Readings:
K&F: 18.1, 18.2, 18.3, 18.4

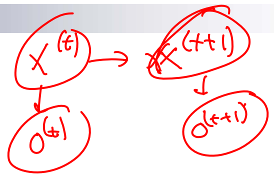
Dynamic Bayesian Networks

Beyond 10708

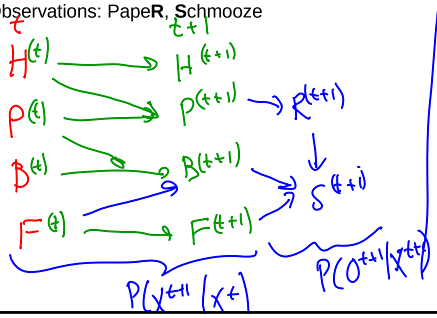
Graphical Models – 10708
Carlos Guestrin
Carnegie Mellon University

December 1st, 2006

Dynamic Bayesian network (DBN)



- HMM defined by
  - ☐ Transition model $P(X^{(t+1)}|X^{(t)})$
  - ☐ Observation model $P(O^{(t)}|X^{(t)})$
  - ☐ Starting state distribution $P(X^{(0)})$
- DBN – Use Bayes net to represent each of these compactly
  - ☐ Starting state distribution $P(X^{(0)})$ is a BN
  - ☐ (silly) e.g, performance in grad. school DBN
    - Vars: **H**appiness, **P**roductivity, Hira**B**lility, **F**ame
    - Observations: Pape**R**, **S**chmooze



binary

$P(X^{(t+1)} | X^{(t)})$

how many params $(2^4-1)2^4$

without DBN 2^8-2^4

with DBN

$P(H^{t+1} | H^t)$ $(2-1)\cdot 2$
$P(P^{t+1} | P^t, H^t)$ $(2-1)\cdot 2^2$
$P(B^{t+1} | P^t, B^t, F^t)$ $(2-1)\cdot 2^3$
$P(F^{t+1} | F^t)$ $(2-1)2$

16

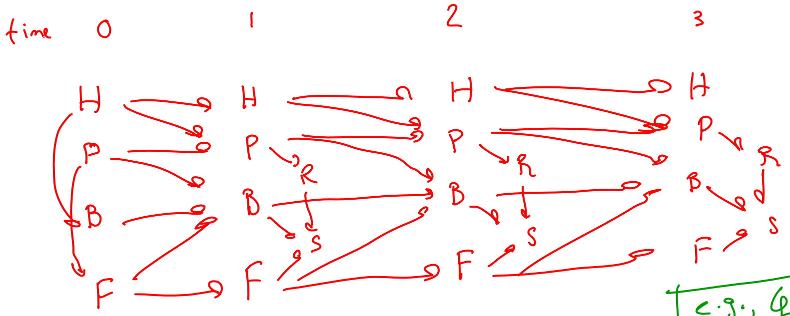
Unrolled DBN

$\overbrace{H,P,B,F}^{X}$ $\overbrace{R,S}^{O}$

$H \rightarrow B$

$P \rightarrow F$

- Start with $P(X^{(0)})$
- For each time step, add vars as defined by 2-TBN



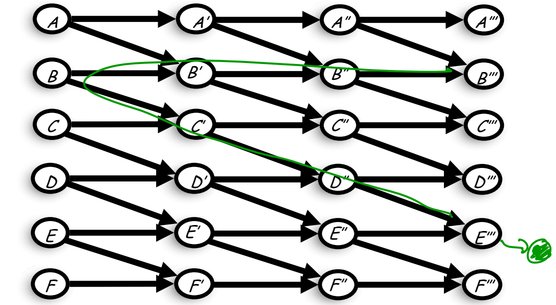
$P(X^{(0)}, X^{(1)}, O^{(1)}, X^{(2)}, O^{(2)}, X^{(3)}, O^{(3)})$

e.g., query:

$P(\underline{B^{(3)}=t} \mid S^{(1)}=t, R^{(1)}=t, R^{(2)}=t, S^{(3)}=f)$

"Sparse" DBN and fast inference

~~"Sparse" DBN~~ ☹ ~~Fast inference~~



$A''' \perp E'''$ true

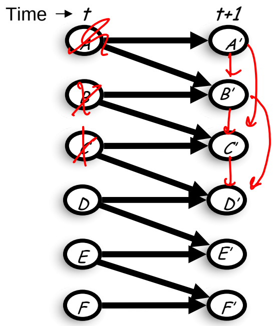
$B''' \perp E'''$ no!! ☹

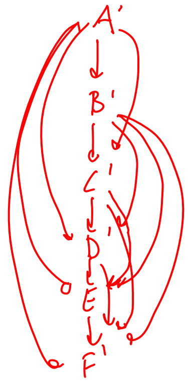
$A'''' \perp E'''$ no!! ☹

Even after one time step!!

What happens when we marginalize out time t?

$P(x^{(t)})$ $P(x^{(t+1)} | x^{(t)})$ $P(x^{(t+1)}) = \sum_{x^{(t)}} P(x^{(t)}) . P(x^{(t+1)}/x^{(t)})$





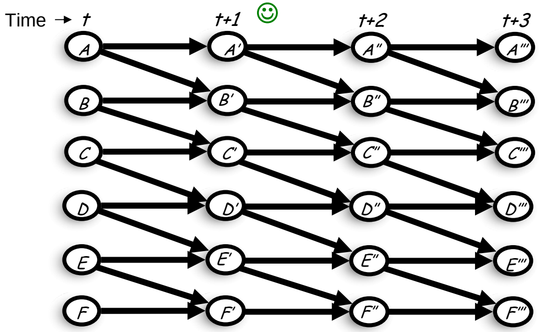
fully connected graph

“Sparse” DBN and fast inference 2

Structured representation of belief often yields good approximate

Almost!

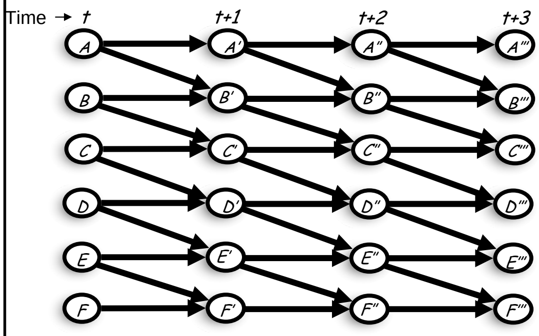
“Sparse” DBN $\xrightarrow{?}$ Fast inference ☺

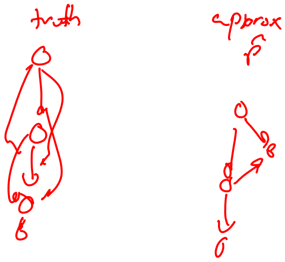


BK Algorithm for approximate DBN inference

[Boyen, Koller '98]

- Assumed density filtering:
  - Choose a factored representation $\hat{P}$ for the belief state
  - Every time step, belief not representable with $\hat{P}$, project into representation

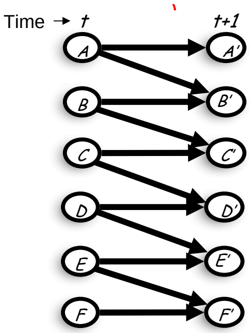




A simple example of BK: Fully-Factorized Distribution

- Assumed density:
  - Fully factorized

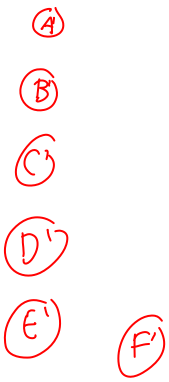
$$\hat{P}^{(t)}(X^{(t)}) = \prod_i \hat{P}_i(X_i^{(t)})$$



True P(X$^{(t+1)}$):

fully connected graph !! :)

