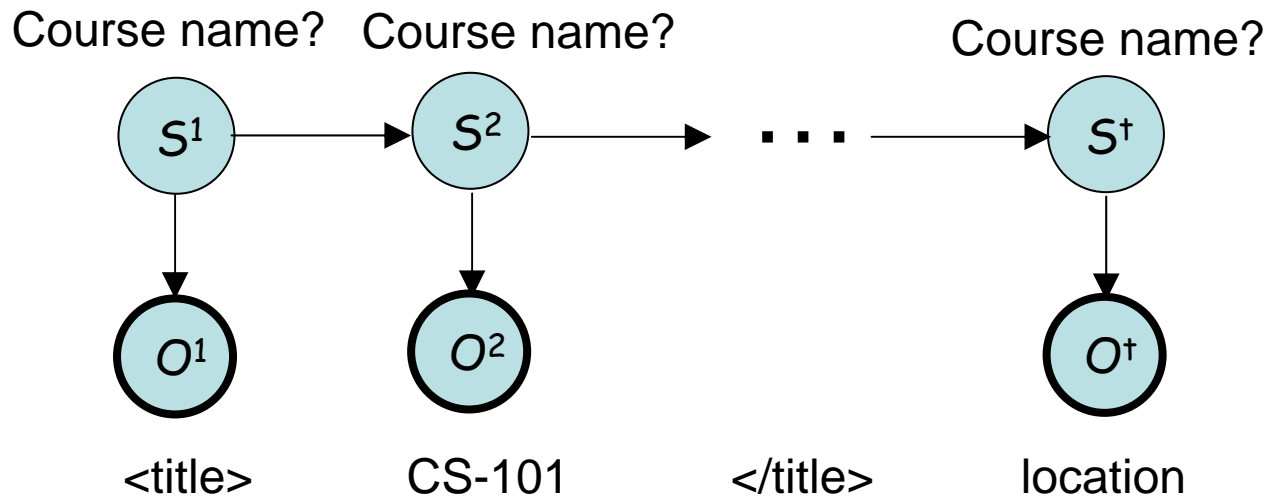
- If have unobserved variables in training data, must use EM
 - But can use most training methods in conjunction with EM
- So which general training method to use (ML, MCL, MM)?
 - MCL and MM usually better than ML, but ML fastest of all three
 - MCL and MM usually perform comparably, some claim MM better
 - MCL training requires expectations, MM training requires MAP inference
- MCL and MM do better when
 - Feature model is impoverished
 - Features are not independent
 - Low data
- If have choice of more features vs. expensive training
 - Try adding in better features first

Machine Learning's Dirty Little Secret:

(Efficient) naïve Bayes with good features outperforms
(Slow) logistic regression or SVMs with bad features!

Example: Information Extraction

- Information Extraction



- Train from labeled data with ML (fully observed)
 - Note: important to “smooth” transition parameters
- Extract information on new example with Viterbi

Kalman Filtering

- So far, we've implicitly assumed discrete state and observation variables
- What about continuous distributions?
- Closed form marginal updates (filtering) if assume a Gaussian initial distribution and transition given by Gaussian
- See [Kalman filtering article](#) for more info

Neither Discrete nor Gaussian?

- All derivations still apply...
 - We just used general properties of probability distributions in graphical model
- But updates will now have complex form (that typically grows in size)
- Need to approximate. How?
 - Sampling
 - Message Compression
 - Variational Methods (approximation guarantees)
 - Expectation Propagation (effective in practice)

General HMM Applications

- Very general model
 - One of most used tools in machine learning
 - Developed independently in multiple fields
- Many uses, again art is in its application...
 - Speech recognition
 - Tracking based on sensor data
 - Objects on Radar
 - Cars from vision input
 - Simultaneous Localization and Mapping
 - Track position with Monte Carlo Localization
 - Learn latent map variables at same time with EM
 - Language
 - Part-of-speech tagging
 - Information extraction
 - E.g., extracting appointment details from email

Sequential Decision Theory

- We've looked at sequential prediction
 - *But* what if we can choose actions that affect model?
 - And differing utilities for different states?
- Fully observable case (MDP):
 - Markov Decision Process (MDP) = Markov Model + Actions
- Partially observable case (POMDP):
 - Partially Observable MDP = HMM + Actions
- Reinforcement Learning (RL):
 - The transition and/or reward model must be sampled from experience
- Multiple adversarial agents?
 - Partially Observable Stochastic Games (from game theory)
- Problem: How to act optimally in a (multi-agent) MDP or POMDP?
 - Marcus' lecture next week
 - Course next semester "Reinforcement Learning and Planning under Uncertainty"