Readings: K&F 4.4, 4.5, 4.6

Undirected Graphical Models II

Lecture 5 – Apr 11, 2011
CSE 515, Statistical Methods, Spring 2011

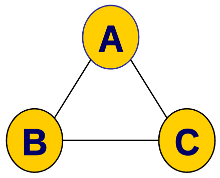
Instructor: Su-In Lee
University of Washington, Seattle

Last time

- Markov networks representation
  - Local factor models (potentials) $\pi_1[\mathbf{D}_1],\ldots,\pi_n[\mathbf{D}_n]$
  - Independence properties
    - Global, pairwise, local independencies
  - I-Map ↔ Factorization $P(X_1,\ldots,X_n)=\frac{1}{Z}\prod \pi_i[\mathbf{D}_i]$
- Today…
  - Parameterization revisited
  - Bayesian nets and Markov nets
  - Partially directed graphs
  - Inference 101

Factor Graphs

- From the Markov network structure, we are do not know how it is parameterized.
  - Example: fully connected graph may have pairwise potentials or one large (exponential) potential over all nodes



Markov network

$$P(A,B,C) = \frac{1}{Z}\prod \pi_i[\mathbf{D}_i]$$

$$P_{\text{mc}}(A,B,C) = \frac{1}{Z_{\text{mc}}}\pi_{ABC}[A,B,C]$$

$$P_{\text{pair}}(A,B,C) = \frac{1}{Z_{\text{pair}}}\pi_{AB}[A,B]\pi_{BC}[B,C]\pi_{AC}[A,C]$$

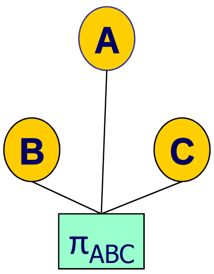
:

- Solution: Factor Graphs
  - Undirected graph
  - Two types of nodes: Variable nodes, Factor nodes
  - Connectivity?

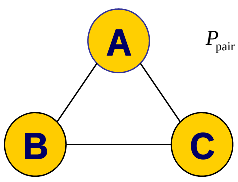
Factor Graphs: example

- Two types of nodes: Variable nodes, Factor nodes
- Connectivity
  - Each factor node is associated with exactly one factor $\pi_i[\boldsymbol{D}_i]$
  - Scope of factor are all neighbor variables of the factor node

$$P_{\text{mc}}(A,B,C) = \frac{1}{Z_{\text{mc}}}\pi_{ABC}[A,B,C]$$

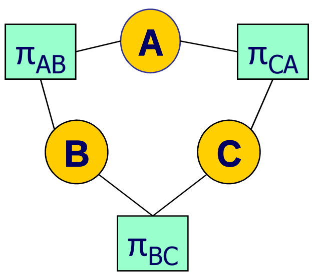


Factor graph for joint parameterization



Markov network

$$P_{\text{pair}}(A,B,C) = \frac{1}{Z_{\text{pair}}}\pi_{AB}[A,B]\pi_{BC}[B,C]\pi_{AC}[A,C]$$

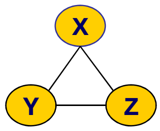


Factor graph for pairwise parameterization

Local Structure: Feature Representation

- Factor graphs still encode complete tables

X	Y	π_{XY}[X,Y]
x^0	y^0	100
x^0	y^1	1
x^1	y^0	1
x^1	y^1	100



- A feature ϕ[**D**] on variables **D** is an indicator function that for some $d \in$ **D**: for example,

$$\phi[X,Y] = \begin{cases} 1 & \text{when } x = y \\ 0 & \text{otherwise} \end{cases}$$

- Several features can be defined on one clique

$$\phi_1[\mathbf{D}] = \begin{cases} 1 & \text{when } x = y \\ 0 & \text{otherwise} \end{cases} \qquad \phi_2[\mathbf{D}] = \begin{cases} 1 & \text{when } x > 50 \\ 0 & \text{otherwise} \end{cases}$$

→ Any factor can be represented by features, where in general case, we define a feature and weight for each entry in the factor

- Apply log-transformation: π_i[D] = exp ($-w_i \phi_i$[**D**])