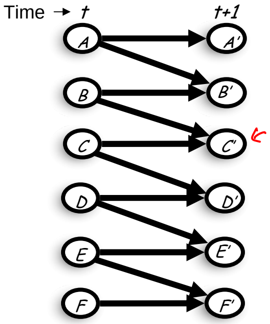
Assumed Density for $\hat{P}$(X$^{(t+1)}$):



Computing Fully-Factorized Distribution at time t+1

- Assumed density:
  - ☐ Fully factorized

Time → t   t+1



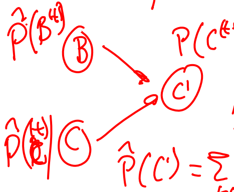
Assumed Density for $\hat{P}(X^{(t+1)})$:

$= \prod_i \hat{P}_i(X_i^{(t)})$

Computing for $\hat{P}(X_i^{(t+1)})$:

$= \prod_i \hat{P}_i(X_i^{(t+1)})$

how would you compute $\hat{P}(C')$



$P(C^{(t+1)} | B^{(t)}, C^{(t)})$

↑ trasition model

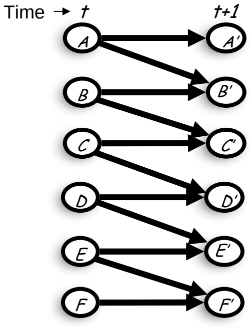
$\hat{P}(C') = \sum_{bc} \hat{P}(b) \cdot \hat{P}(c) \cdot P(C' | b, c)$

Variable elimination ingeneral

General case for BK: Junction Tree Represents Distribution

- Assumed density: ↙ too simple
  - ☐ Fully factorized
  - ☐ Junction trees

Time → t   t+1



True P(X^(t+1)):

A'B'C'
D'E'F'

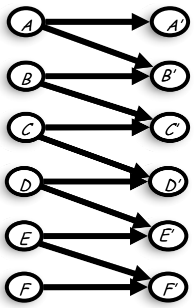
Assumed Density for $\hat{P}(X^{(t+1)})$:

choose JT
For example

A'B'
|
B'C'
|
C'D'
|
D'E'
|
E'F'

Computing factored belief state in the next time step

- Introduce observations in current time step
  - ☐ Use J-tree to calibrate time *t* beliefs
- Compute *t+1* belief, project into approximate belief state
  - ☐ marginalize into desired factors
  - ☐ corresponds to KL projection
- Equivalent to computing marginals over factors directly
  - ☐ For each factor in *t+1* step belief
    - Use variable elimination



KL-Projection

start $\hat{p}^{(t)}$

compute true $P^{(t+1)}$

$\arg\min KL(P^{(t+1)}, q)$

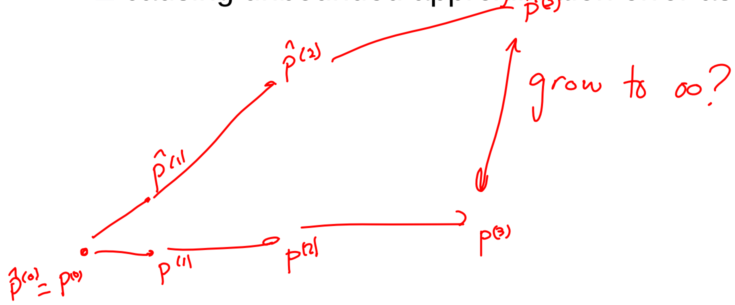
$q \in$ family of JT for $\hat{p}$ chosen

$= \hat{p}^{(t+1)}$

if $\hat{p}^{(t+1)}$ is a JT & $\hat{P}_i(C_i^{(t+1)})$ computed exactly with V.E.

