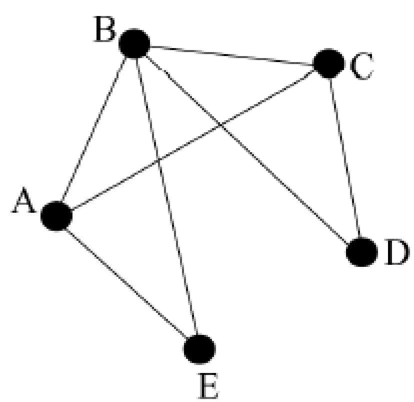
Reasoning with uncertainty

- (Very) basic review of probability and uncertainty
- Joint distribution and inference
- Exploiting independences
- Special case: Directed graphs
- General case: Undirected graphs, factor graphs
- Examples

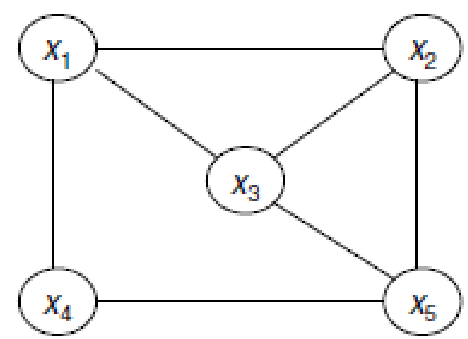
- Deterministic:
  - Represent facts and constraints
  - Find configuration that satisfies representation
- Examples:
  - Clauses $A \wedge B \Rightarrow C$
  - Satisfiability problems

$\varphi = \{(\neg C), (A \vee B \vee C), (\neg A \vee B \vee E), (\neg B \vee C \vee D)\}.$



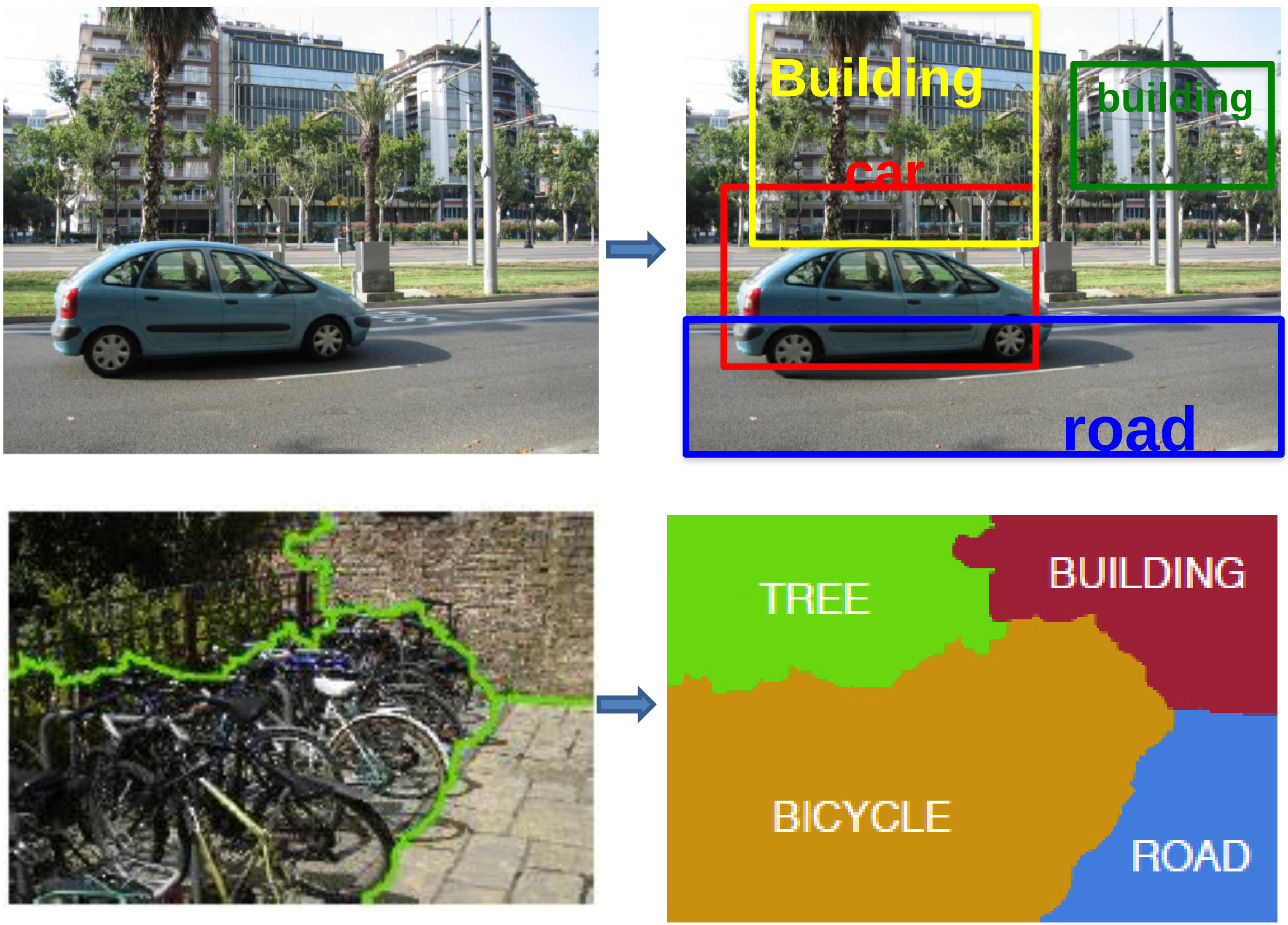
$f_1(x_1, x_2, x_3)$
$f_2(x_2, x_3, x_5)$
$f_3(x_1, x_4)$
$f_4(x_4, x_5)$

$$\min_{t \in Sol}\{\sum_{i=1}^{m'} f_i(t)\}$$



- Generalization to include uncertainty due to imperfect knowledge
  - Variables: Deterministic → Random variables
  - Constraints: Deterministic functions (e.g., CNF, CSP, SAT) → continuous output
- Similar problems, generalization

- What we have: scores from noisy classifiers from local features (P(label|image features))
- What would like:



- What we get:

Reasoning

- Need to use knowledge about the world
- Need to integrate uncertainty in “sensing”
- Scenes:
  - “road scenes” contain “cars”, “building”….
  - “Office scenes” contain “desks”, “computers”…
- Co-occurrence:
  - “keyboard” implies “mouse”
- Location:
  - “Cars” are on top of “roads”
  - “Sky” is above “buildings”

Reasoning with uncertainty

- Need use *uncertain* knowledge about the world
- Scenes:
  - “road scenes” *is likely to* contain “cars”, “building”….
  - “Office scenes” *is likely to* contain “desks”, “computers”…
- Co-occurrence:
  - “keyboard” *usually* implies “mouse”
- ……
- Probably possible to represent uncertainty on each individual piece of knowledge
- Intractable to integrate them all to find the “optimal” interpretation

Probability Reminder

- Conditional probability for 2 events A and B:

$$P(A|B) = \frac{P(A,B)}{P(B)}$$

- Chain rule:

$$P(A,B) = P(A|B)\,P(B)$$

Probability Reminder

- Conditional probability for 2 variables X and Y:

$$P(X=x \mid Y=y) = \frac{P(X=x,Y=y)}{P(Y=y)}$$

- Chain rule:

$$P(X=x,Y=y) = P(X=x|Y=y)\ P(Y=y)$$

- For any values x,y

The Joint Distribution

- Joint distribution = collection of all the probabilities P(X = x,Y = y,Z = z…) for all possible combinations of values.
- For m binary variables, size is 2^m
- Any query can be computed from the joint distribution

X	Y	Z	Prob
T	T	T	0.1
T	T	F	0.22
T	F	T	0.2
T	F	F	0.08
F	T	T	0.1
F	T	F	0.15
F	F	T	0.07
F	F	F	0.08

The Joint Distribution

- Any query can be computed from the joint distribution
- Marginal distribution

  $P(X = True), P(X = False)$

- Conditional distribution:

  $P(X = True \mid Y = True) =$

  $P(X = True, Y = True)/P(Y = True)$

- In general:

  $P(E_1 \mid E_2) = P(E_1, E_2)/P(E_2)$

  $$P(E_2) = \sum_{\text{Entries that match } E_2} P(\text{Joint Entries})$$

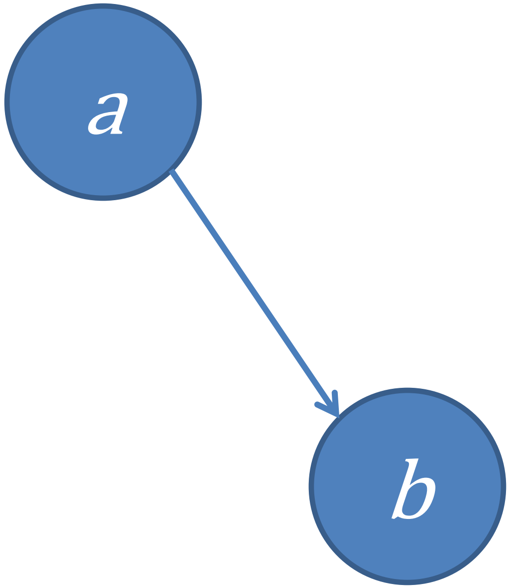
X	Y	Z	Prob
T	T	T	0.1
T	T	F	0.22
T	F	T	0.2
T	F	F	0.08
F	T	T	0.1
F	T	F	0.15
F	F	T	0.07
F	F	F	0.08