Log-linear model

- A distribution P is a log-linear model over H if it has
  - Features $\phi_1[\mathbf{D_1}],\ldots,\phi_k[\mathbf{D_k}]$ where each $\mathbf{D_i}$ is a complete subgraph in H
  - A set of weights $w_1,\ldots,w_k$ such that

$$P(X_1,\ldots,X_n) = \frac{1}{Z}\prod \pi_i[\mathbf{D_i}] = \frac{1}{Z}\exp\left[-\sum_{i=1}^{k} w_i \phi_i[\mathbf{D_i}]\right]$$

- Advantages
  - Log-linear model is more compact for many distributions especially with large domain variables
    - Representation is intuitive and modular – Features can be modularly added between any interacting sets of variables

Markov Network Parameterizations

- Choice 1: Markov network
  - Product over potentials
  - Right representation for discussing independence queries
- Choice 2: Factor graph
  - Product over potentials
  - Useful for inference (later)
- Choice 3: Log-linear model
  - Product over feature weights
  - Useful for discussing parameterizations
  - Useful for representing context specific structures
- All parameterizations are interchangeable

Outline

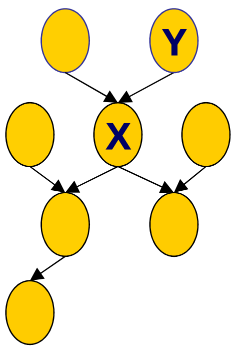
- Markov networks representation
  - Local factor models $\pi_1[\mathbf{D}_1],\ldots,\pi_n[\mathbf{D}_n]$
  - Independencies
    - global, pairwise, local independencies
  - I-Map ↔ Factorization $P(X_1,\ldots,X_n) = \frac{1}{Z}\prod \pi_i[\mathbf{D}_i]$
- Today…
  - Parameterization revisited
  - Bayesian nets and Markov nets
  - Partially directed graphs
  - Inference 101

From Bayesian nets to Markov nets

- Goal: build a Markov network H capable of representing any distribution P that factorizes over G
  - Equivalent to requiring $I(H) \subseteq I(G)$
- Construction process
  - Based on local Markov independencies
    - If X is connected with **Y** in H, $(X \perp \mathcal{U} - \{X\} - \mathbf{Y} \mid \mathbf{Y})$.
  - Connect each X to every node in the smallest set **Y** s.t.: $\{(X \perp \mathcal{U} - \{X\} - \mathbf{Y} \mid \mathbf{Y}) : X \in H\} \subseteq I(G)$
  - How can we find Y by querying G?
    - Y = Markov blanket of X in G?

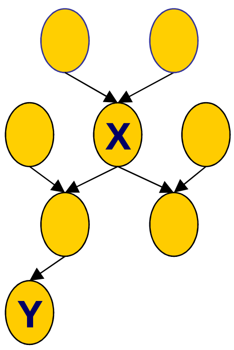
Blocking Paths

Active path: parents



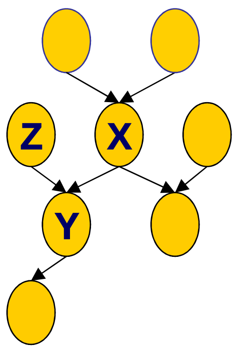
Block path: parents

Active path: descendants



Block path: children

Active path: v-structure



Block path: children children's parents

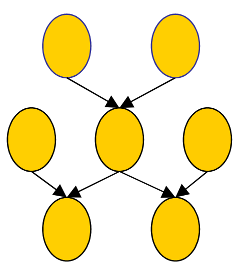
From Bayesian nets to Markov nets

- Goal: build a Markov network H capable of representing any distribution P that factorizes over G
  - Equivalent to requiring $I(H) \subseteq I(G)$

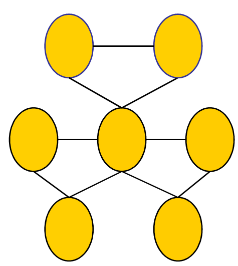
- Construction process
  - Based on local Markov independencies
    - If X is connected with **Y** in H, $(X \perp \mathcal{U}\text{-}\{X\}\text{-}\mathbf{Y} \mid \mathbf{Y})$.
  - Connect each X to every node in the smallest set **Y** s.t.: $\{(X \perp \mathcal{U}\text{-}\{X\}\text{-}\mathbf{Y} \mid \mathbf{Y}) : X \in H\} \subseteq I(G)$
  - How can we find Y by querying G?
    - Y = Markov blanket of X in G (parents, children, children's parents)