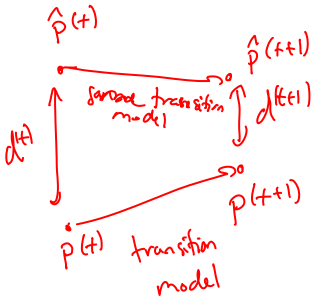
Error accumulation

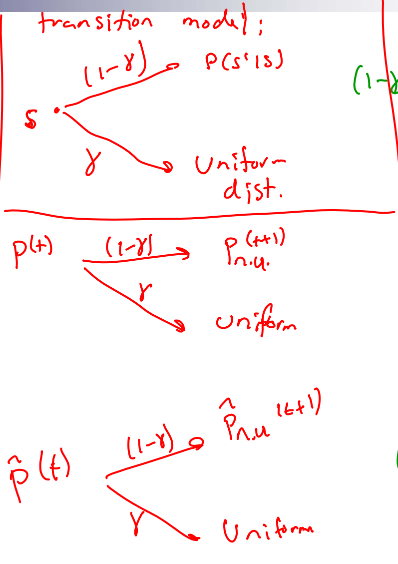
- Each time step, projection introduces error
- Will error add up?
  - ☐ causing unbounded approximation error as $t \rightarrow \infty$



Contraction in Markov process

$d^{(t)} > d^{(t+1)}$





$\Rightarrow$

$$(1-\gamma) KL\left(p^{(t)}, \hat{p}^{(t)}\right) \geq KL\left(p^{(t+1)}, \hat{p}^{(t+1)}\right)$$

side note: if make obs. o

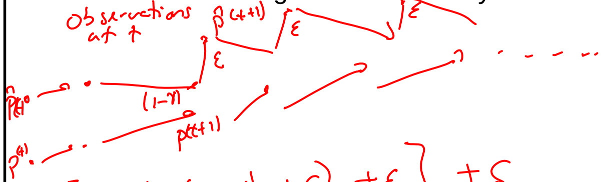
$$KL\left(p^{(t)}, \tilde{p}^{(t)}\right) \geq E_o\left[KL\left(p^{(t)}|o, \hat{p}^{(t)}|o\right)\right]$$

non-expansion in expectation

BK Theorem

$$\sum_{i=0}^{\infty} (1-\gamma)^i = \frac{1}{\gamma}$$

- Error does not grow unboundedly!

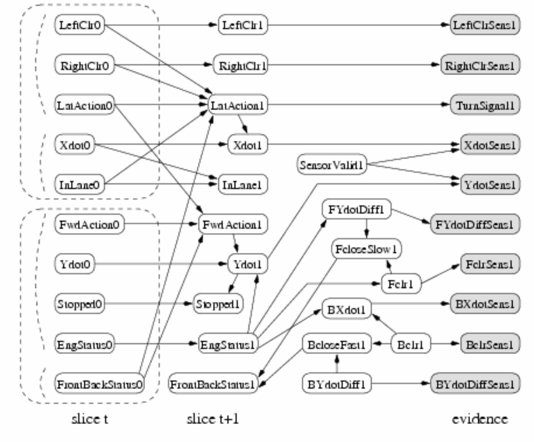


$$(1-\gamma)\left[(1-\gamma)\left(\varepsilon(1-\gamma)+\varepsilon\right)+\varepsilon\right]+\varepsilon \dots \leq \frac{\varepsilon}{\gamma}$$

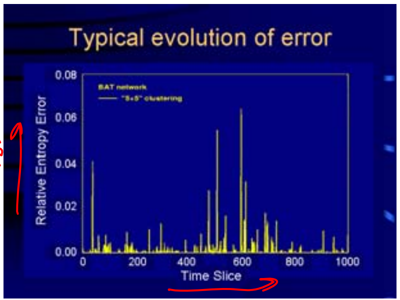
Error ε is error of approx. one step by ST. (or assumed density)

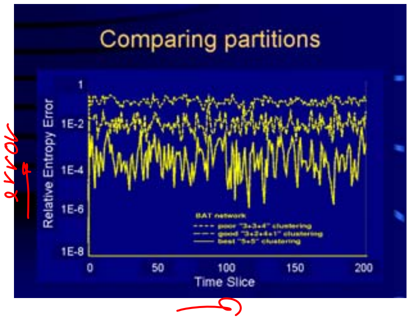
- **Theorem**: If Markov chain **contracts at a rate of** γ (usually very small), and **assumed density projection at each time step has error bounded by** ε (usually large) then the **expected error at every iteration is bounded by** ε/γ.