Summary

- Any query computable from
- Sum rule:
- Product rule

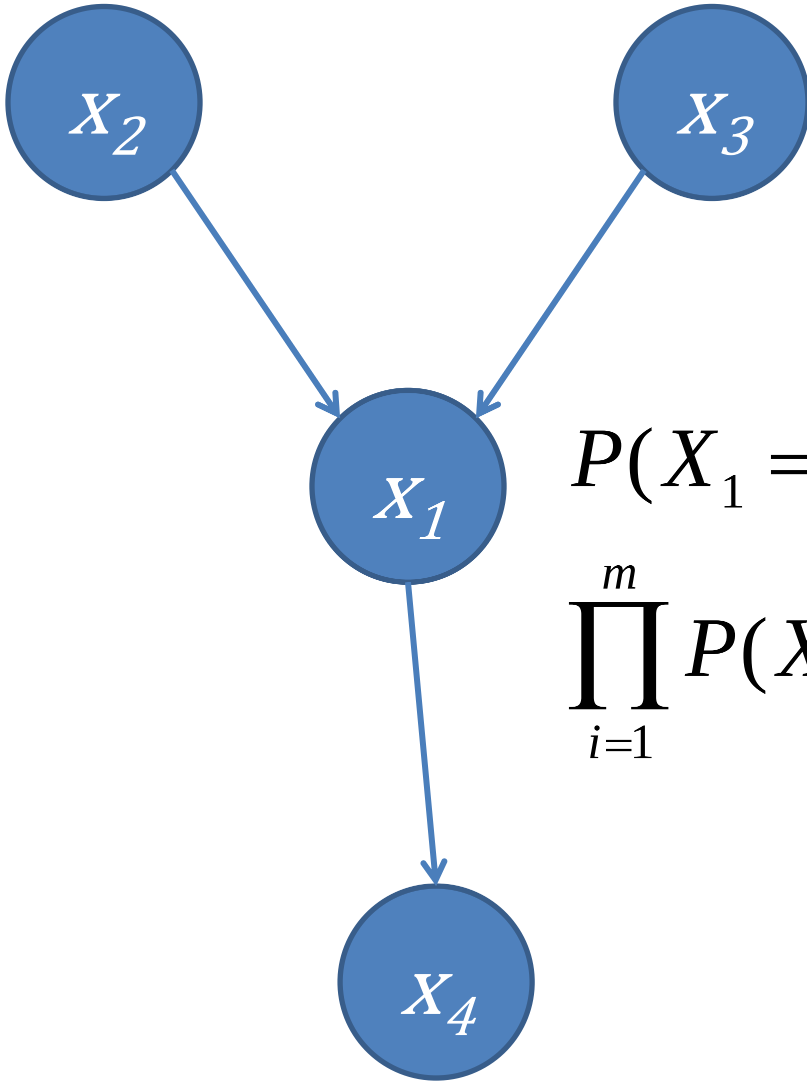
But requires entire joint distribution →
Represent dependencies between variables

First case: Directed



$$P(a, b) = P(b|a)P(a)$$

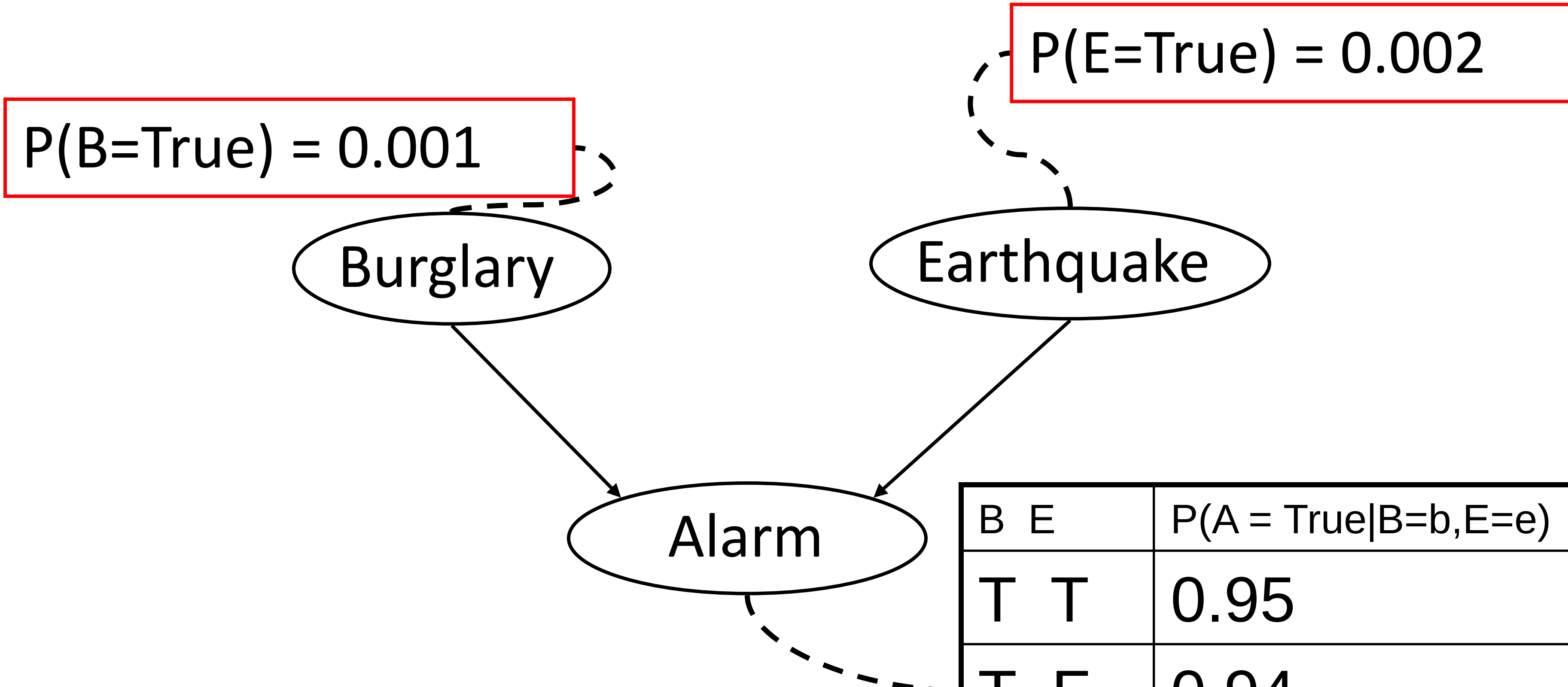
First case: Directed



$$P(X_1 = x_1, X_2 = x_2, \ldots X_m = x_m) =$$

$$\prod_{i=1}^{m} P(X_i = x_i \mid \mathrm{Parents}(X_i))$$

Graphical Representation



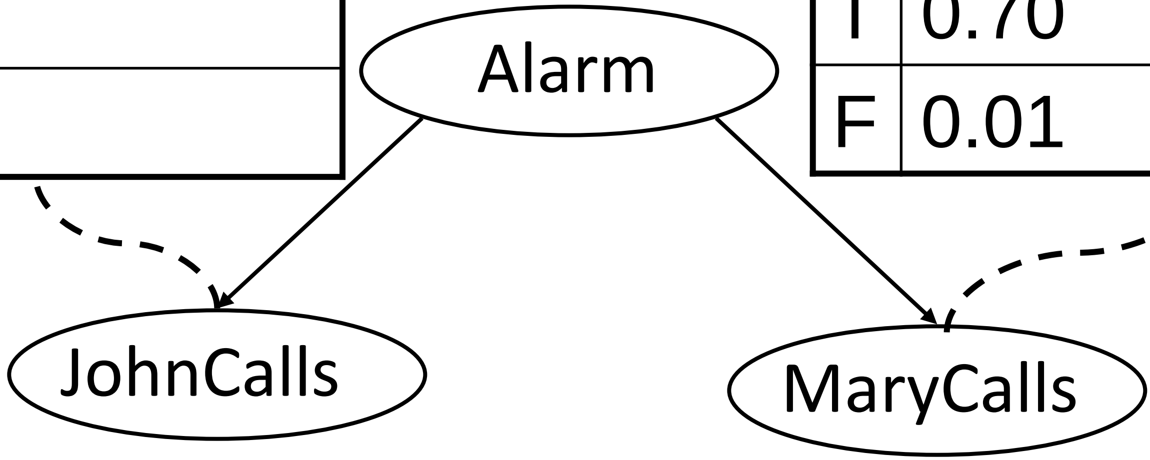
B	E	P(A = True|B=b,E=e)
T	T	0.95
T	F	0.94
F	T	0.29
F	F	0.001

$$P(A,J,M) = P(A|B,E)P(B)P(E)$$

Graphical Representation

A	P(J = True|A=a)
T	0.90
F	0.05

A	P(M = True|A= a)
T	0.70
F	0.01



Given knowledge of A, knowing anything else in the diagram won't help with J and M

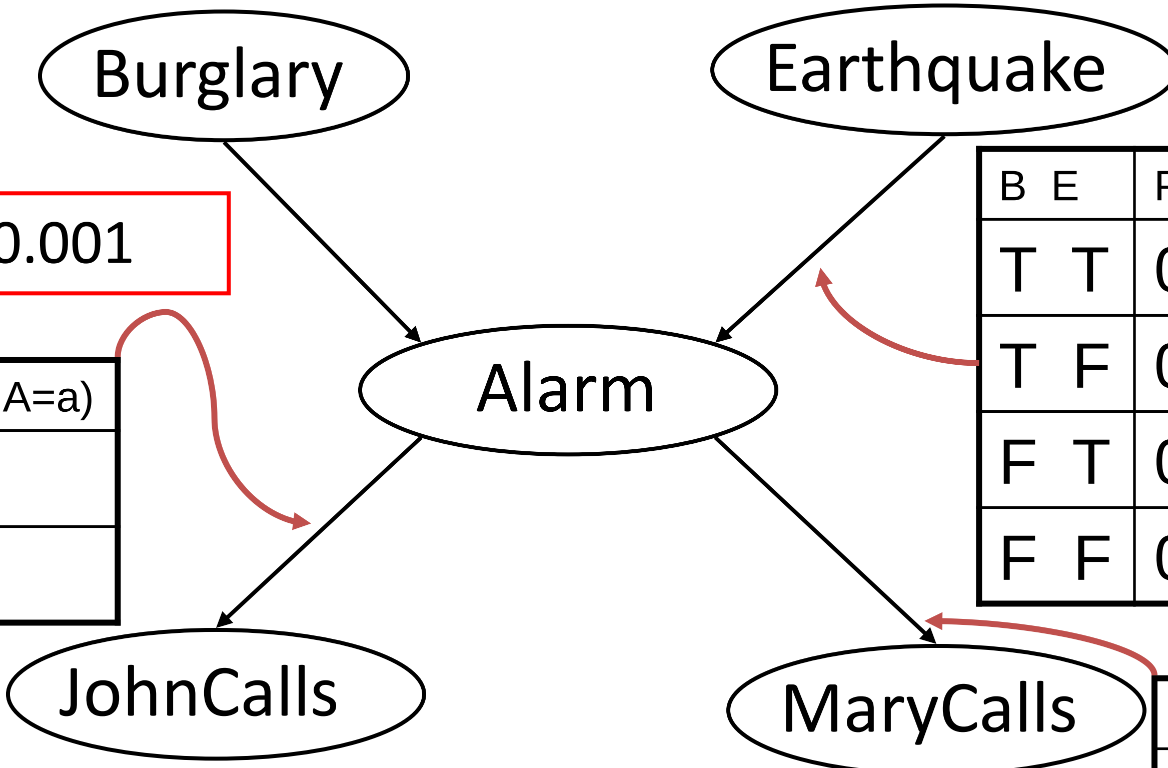
$$P(A,J,M) = P(A)P(J|A)P(M|A)$$

Inference

- *Any* inference operation of the form P(values of some variables | values of the other variables) can be computed