Moralized Graphs

- The Moral graph of a Bayesian network structure G is the undirected graph that contains an undirected edge between X and Y if
  - X and Y are directly connected in G
  - X and Y have a common child in G



Bayesian network G

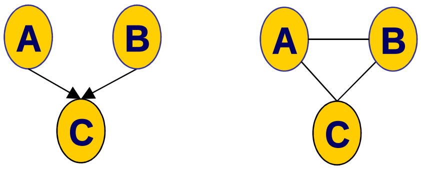


Moralized graph H

Parameterizing Moralized Graphs

- Moralized graph contains a full clique for every X_i and its parents $Pa(X_i)$
  → We can associate CPDs with a clique

- Do we lose independence assumptions implied by the graph structure?
  - Yes, immoral v-structures



$(A \perp B)$

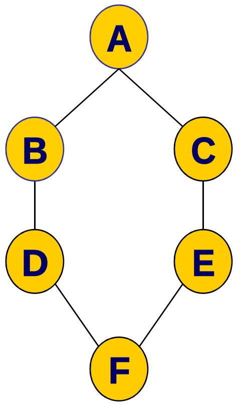
From Markov nets to Bayesian nets

- Transformation is more difficult and the resulting network can be much larger than the Markov network

- Construction algorithm
  - Use Markov network as template for independencies I(H)
  - Fix ordering of nodes
  - Add each node along with its minimal parent set according to the independencies defined in the distribution

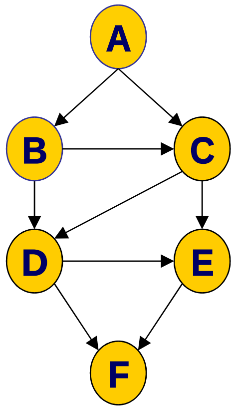
From Markov nets to Bayesian nets

Markov network *H*



Order: A,B,C,D,E,F

Bayesian network *G*



Chordal Graphs

- Let X_1—X_2—...—X_k—X_1 be a loop in the graph
- A chord in the loop is an edge connecting X_i and X_j for two nonconsecutive nodes X_i and X_j

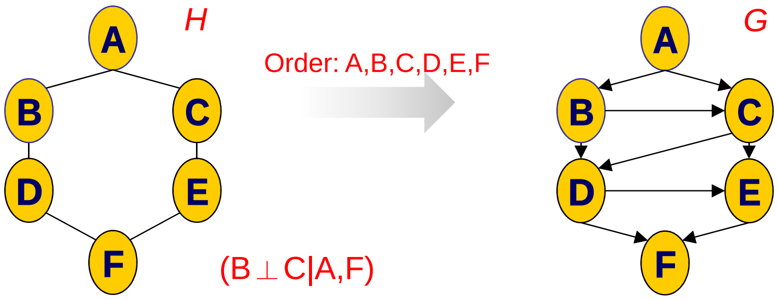
- An undirected graph is chordal if any loop X_1—X_2—...—X_k—X_1 for $k \geq 4$ has a chord
  - That is, longest minimal loop is a triangle
  - Chordal graphs are often called triangulated

- A directed graph is chordal if its underlying undirected graph is chordal

From Markov Nets to Bayesian Nets

- Theorem: Let H be a Markov network structure and G be any minimal I-map for H. Then G is chordal.

- The process of turning a Markov network into a Bayesian network is called triangulation
  - The process loses independencies



Last time

- Markov networks representation
  - Local factor models $\pi_1[\mathbf{D}_1],\dots,\pi_n[\mathbf{D}_n]$
  - Independencies
    - global, pairwise, local independencies
  - I-Map ↔ Factorization $P(X_1,\dots,X_n)=\frac{1}{Z}\prod \pi_i[\mathbf{D}_i]$

- Today…
  - Parameterization revisited
  - Bayesian nets and Markov nets
  - Partially directed graphs
  - Inference 101

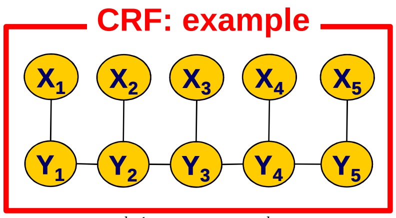
Conditional Random Fields (CRFs)