Example – BAT network [Forbes et al.]



BK results [Boyen, Koller '98]





error

time

error

time

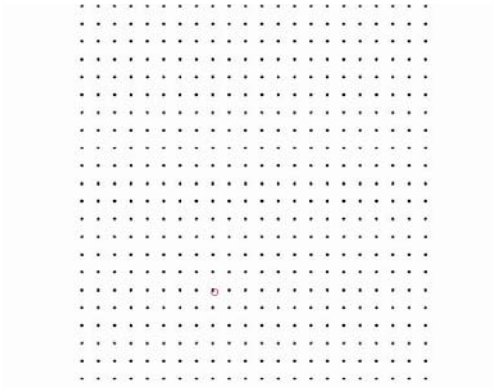
different choices of ST.

bounded error

Spikes because some observations introduce lots of error (that gets contracted later)

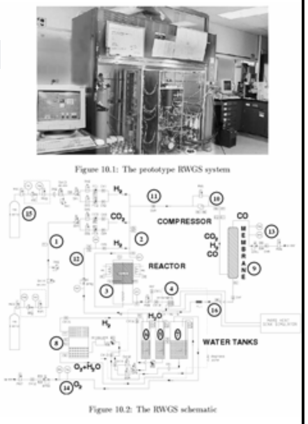
Thin Junction Tree Filters [Paskin '03]

- BK assumes fixed approximation clusters
- TJTF adapts clusters over time
  - ☐ attempt to minimize projection error



Hybrid DBN (many continuous and discrete variables)

- DBN with large number of discrete and continuous variables
- # of mixture of Gaussian components blows up in one time step!
- Need many smart tricks…
  - ☐ e.g., see Lerner Thesis



Reverse Water Gas Shift System (RWGS) [Lerner et al. '02]

DBN summary

- **DBNs**
  - ☐ factored representation of HMMs/Kalman filters
  - ☐ sparse representation does not lead to efficient inference
- **Assumed density filtering**
  - ☐ BK – factored belief state representation is assumed density
  - ☐ Contraction guarantees that error does blow up (but could still be large)
  - ☐ Thin junction tree filter adapts assumed density over time
  - ☐ Extensions for hybrid DBNs

This semester…

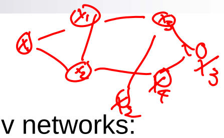
- Bayesian networks, Markov networks, factor graphs, decomposable models, junction trees, parameter learning, structure learning, semantics, exact inference, variable elimination, context-specific independence, approximate inference, sampling, importance sampling, MCMC, Gibbs, variational inference, loopy belief propagation, generalized belief propagation, Kikuchi, Bayesian learning, missing data, EM, Chow-Liu, IPF, GIS, Gaussian and hybrid models, discrete and continuous variables, temporal and template models, Kalman filter, linearization, switching Kalman filter, assumed density filtering, DBNs, BK, Causality,…
- **Just the beginning… ☺**

Quick overview of some hot topics...

- **Conditional Random Fields**
- **Maximum Margin Markov Networks**
- **Relational Probabilistic Models**
  - ☐ e.g., the parameter sharing model that you learned for a recommender system in HW1
- **Hierarchical Bayesian Models**
  - ☐ e.g., Khalid’s presentation on Dirichlet Processes
- **Influence Diagrams**

Generative v. Discriminative models – Intuition

$P(X,Y)$



- **Want to Learn**: h:**X** ↦ **Y**
  - ☐ **X** – features
  - ☐ **Y** – set of variables
- **Generative classifier**, e.g., Naïve Bayes, Markov networks:
  - ☐ Assume some **functional form for P(X|Y), P(Y)**
  - ☐ Estimate parameters of **P(X|Y)**, **P(Y)** directly from training data
  - ☐ Use Bayes rule to calculate **P(Y|X= x)**