P(E=true) = 0.002

P(B=True) = 0.001



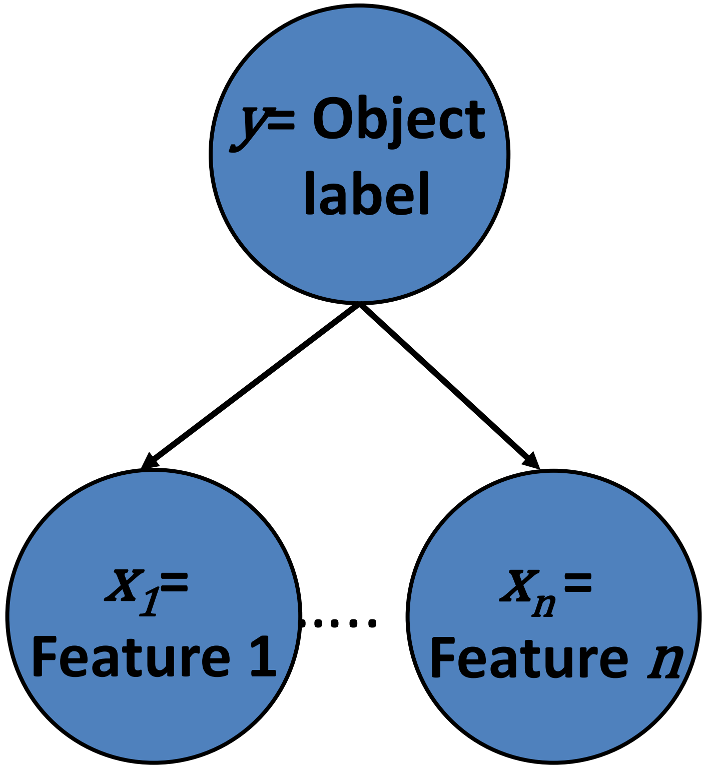
B	E	P(A = True|B=b,E=e)
T	T	0.95
T	F	0.94
F	T	0.29
F	F	0.001

A	P(J = True|A=a)
T	0.90
F	0.05

A	P(M = True|A=a)
T	0.70
F	0.01

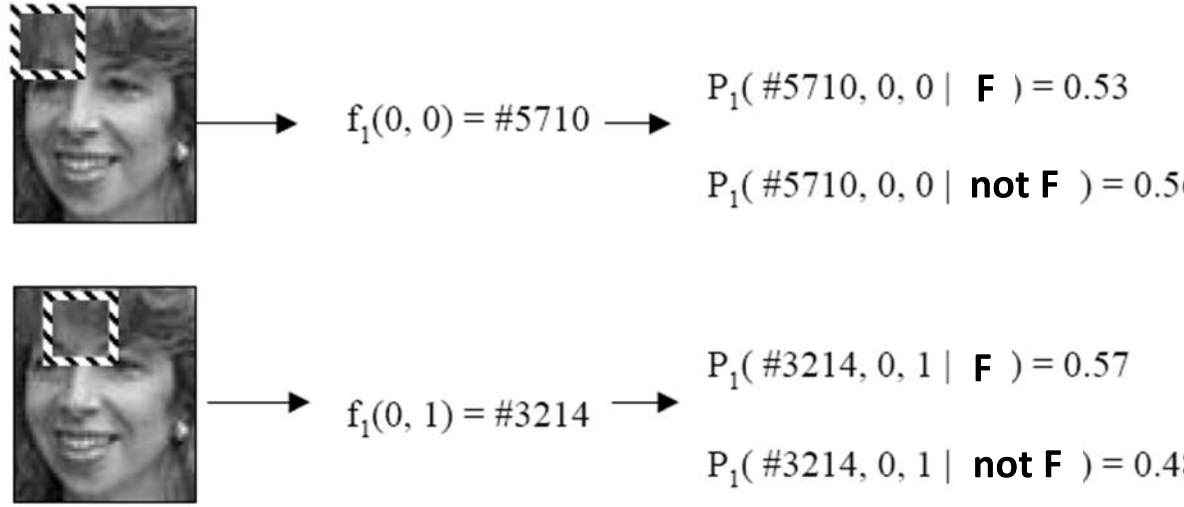
$$P(A,J,M,B,E) = P(B)P(E)P(A|B,E)P(J|A)P(M|A)$$

Example Naïve Bayes classification



$$P(x_1,..,x_n|y) = \prod_i P(x_i|y)$$

Example



$y = 1$ if face

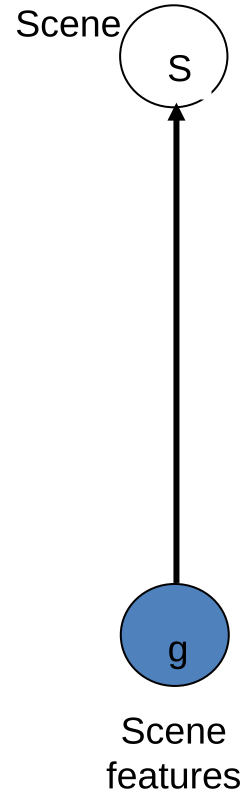
- Lots of (discretized) features from local filters
- Estimate likelihood ratio

$$\frac{P(x_1,..,x_n|y=face)}{P(x_1,..,x_n|y=not\ face)}$$

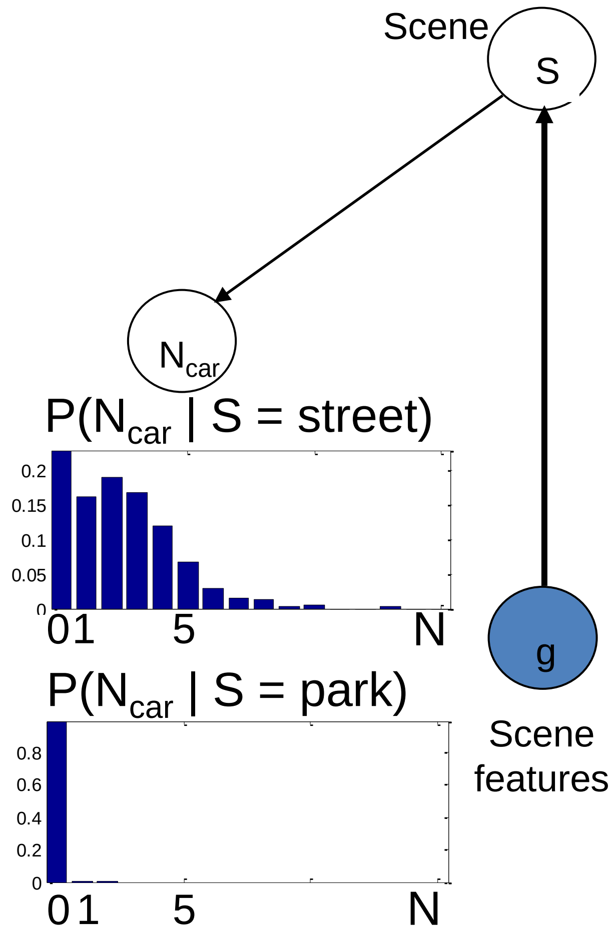
Example

- Need use *uncertain* knowledge about the world
- Scenes:
  - "road scenes" *is likely to* contain "cars", "building"….
  - "Office scenes" *is likely to* contain "desks", "computers"…
- Co-occurrence:
  - "keyboard" *usually* implies "mouse"
- ……
- Probably possible to represent uncertainty on each individual piece of knowledge
- Intractable to integrate them all to find the "optimal" interpretation

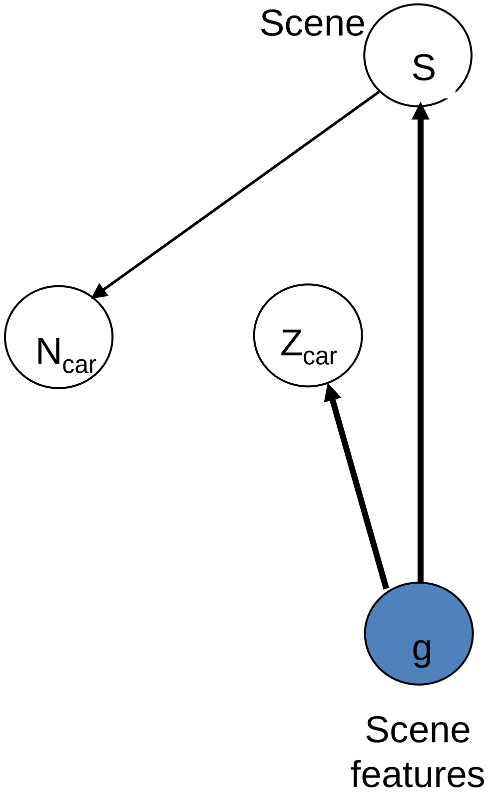
Examples in the next few slides from: Murphy, Torralba, Freeman; NIPS 2003. Torralba, Murphy, Freeman, CACM 2010.



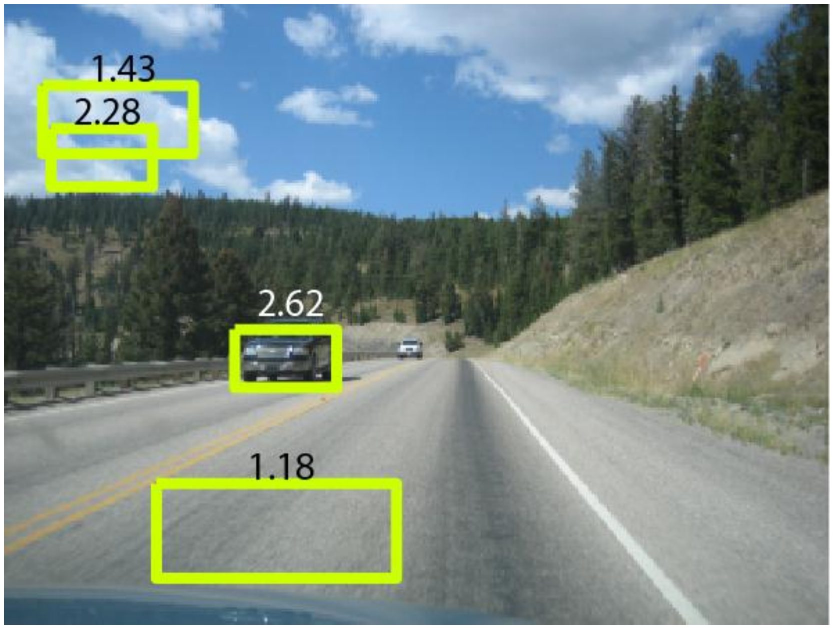
Murphy, Torralba, Freeman; NIPS 2003. Torralba, Murphy, Freeman, CACM 2010.