- Special case of partially directed models
- A conditional random field is an undirected graph *H* whose nodes correspond to **X**U**Y**; the network is annotated with a set of factors $\varphi_1(\mathbf{D}_1),\ldots, \varphi_m(\mathbf{D}_m)$ such that each $\mathbf{D}_i$ **X.** The network encodes a conditional distribution as follows:

$$\tilde{P}(\mathbf{Y},\mathbf{X}) = \prod_{i=1}^{m} \phi_i(\mathbf{D}_i)$$

$$P(\mathbf{Y} \mid \mathbf{X}) = \frac{1}{Z(\mathbf{X})} \tilde{P}(\mathbf{Y},\mathbf{X})$$

$$Z(\mathbf{X}) = \sum_{\mathbf{Y}} \tilde{P}(\mathbf{Y},\mathbf{X})$$



$$\tilde{P}(\mathbf{Y},\mathbf{X}) = \prod_{i=1}^{k-1} \phi_i(Y_i, Y_{i+1}) \prod_{i=1}^{k} \phi_i(Y_i, X_i)$$

- Two variables in *H* are connected by an undirected edge whenever they appear together in the scope of some factor ϕ.

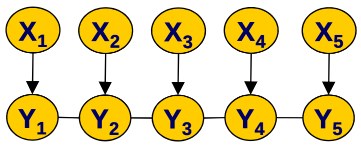
Why Conditional?

- Why P(**Y**|**X**), not P(**Y**,**X**)?
  - The network explicitly does not encode any distribution over the variables in **X**.
    - → One of the main strengths of the CRF representation.
- This flexibility allows us to do many things:
  - Incorporating into the model a rich set of observed variables **X** whose dependencies may be quite complex or even poorly understood.
  - Including continuous variables **X** whose distribution may not have a simple parametric form
  - Using domain knowledge in order to define a rich set of features characterizing our domain, without worrying about modeling their joint distribution.
- Many applications: Computer vision (detail later), text analysis, part-of-speech labeling, many more

Conditional Random Fields

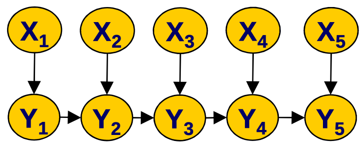
- Directed and undirected dependencies.
- A CRF defines conditional distribution of **Y** on **X**, P(**Y**|**X**).
  - It can be viewed as a partially directed graph, where we have an undirected component over Y, which has the variables in X as parents.
- Any difference with Bayesian networks?

CRF
(partially directed)



$$\tilde{P}(\mathbf{Y},\mathbf{X}) = \prod_{i=1}^{k-1} \phi_i(Y_i, Y_{i+1}) \prod_{i=1}^{k} \phi_i(Y_i, X_i)$$

Conditional Bayesian network (fully directed)

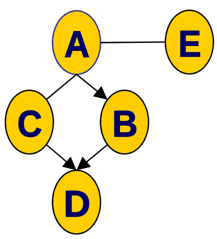


$$P(\mathbf{Y} \mid \mathbf{X}) = \prod_{i=1}^{k-1} P(Y_{i+1} \mid Y_i) \prod_{i=1}^{k} P(Y_i \mid X_i)$$

Chain Networks

- Combines Markov networks and Bayesian networks
- Partially directed graph (PDAG)
- As for undirected graphs, we have three distinct interpretations for the independence assumptions implied by a P-DAG

Example:



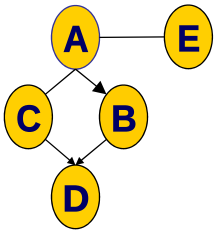
Pairwise Independencies

- Every node X is independent from any node which is not its descendant given all non-descendants of X
- Formally:
  - $I_P(K) = \{(X \perp Y | ND(X)-\{X,Y\}) : X—Y \notin K, Y \in ND(X)\}$

Example:

(D ⊥ A | B,C,E)

(C ⊥ E | A)

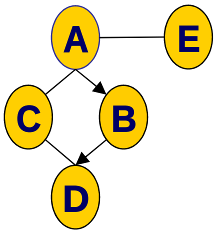


Local Markov Independencies

- Let Boundary(X) be the union of the parents of X and the neighbors of X
- Local Markov independencies state that a node X is independent of its non-descendants given its boundary
- Formally:
  - $I_L(K) = \{(X \perp ND(X)-Boundary(X) | Boundary(X)) : X \in \mathcal{U}\}$

Example:

(D ⊥ A,E | B,C)



Global Independencies

- I(K) = {($\mathbf{X} \perp \mathbf{Y} | \mathbf{Z}$) : **X**,**Y**,**Z**, **X** is c-separated from **Y** given **Z**}