    $P(Y / X = text)$

    webpage
  - ☐ This is a **‘*generative*’ model**
    - **Indirect** computation of **P(Y|X)** through Bayes rule
    - But, **can generate a sample of the data**, **P(X) = $\sum_y$ P(y) P(X|y)**
- **Discriminative classifiers**, e.g., Logistic Regression, Conditional Random Fields:

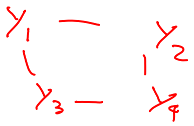
  $y^* = \underset{y}{\text{argmax}}\ P(y|X=x)$
  - ☐ Assume some **functional form for P(Y|X)**
  - ☐ Estimate parameters of **P(Y|X)** directly from training data
  - ☐ This is the **‘*discriminative*’ model**
    - Directly learn **P(Y|X)**, can have **lower sample complexity**
    - But **cannot obtain a sample of the data**, because P(X) is not available

Conditional Random Fields

[Lafferty et al. '01]

- Define a Markov network using a log-linear model for P(**Y**|X):

$P(Y|X) = \frac{1}{Z(X)} e^{\sum_i w_i f_i(X,Y)}$



- Features, e.g., for pairwise CRF:

$f_{17}(Y_1, Y_3, X)$

- Learning: maximize conditional log-likelihood
  - sum of log-likelihoods you know and love…
  - learning algorithm based on gradient descent, very similar to learning MNs

$\hat{w} = \underset{w}{\operatorname{argmax}} \log P_w(Y|X)$ ↑ in data

$\log P_w(Y_D | X_D) = \sum_j \log P(Y^{(j)} | X = x^{(j)})$

Max (Conditional) Likelihood

$P_w(Y|X) = \frac{1}{Z(X)} e^{\sum w_i f_i(X,Y)}$

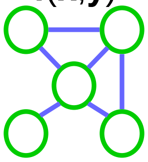
D

x^1,t(x^1) $= y^1$

…

x^m,t(x^m)

f(x,y)



Estimation

maximize$_{\mathrm{w}}$ $\sum_{\mathrm{x} \in D} \log P_{\mathrm{w}}(\mathrm{t}(\mathrm{x}) \mid \mathrm{x})$

Classification

arg max$_{\mathrm{y}}$ $\mathrm{w}^\top \mathrm{f}(\mathrm{x}, \mathrm{y})$

$\underset{y}{\operatorname{argmax}}\, g(y,x) = \underset{y}{\operatorname{argmax}} \log g(y,x)$

$\underset{y}{\operatorname{argmax}} \log P_{\mathrm{w}}(\mathrm{y} \mid \mathrm{x}) = \underbrace{\mathrm{w}^\top \mathrm{f}(\mathrm{x}, \mathrm{y})}_{\sum_i w_i f_i(x,y)} - \log \cancel{Z}_{\mathrm{w}}(\mathrm{x})$ → doesn't matter at classifi.

Don't need to learn entire distribution!

OCR Example

don't need Z(x)

- We want:

$$\text{argmax}_{\textbf{word}}\ \mathbf{w}^{T}\,\mathbf{f}(\underline{\text{brace}},\textbf{word}) = \text{“}\underline{\textbf{brace}}\text{”}$$

- Equivalently:

$$\left.\begin{array}{l}\mathbf{w}^{T}\,\mathbf{f}(\text{brace},\textbf{“brace”}) > \mathbf{w}^{T}\,\mathbf{f}(\text{brace},\textbf{“aaaaa”})\\ \mathbf{w}^{T}\,\mathbf{f}(\text{brace},\textbf{“brace”}) > \mathbf{w}^{T}\,\mathbf{f}(\text{brace},\textbf{“aaaab”})\\ \ldots\\ \mathbf{w}^{T}\,\mathbf{f}(\text{brace},\textbf{“brace”}) > \mathbf{w}^{T}\,\mathbf{f}(\text{brace},\textbf{“zzzzz”})\end{array}\right\}\ \textbf{a lot!}$$