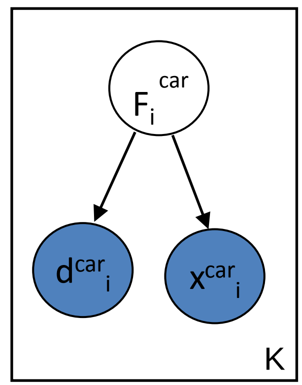
Murphy, Torralba, Freeman; NIPS 2003. Torralba, Murphy, Freeman, CACM 2010.


Scene
S
N_{car}
Z_{car}
g
Scene
features


STOP



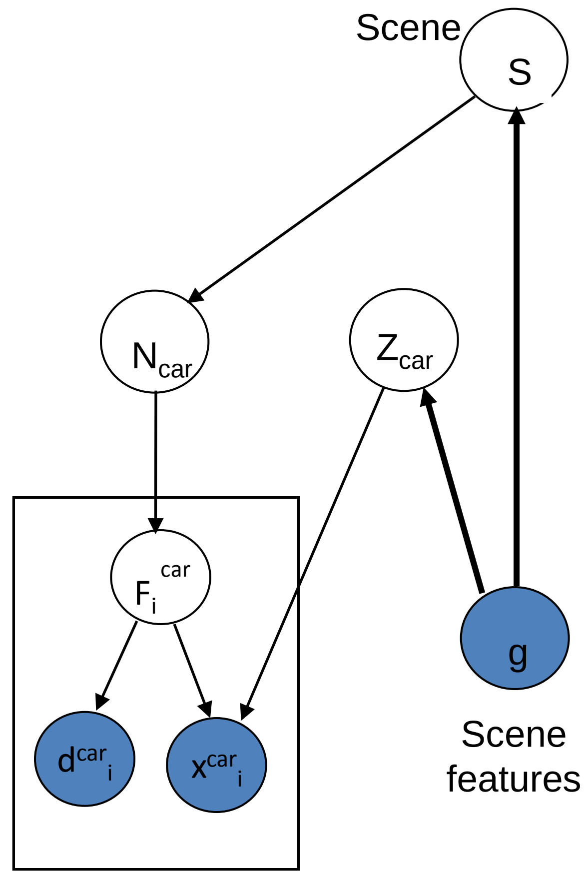
Multiview car detector.

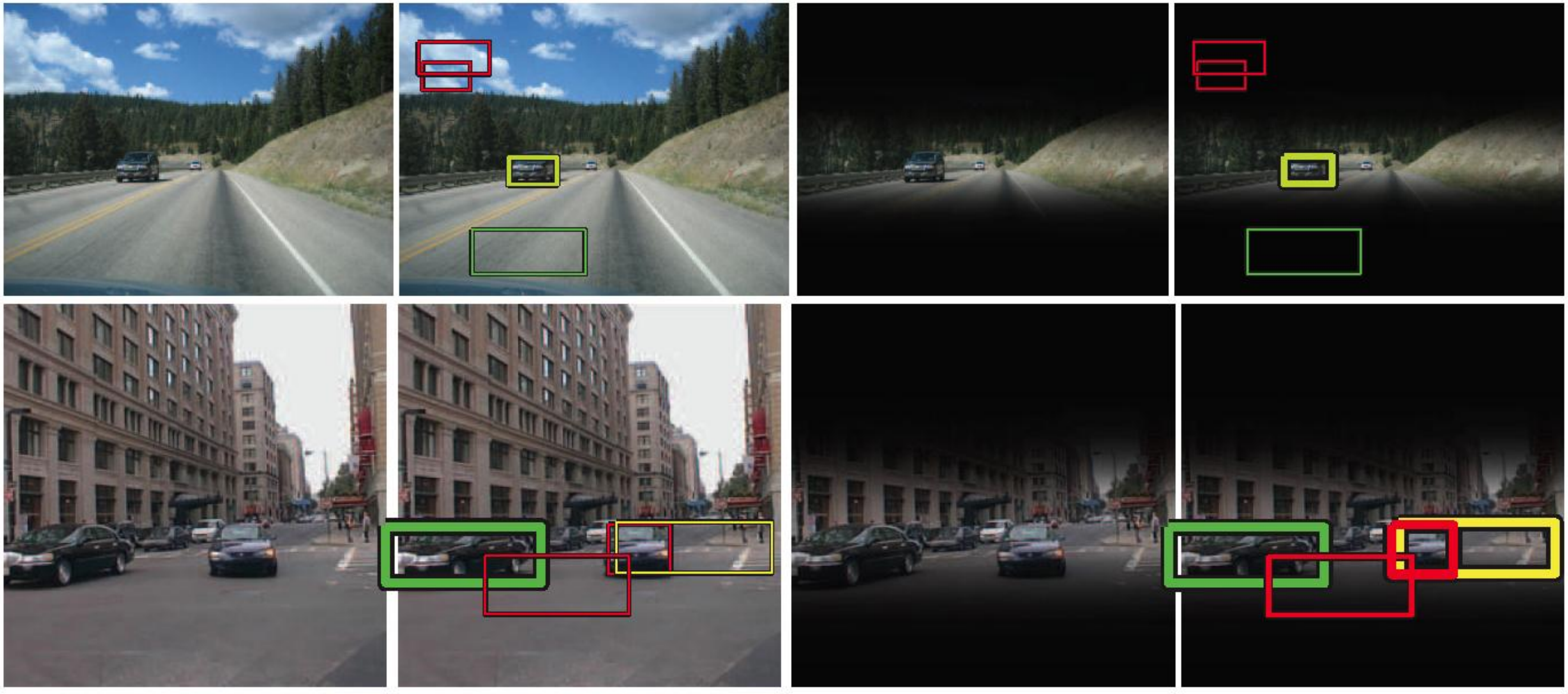


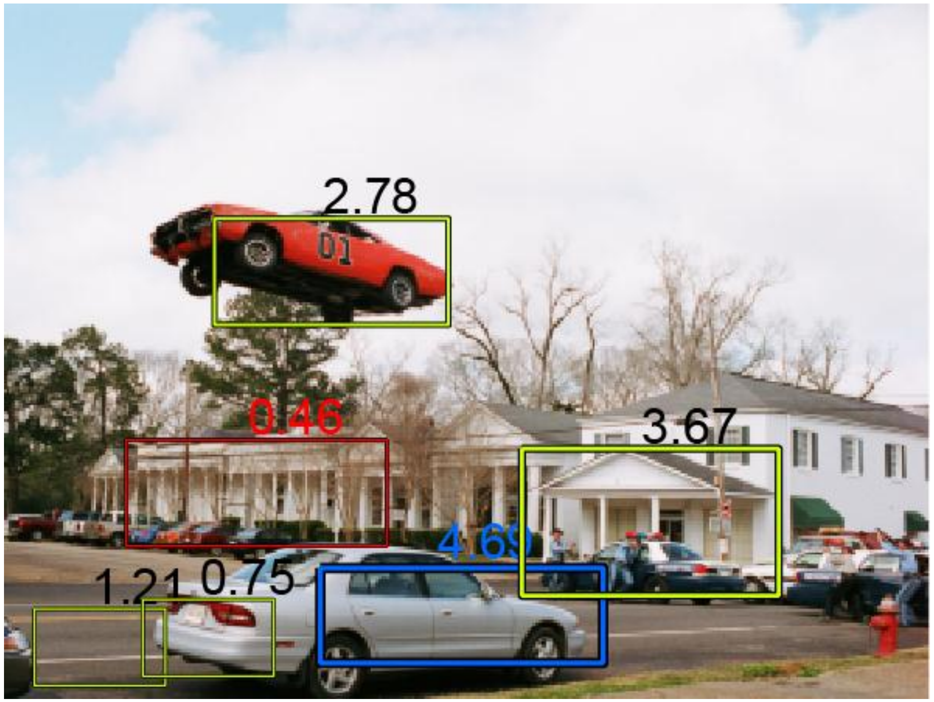
F = 1 if car present in box

p(d | F=1)

An integrated model of Scenes, Objects, and Parts







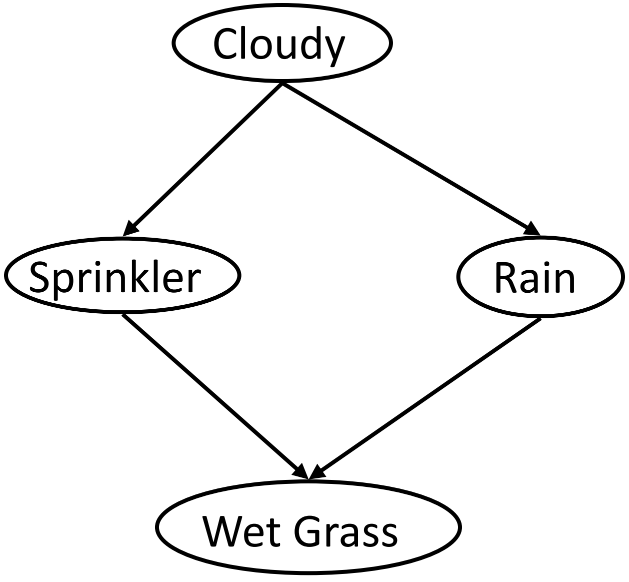
- No miracle: Fancy representation can only model the knowledge that we encoded.

Example from Antonio Torralba

Inference

- Can answer any query but : Need to sum over the possible assignments of the hidden variables.
  - Variable elimination
  - Separation

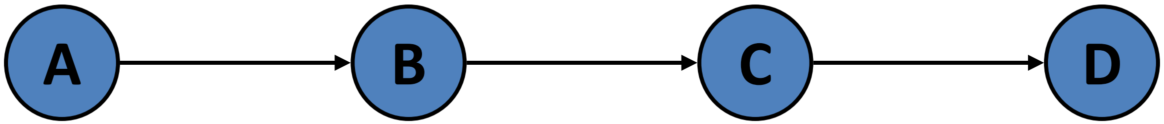
- Query variables: E_1
- Evidence variables: E_2
- The rest, E_3



P(W | Cloudy = True)

- E_1 = {W}
- E_2 = {Cloudy=True}
- E_3 = {Sprinkler, Rain}

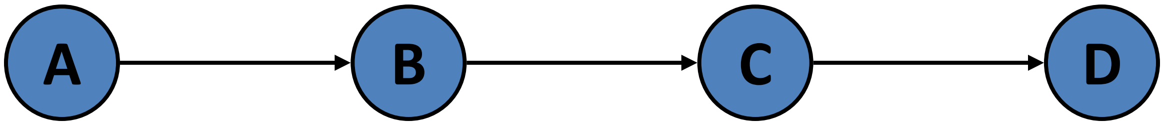
Inference: A Simple Case



- Suppose that we want to compute

P(D = d) from this network.