- **X** is c-separated from **Y** given **Z** if **X** is separated from **Y** given **Z** in the undirected moralized graph M[K]

- The moralized graph of a P-DAG K is an undirected graph M[K] by
  - Connecting any pair of parents of a given node
  - Converting all directed edges to undirected edges

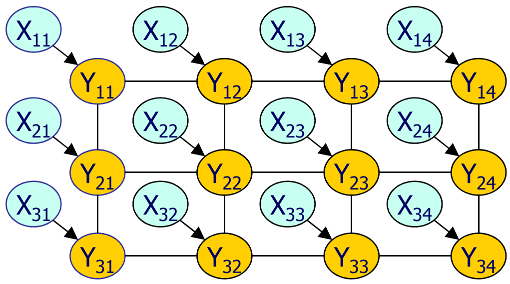
For positive distributions: $I(K) \leftrightarrow I_L(K) \leftrightarrow I_P(K)$

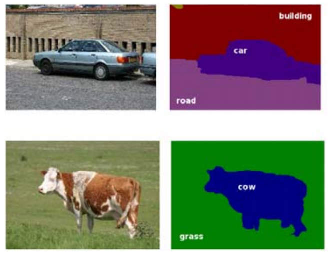
Domain Application: Vision

- The image segmentation problem
  - Task: Partition an image into distinct parts of the scene
  - Example: separate water, sky, background

Markov Network for Segmentation

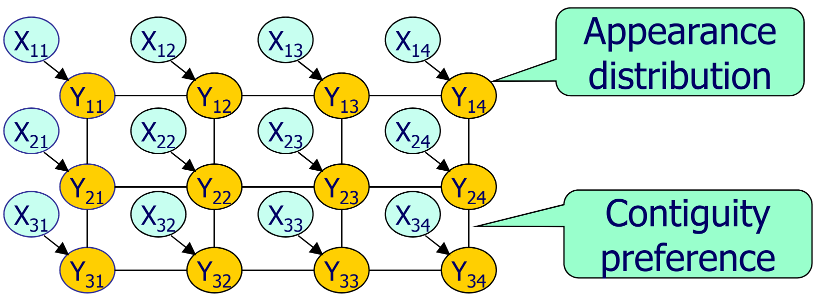
- Grid structured Markov network (CRF)
- Random variables (X_i, Y_i) correspond to pixel i
  - X_i: Input image for pixel i (always given)
    - Color, texture, location …
  - Y_i: Domain is {1,…K} (e.g. 1:road, 2:car, 3:bldg) (generally not given)
    - Value represents region assignment to pixel i
- Neighboring pixels are connected in the network



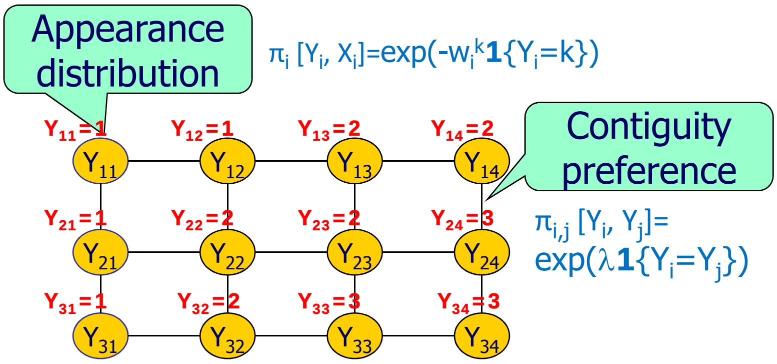


Markov Network for Segmentation

- Node potentials (appearance distribution)
  - Introduce node potential $\exp(-w_i^k \mathbf{1}\{Y_i=k\})$
  - w_i^k – extent to which pixel i "fits" region k (e.g., based on X_i containing various info such as color, location, texture on pixel i)
- Edge potentials (contiguity preference)
  - Encodes contiguity preference by edge potential $\exp(\lambda \mathbf{1}\{Y_i=Y_j\})$ for $\lambda>0$