Max Margin Estimation

- Goal: find **w** such that

$$\mathbf{w}^{T}\mathbf{f}(\mathbf{x},\mathbf{t}(\mathbf{x})) > \mathbf{w}^{T}\mathbf{f}(\mathbf{x},\mathbf{y}) \qquad \forall \mathbf{x}\in D \ \ \forall \mathbf{y}\neq \mathbf{t}(\mathbf{x})$$

$$\mathbf{w}^{T}\underbrace{[\mathbf{f}(\mathbf{x},\mathbf{t}(\mathbf{x})) - \mathbf{f}(\mathbf{x},\mathbf{y})]} > 0$$

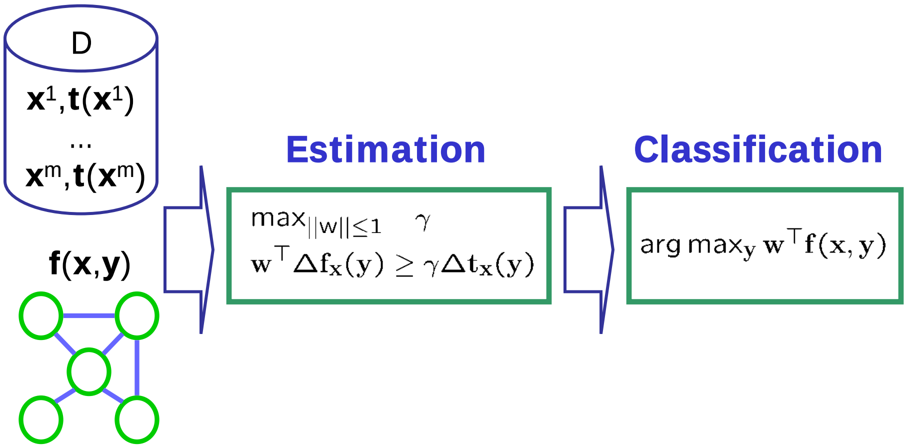
$$\boxed{\mathbf{w}^{\top}\Delta\mathbf{f}_{\mathbf{x}}(\mathbf{y}) \geq \gamma\Delta\mathbf{t}_{\mathbf{x}}(\mathbf{y})}$$

- Maximize $\underline{\text{margin}}\ \gamma$
- Gain over **y** grows with # of mistakes in **y**: $\Delta\mathbf{t}_{\mathbf{x}}(\mathbf{y})$

$\Delta\mathbf{t}_{\text{brace}}(\textbf{“craze”}) = 2$ $\qquad$ $\Delta\mathbf{t}_{\text{brace}}(\textbf{“zzzzz”}) = 5$

$\mathbf{w}^{\top}\Delta\mathbf{f}_{\text{brace}}(\textbf{“craze”}) \geq 2\gamma$ $\qquad$ $\mathbf{w}^{\top}\Delta\mathbf{f}_{\text{brace}}(\textbf{“zzzzz”}) \geq 5\gamma$

M³Ns: Maximum Margin Markov Networks [Taskar et al. '03]



D

$\mathbf{x}^1, \mathbf{t}(\mathbf{x}^1)$

…

$\mathbf{x}^m, \mathbf{t}(\mathbf{x}^m)$

$\mathbf{f}(\mathbf{x},\mathbf{y})$

Estimation

$\max_{||w||\le 1} \quad \gamma$

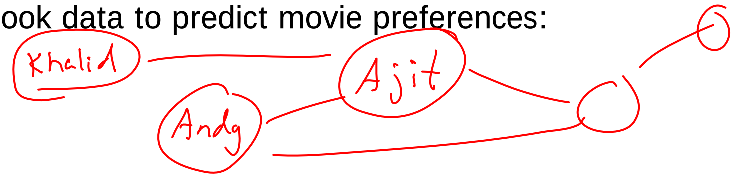
$\mathbf{w}^\top \Delta \mathbf{f}_\mathbf{x}(\mathbf{y}) \ge \gamma \Delta \mathbf{t}_\mathbf{x}(\mathbf{y})$