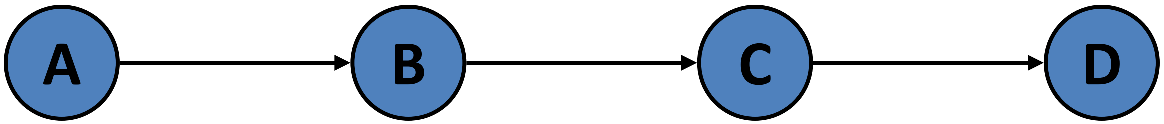
A Simple Case



- Compute P(D = d) by summing the joint probability over all possible values of the remaining variables A, B, and C:

$$P(D=d)=\sum_{a,b,c} P(A=a,B=b,C=c,D=d)$$

A Simple Case

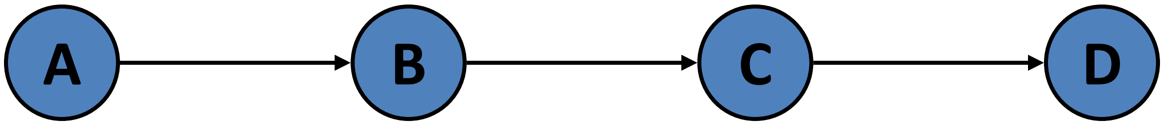


- Decompose the joint by using the fact that it is the product of terms of the form:

P(X | Parents(X))

$$P(D=d)=\sum_{a,b,c} P(D=d \mid C=c)P(C=c \mid B=b)P(B=b \mid A=a)P(A=a)$$

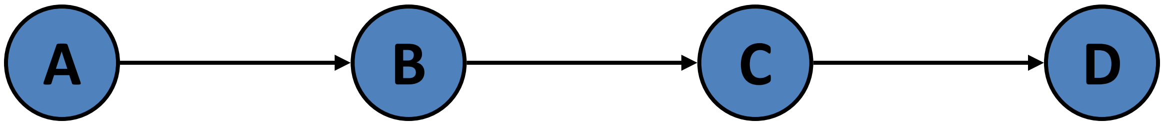
A Simple Case



- We can avoid computing the sum for all possible triplets (A,B,C) by distributing the sums inside the product

$$P(D=d)=\sum_{c}P(D=d\mid C=c)\sum_{b}P(C=c\mid B=b)\sum_{a}P(B=b\mid A=a)P(A=a)$$

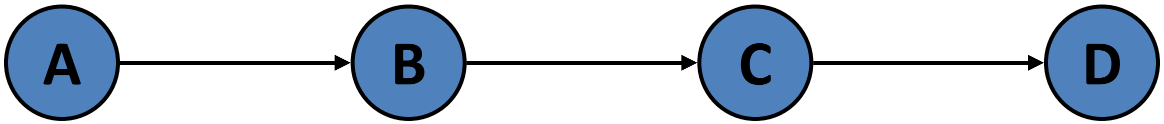
A Simple Case



This term depends only on B and can be written as a function $f_A(b)$

$$P(D=d)=\sum_{c} P(D=d \mid C=c)\sum_{b} P(C=c \mid B=b)\boxed{\sum_{a} P(B=b \mid A=a)P(A=a)}$$

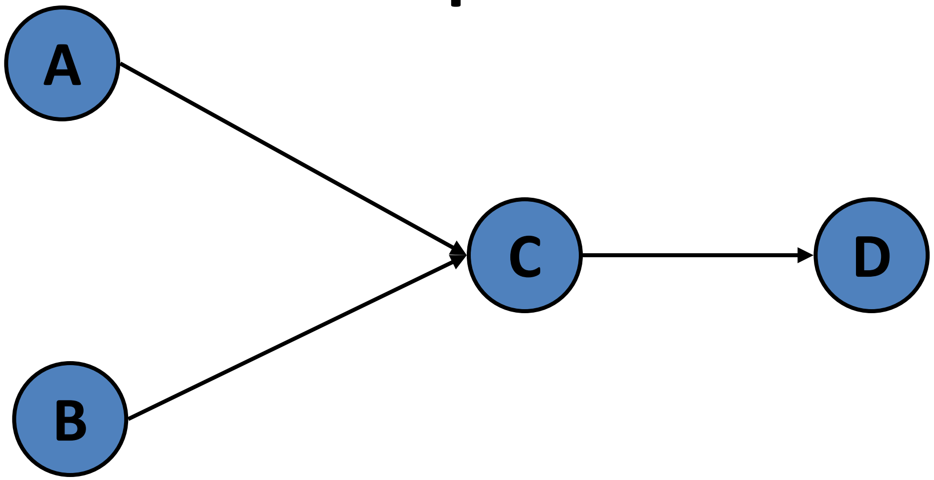
A Simple Case



This term depends only on c and can be written as a function $f_B(c)$

$$P(D=d)=\sum_{c} P(D=d \mid C=c) \boxed{\sum_{b} P(C=c \mid B=b) f_A(b)}$$

Example



$$P(D=d)=\sum_{a,b,c} P(D=d \mid C=c)P(C=c \mid B=b, A=a)P(B=b)P(A=a)$$

$$=\sum_{c} P(D=d \mid C=c)\sum_{b} P(B=b)\sum_{a} P(C=c \mid B=b, A=a)P(A=a)$$

$$=\sum_{c} P(D=d \mid C=c)\sum_{b} P(B=b)\sum_{a} f_1(a,b,c)$$

$$=\sum_{c} P(D=d \mid C=c)\sum_{b} f_2(b,c)$$

General Case: Variable Elimination

- Write the desired probability as a sum over all the unassigned variables

$$P(D=d)=\sum_{a,b,c} P(A=a, B=b, C=c, D=d)$$

- Distribute the sums inside the expression
  - Pick a variable
  - Group together all the terms that contain this variable

$$P(D=d)=\sum_{c} P(D=d \mid C=c)\sum_{b} P(C=c \mid B=b)\boxed{\sum_{a} P(B=b \mid A=a)P(A=a)}$$

  - Represent as a single function of the variables appearing in the group

$$P(D=d)=\sum_{c} P(D=d \mid C=c)\sum_{b} P(C=c \mid B=b) f_A(b)$$

  - Repeat until no more variables are left

General Case: Variable Elimination

- Write the desired probability as a sum over all the unassigned variables

Computation exponential in the size of the largest group → The order in which the variables are selected is important.

on

- – Pic e
- – Group ner all the terms that contain this variable

$$P(D=d)=\sum_{c} P(D \quad d \mid C=c)\sum_{b} P(C=c \mid B=b)\boxed{\sum_{a} P(B=b \mid A=a)P(A=a)}$$

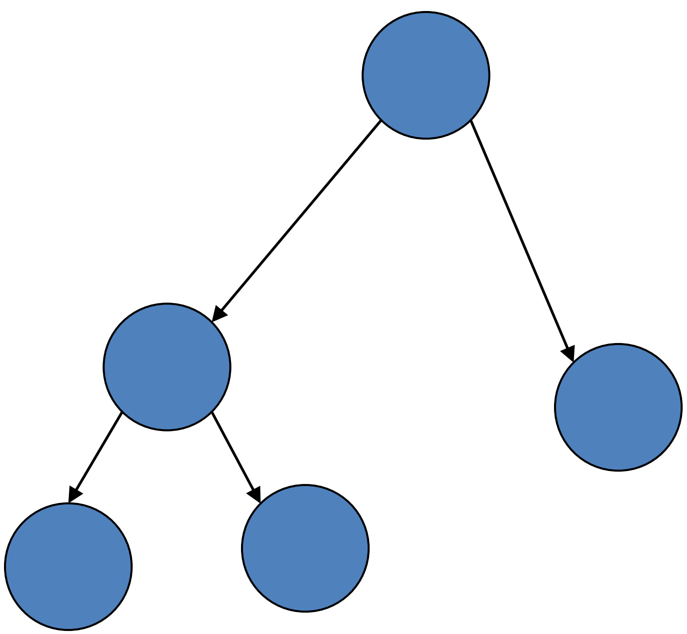
- – Represent as a single function of the variables appearing in the group

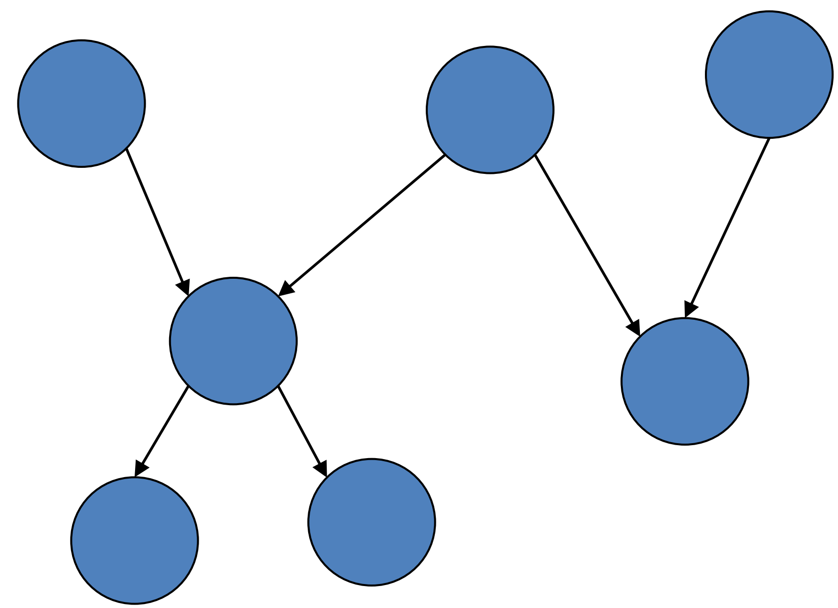
$$P(D=d)=\sum_{c} P(D=d \mid C=c)\sum_{b} P(C=c \mid B=b)f_{A}(b)$$