Markov Network for Segmentation



$\pi_i [Y_i, X_i]=\exp(-w_i^k \mathbf{1}\{Y_i=k\})$

$\pi_{i,j} [Y_i, Y_j]= \exp(\lambda \mathbf{1}\{Y_i=Y_j\})$

Y=1: sky, Y=2: car, Y=3: building

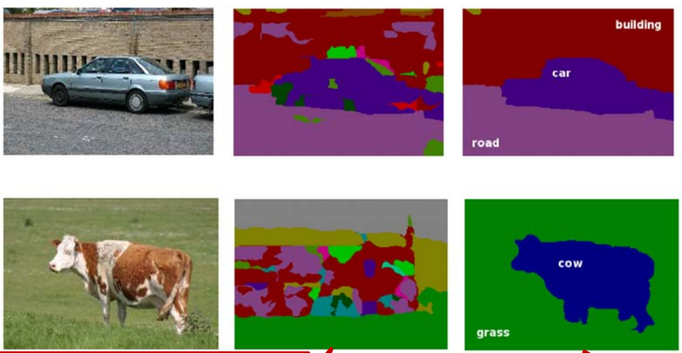
- Solution: inference on the pairwise Markov network
  - Find most likely assignment k (=sky, building, etc) to Y_i variables

$$P(\mathbf{Y} \mid \mathbf{X}) = \frac{1}{Z(\mathbf{X})} \prod \pi_i [Y_i, X_i] \prod \pi_{i,j} [Y_i, Y_j]$$

Example Results

Input 1

Input 2



Baseline (a simple classifier): Result of segmentation **using node potentials alone**, so that each pixel is classified independently

Result of segmentation **using a pairwise Markov network** encoding interactions between adjacent pixels

Last time

- Markov networks representation
  - Local factor models $\pi_1[\mathbf{D}_1],\ldots,\pi_n[\mathbf{D}_n]$
  - Independencies
    - global, pairwise, local independencies
  - I-Map ↔ Factorization $P(X_1,\ldots,X_n)=\frac{1}{Z}\prod\pi_i[\mathbf{D}_i]$

- Today…
  - Parameterization revisited
  - Bayesian nets and Markov nets
  - Partially directed graphs
  - Inference 101

Inference

- Markov networks and Bayesian networks represent a joint probability distribution

⇓

- Networks contain information needed to answer any query about the distribution
- Inference is the process of answering such queries

- Direction between variables does not restrict queries
- Inference combines evidence from all network parts

Likelihood Queries

- Compute probability (=likelihood) of the evidence
  - Evidence: subset of variables **E** and an assignment ***e***
  - Task: compute P(**E**=***e***)

- Computation

$$P(\mathbf{E}=\mathbf{e}) = \sum_{z \in \mathbf{U}-\mathbf{E}} P(\mathbf{Z}=\mathbf{z}, \mathbf{E}=\mathbf{e})$$

Conditional Probability Queries

- Conditional probability queries
  - Evidence: subset of variables **E** and an assignment ***e***
  - Query: a subset of variables **Y**
  - Task: compute P(**Y** | **E**=***e***)

- Applications
  - Medical and fault diagnosis
  - Genetic inheritance

- Computation

$$P(\mathbf{Y}=\mathbf{y} \mid \mathbf{E}=\mathbf{e}) = \frac{P(\mathbf{Y}=\mathbf{y}, \mathbf{E}=\mathbf{e})}{P(\mathbf{E}=\mathbf{e})} = \frac{\sum_{w \in \mathbf{U}-\mathbf{Y}-\mathbf{E}} P(\mathbf{W}=\mathbf{w}, \mathbf{Y}=\mathbf{y}, \mathbf{E}=\mathbf{e})}{\sum_{z \in \mathbf{U}-\mathbf{E}} P(\mathbf{Z}=\mathbf{z}, \mathbf{E}=\mathbf{e})}$$

Maximum A Posteriori Assignment

- Maximum A Posteriori Assignment (MAP)
  - Evidence: subset of variables **E** and an assignment ***e***
  - Query: a subset of variables **Y**
  - Task: compute MAP(**Y**|**E**=**e**) = argmax$_y$ P(**Y**=**y** | **E**=***e***)
  - Note 1: there may be more than one possible solution
  - Note 2: equivalent to computing

    argmax$_y$ P(**Y**=**y** , **E**=***e***)

    Why? P(**Y**=**y** | **E**=***e***) = P(**Y**=**y** , **E**=***e***) / P(**E**=***e***)