Classification

$\arg\max_\mathbf{y} \mathbf{w}^\top \mathbf{f}(\mathbf{x},\mathbf{y})$

Propositional Models and Generalization

- Suppose you learn a model for social networks for CMU from FaceBook data to predict movie preferences:



- How would you apply when new people join CMU?
- Can you apply it to make predictions a some "little technical college" in Cambridge, Mass?

Generalization requires Relational Models (e.g., see tutorial by Getoor)

- Bayes nets defined specifically for an instance, e.g., CMU FaceBook today
  - fixed number of people
  - fixed relationships between people
  - …
- Relational and first-order probabilistic models
  - talk about objects and relations between objects
  - allow us to represent different (and unknown) numbers
  - generalize knowledge learned from one domain to other, related, but different domains

---

Reasoning about decisions

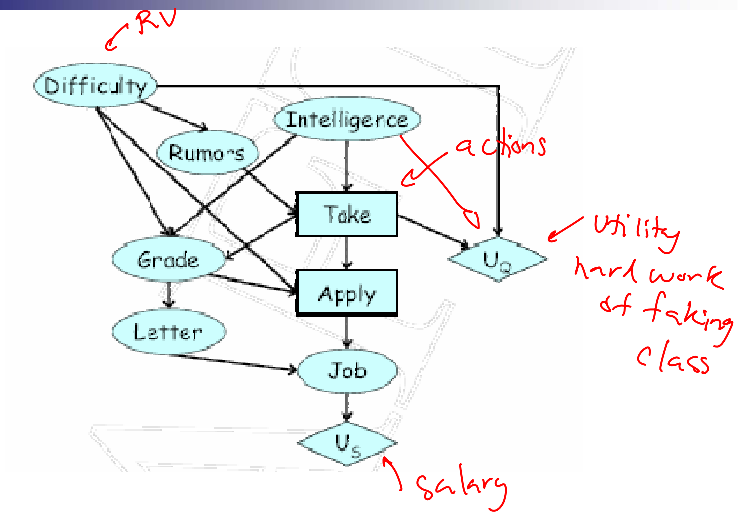
K&F Chapters 20 & 21

- So far, graphical models only have random variables

P(X)

- What if we could make decisions that influence the probability of these variables?
  - e.g., steering angle for a car, buying stocks, choice of medical treatment
- How do we choose the best decision?
  - the one that maximizes the expected long-term utility
- How do we coordinate multiple decisions?

Example of an Influence Diagram



Many, many, many more topics we didn’t even touch, e.g.,…

- **Active learning**
- **Non-parametric models**
- **Continuous time models**
- **Learning theory for graphical models**
- **Distributed algorithms for graphical models**
- **Graphical models for reinforcement learning**
- **Applications**
- **…**

What next?

- Seminars at CMU:
  - Machine Learning Lunch talks: http://www.cs.cmu.edu/~learning/
  - Intelligence Seminar: http://www.cs.cmu.edu/~iseminar/
  - Machine Learning Department Seminar: http://calendar.cs.cmu.edu/cald/seminar
  - Statistics Department seminars: http://www.stat.cmu.edu/seminar
  - …

- Journal:
  - JMLR – Journal of Machine Learning Research (free, on the web)
  - JAIR – Journal of AI Research (free, on the web)
  - …

- Conferences:
  - UAI: Uncertainty in AI
  - NIPS: Neural Information Processing Systems
  - Also ICML, AAAI, IJCAI and others

- Some MLD courses:
  - 10-705 Intermediate Statistics (Fall)
  - 10-702 Statistical Foundations of Machine Learning (Spring)
  - 10-801 Advanced Topics in Graphical Models: statistical foundations, approximate inference, and Bayesian methods (Spring)
  - …