- – Repeat until no more variables are left

Special Case

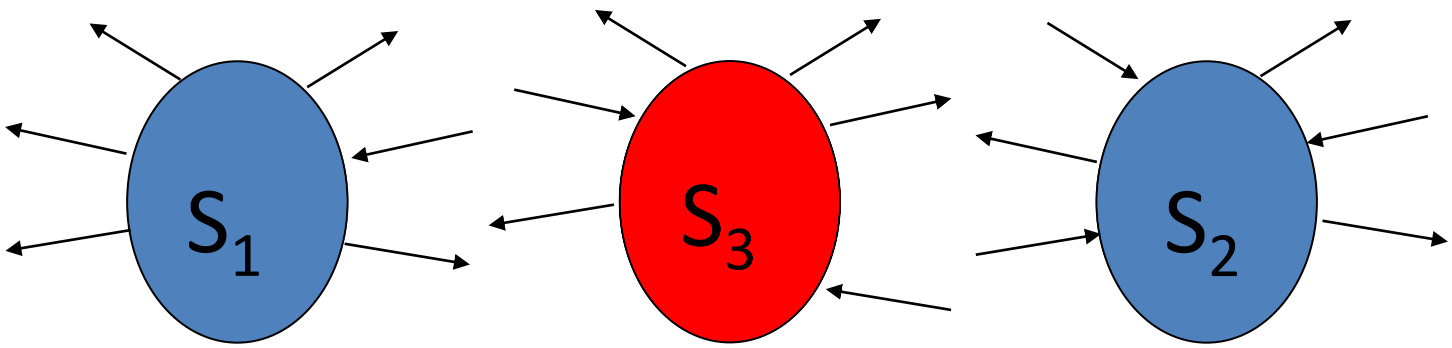
- Polytrees: Undirected version of the graph is a tree = there is a single undirected path between two nodes
- In this case: Inference linear in the number of nodes ($d^{k+1}n$)
- General case: See later approximate inference (e.g., sampling)





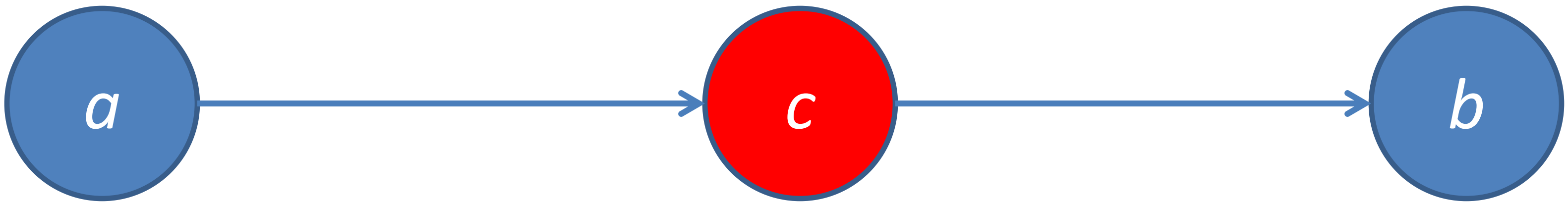
Conditional independence

- P(any assignments to S_1 | any assignments to S_2 ,any assignments to S_3) = P(assignment to S_1 | assignments to S_3)
- P(any assignments to $S_{1,}$ any assignments to S_2 |any assignments to S_3) = P(assignment to S_1)P(assignments to S_2)

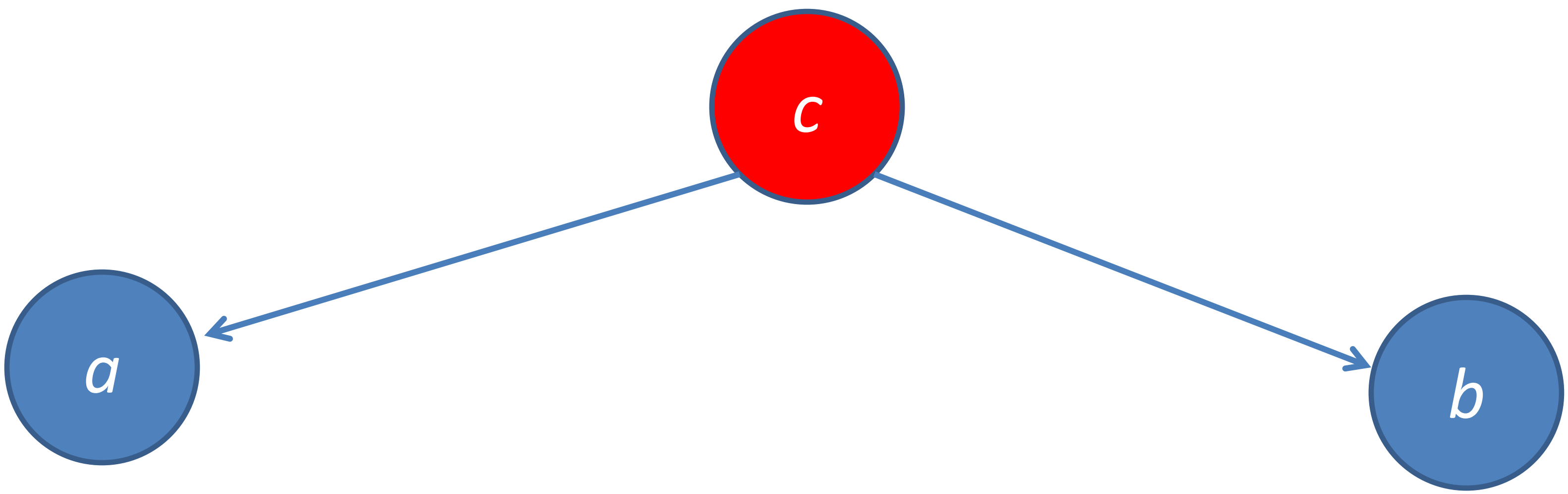


Finding independences

- The more *in*dependence relations we can find, the faster the inference → Test to find independences?



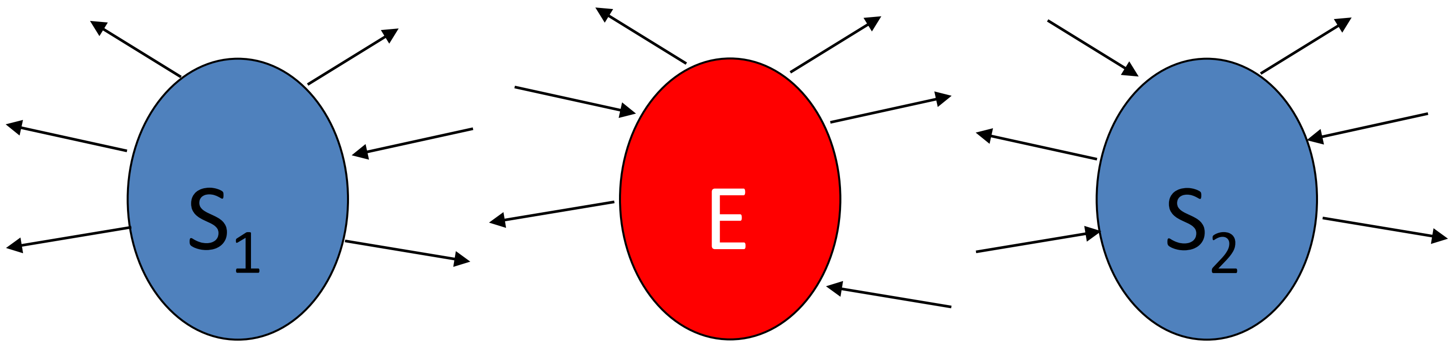
$p(b|c,a) = P(b|c) \qquad a \perp b \mid c$



$p(a,b|c) = P(a|c)\,P(b|c) \qquad a \perp b \mid c$

More General

- How can we find if S_1 and S_2 are conditionally independent given E?

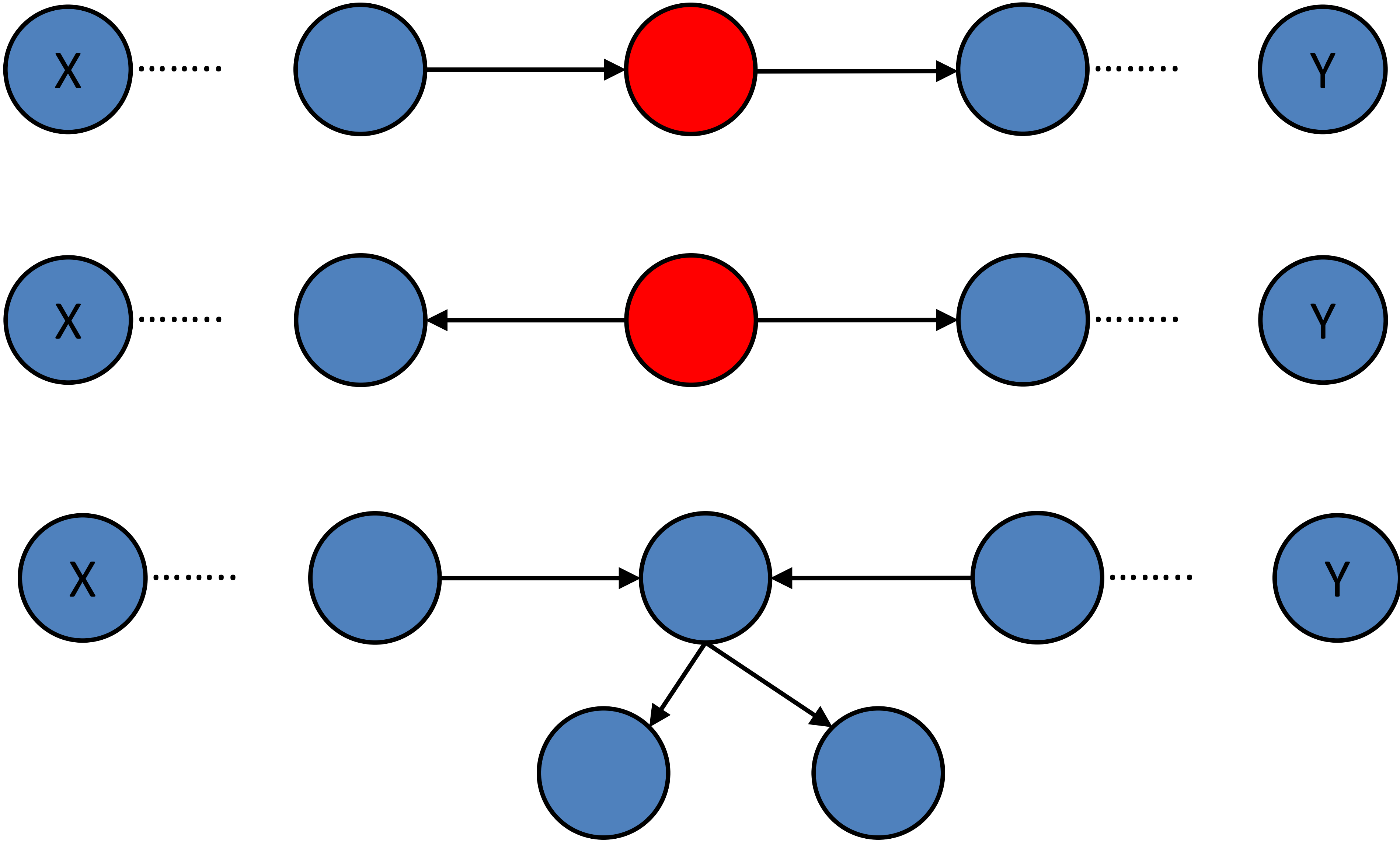


P (assignments to S_1 | E and assignments to S_2) = P (assignments to S_1 | E)

- Why is it important and useful?