- Computation

$$\text{MAP}(\mathbf{Y}=\mathbf{y} \mid \mathbf{e}) = \text{argmax}_{y'} \sum_{\mathbf{w} \in \mathbf{U}-\mathbf{Y}-\mathbf{E}} P(\mathbf{W}=\mathbf{w}, \mathbf{Y}=\mathbf{y}' \mid \mathbf{E}=\mathbf{e})$$

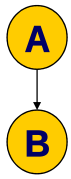
Most Probable Assignment: MPE

- Most Probable Explanation (MPE)
  - Evidence: subset of variables **E** and an assignment ***e***
  - Query: **all other variables** **Y** (**Y**=**U**-**E**)
  - Task: compute MPE(**Y**|**E**=**e**) = argmax$_y$ P(**Y**=**y** | **E**=***e***)
  - Note: there may be more than one possible solution

- Applications
  - Decoding messages: find the most likely transmitted bits
  - Diagnosis: find a single most likely consistent hypothesis

Most Probable Assignment: MPE

- Note: We are searching for the most likely joint assignment to all variables
  - May be different than most likely assignment (MAP) of each variable.
  - Any example?



- Given **E**= ϕ
- $P(a^1)>P(a^0)$ → $MAP(A) = a^1$
- $MPE(A,B) = \{a^0, b^1\}$
  - $P(a^0, b^0) = 0.04$
  - $P(a^0, b^1) = 0.36$
  - $P(a^1, b^0) = 0.3$
  - $P(a^1, b^1) = 0.3$

P(A)

/	
a^0	a^1
0.4	0.6

P(B|A)

	B	
A	B^0	B^1
a^0	0.1	0.9
a^1	0.5	0.5

Exact Inference in Graphical Models

- Graphical models can be used to answer
  - Conditional probability queries
  - MAP queries
  - MPE queries

- Naïve approach
  - Generate joint distribution
  - Depending on query, compute sum/max
    → Exponential blowup

- Exploit independencies for efficient inference

Summary: Markov network representation

- Markov Networks – undirected graphical models
  - Like Bayesian networks, define independence assumptions
  - Three definitions exist, all equivalent in positive distributions
  - Factorization is defined as product of factors over complete sub-graph
- Alternative parameterizations
  - Factor graphs
  - Log-linear models
- Relationship to Bayesian networks
  - Represent different families of independencies
  - Moralization – transforming Bayesian networks to Markov networks.
  - Triangulation – transforming Markov networks to Bayesian networks.
- Partially directed graphs
  - Conditional random fields (CRFs)
  - Application to image segmentation

Announcements

- Feedback on the course
  - Email your comments anonymously.
  - See the course website.
- Additional OH
  - Tuesday in the morning 9-10am
- Slightly modified course outline

Where are we? What next?

Week	Dates	Topics and Lecture Notes	Readings
I. Probabilistic Graphical Models Representation			
1	3/28	Introduction to the class	2.1, 2.2, 2.3
	3/30	Bayesian network representation	3.1, 3.2, 3.3
2	4/4	Local probability models	3.4, 5
	4/6	Undirected graphical models I	4.1, 4.2, 4.3
3	4/11	Undirected graphical models II + P-DAGs	4.4, 4.5, 4.6
II. Exact Inference			
	4/13	Inference: exact inference	9.1, 9.2, 9.3
4	4/18	Exact inference in BNs	9.4, 9.5, 9.6
	4/20	Exact inference: Clique Trees	10.1, 10.2, 10.3, 10.4
III. Learning			
5	4/25	Learning: parameter estimation	17
	4/27	Parameter learning in BNs	17
6	5/2	Structure learning in BNs	18
	5/4	Partially observed data (learning with missing data)	19
7	5/9	More on learning (TBD)	
IV. Approximate Inference			
	5/11	Approximate inference: particle-based I	12
8	5/16	Approximate inference: particle-based II	12
	5/18	Global approximate inference I	11
9	5/23	Global approximate inference II	11
V. Special Topics & Applications			
	5/25	Markov Decision Processes (Instructor: Mausam)	
10	5/30	(memorial day)	
	6/1	Temporal models (DBNs, HMMs)	
		Final examination @	

1. Probabilistic model **representation**

2. Exact inference in BNs
- P(**X**=x|**E**=e)=?

3. Learning parameters/structure
- Learning CPDs, structure from data

4. Approximate inference
- P(**X**=x|**E**=e)≈?

5. Applications
- Decision making, temporal processes

Acknowledgement

- These lecture notes were generated based on the slides from Prof Eran Segal.