We can simplify any computation that contains something like $P(S_1 \mid E , S_2)$ by $P(S_1 \mid E)$

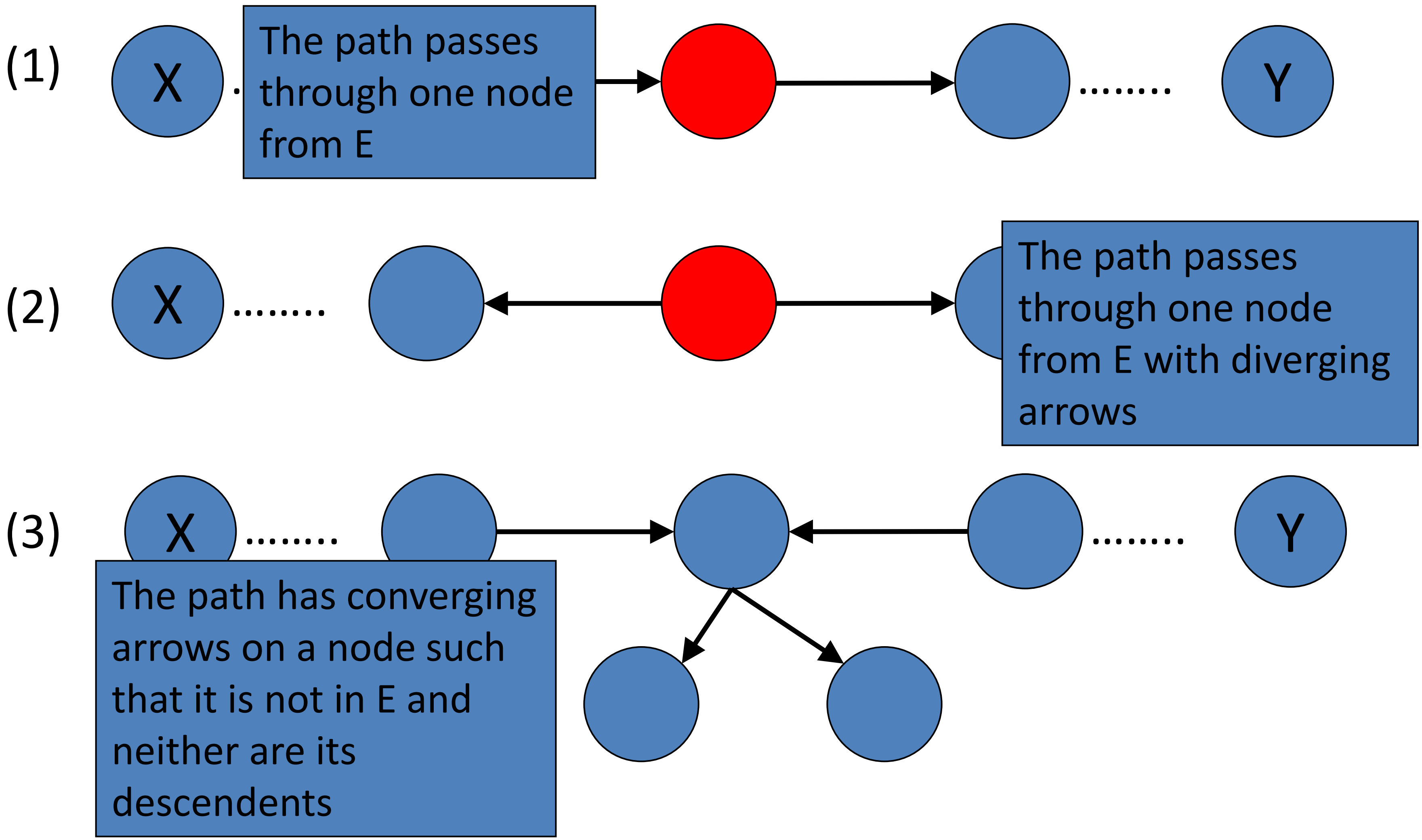
Intuitively E stands in between or "blocks" S_1 from S_2



X
Y
X
Y
X
Y

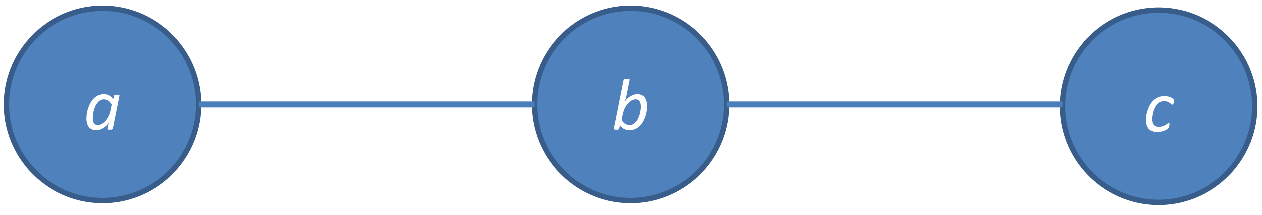
Blockage: Formal Definition

- A path from a node X to a node Y is *blocked* by a set E if either:

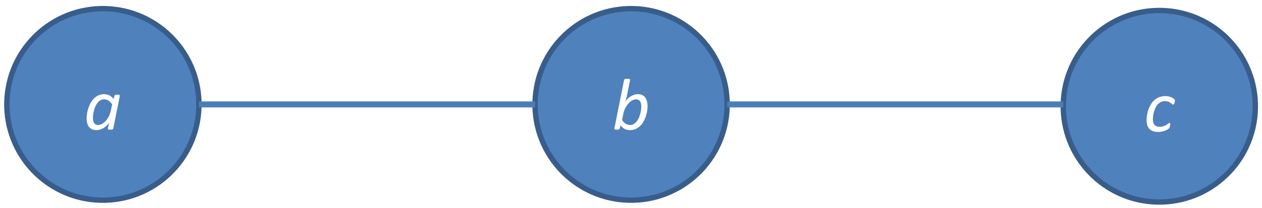


General case: Undirected

$$P(a, b) = \varphi(a, b) \qquad P(a, b) = \varphi_1(a)\varphi_2(b)$$

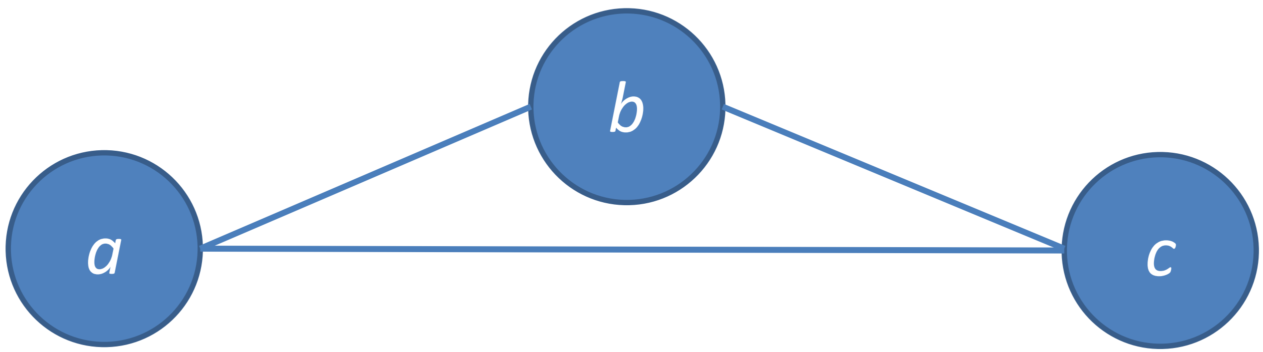


$$P(a,b,c) = \varphi_1(a,b)\,\varphi_2(b,c)$$



$$P(a, b, c) = \varphi_1(a, b)\,\varphi_2(b, c)$$

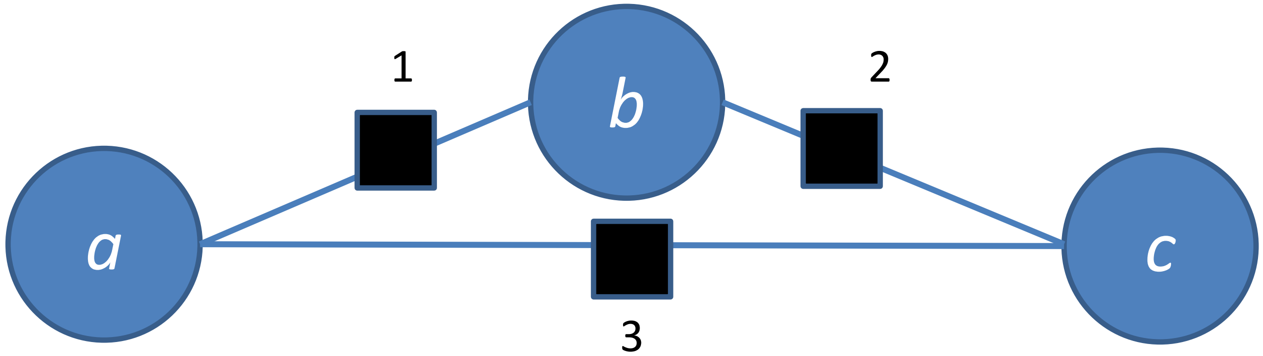
$a \perp c \mid b$ because all paths between a and c go through b



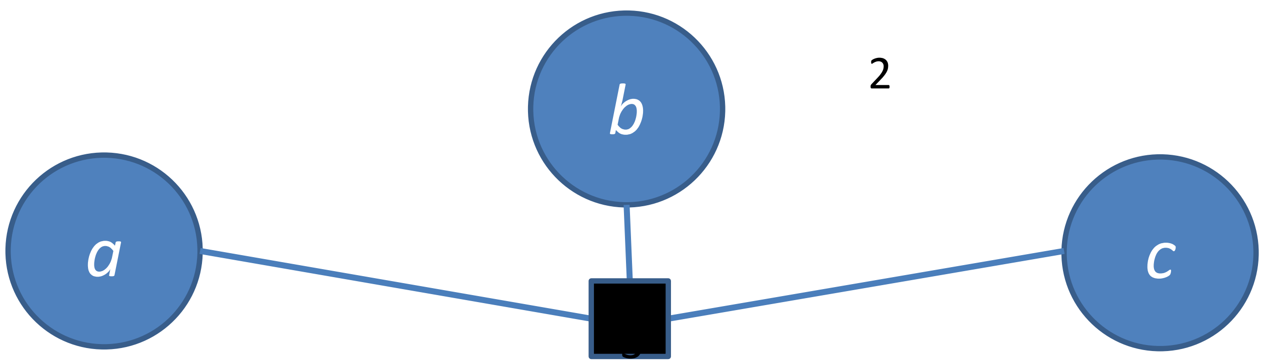
$$P(a,b,c) = \varphi_1(a,b)\,\varphi_2(b,c)\,\varphi_3(a,c)$$

$$P(a,b,c) = \varphi\,(a,b,c)$$

Factor graphs

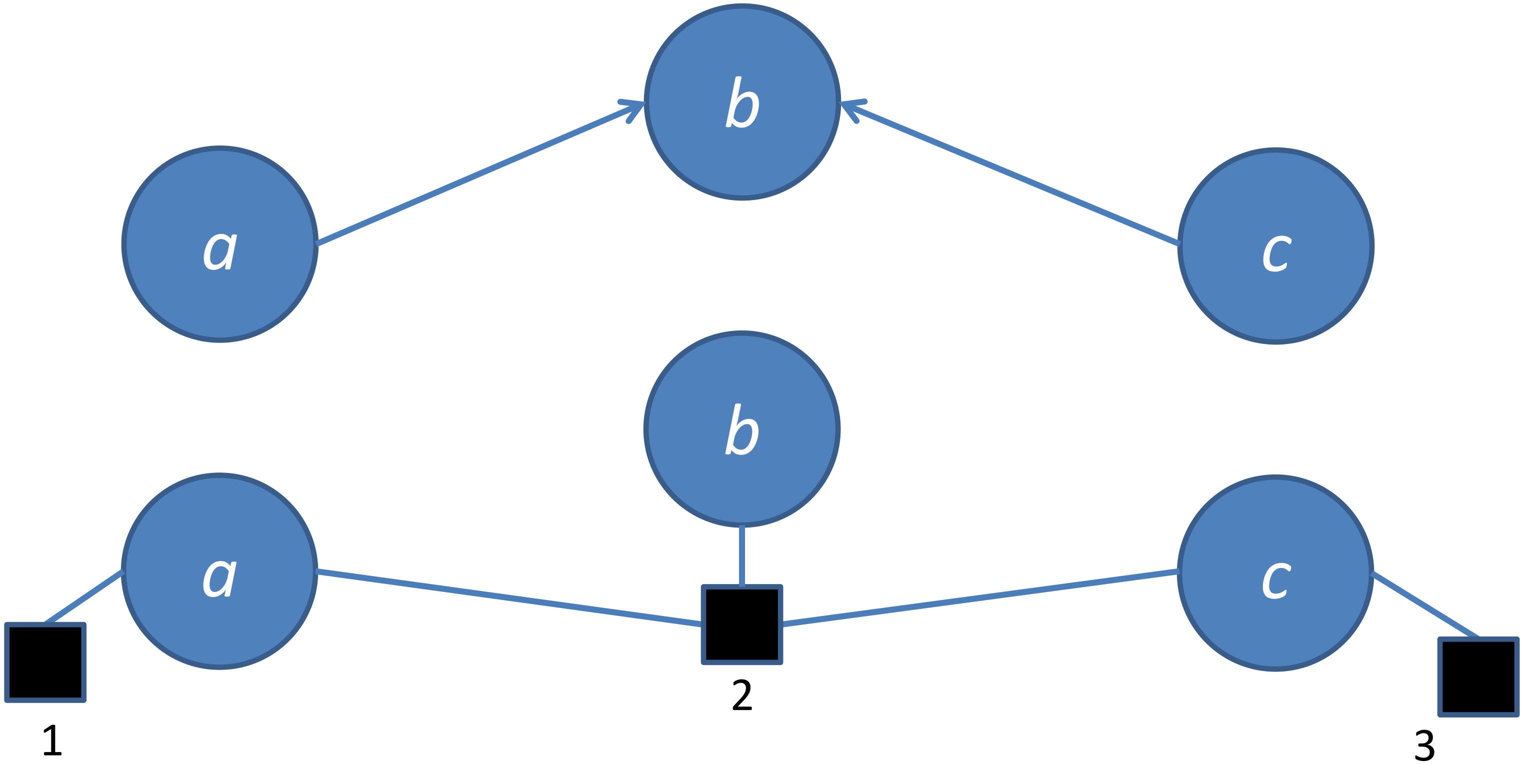


$$P(a,b,c) = \varphi_1(a,b)\,\varphi_2(b,c)\,\varphi_3(a,c)$$



$$P(a,b,c) = \varphi(a,b,c)$$

Directed vs. undirected



$$P(a,b,c) = \varphi_1(a,b)\,\varphi_2(b,c)\,\varphi_3(a,c)$$

$$\varphi_1(a) = P(a) \quad \varphi_3(c) = P(c)$$

$$\varphi_2(a,b,c) = P(b|a,c)$$

- N regions
- M possible labels
- Somehow, there is a way to estimate how likely a label is given image features $P(l_i|f)$
- We want to find the assignment of labels that optimizes $P(l_1, .., l_N|f)$

A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora and S. Belongie. Objects in Context. ICCV 2007

- Everything is independent:

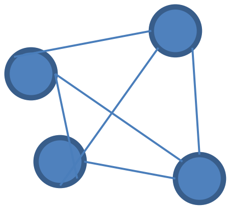
l_1 .... l_N

$$P(l_1, .., l_N | f) = \prod P(l_i | f)$$

Gives really stupid results because it does not take into account the distribution of likely relative occurrence of the labels

A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora and S. Belongie. Objects in Context. ICCV 2007

- Everything is dependent:

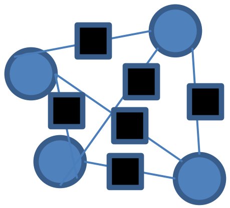


$$P(l_1,..,l_N|f)\alpha \prod P(f|l_i)P(l_1,..,l_N)$$

Hard to learn or represent $P(l_1,..,l_N)$

A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora and S. Belongie. Objects in Context. ICCV 2007

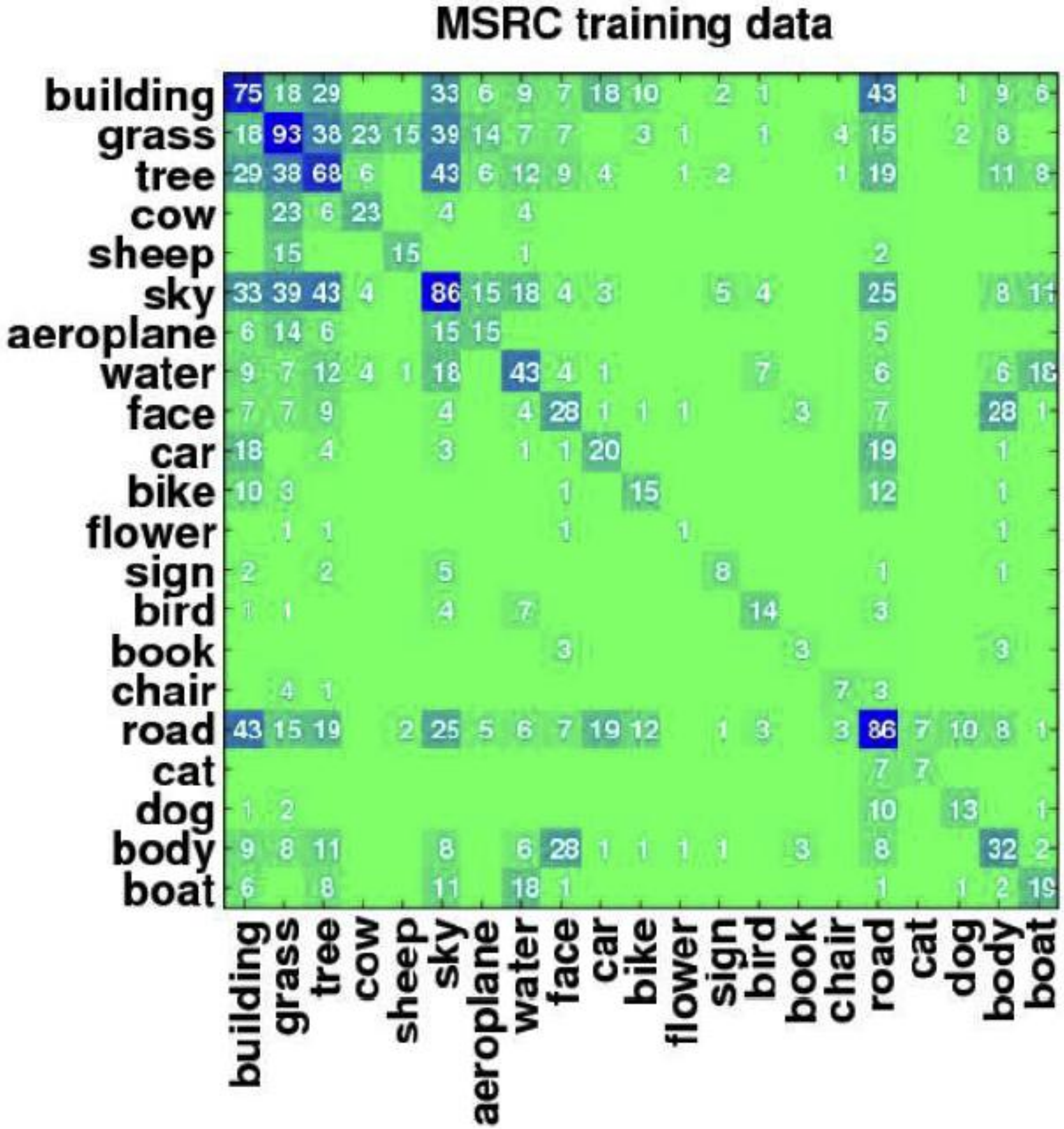
- Factor pairwise dependencies:



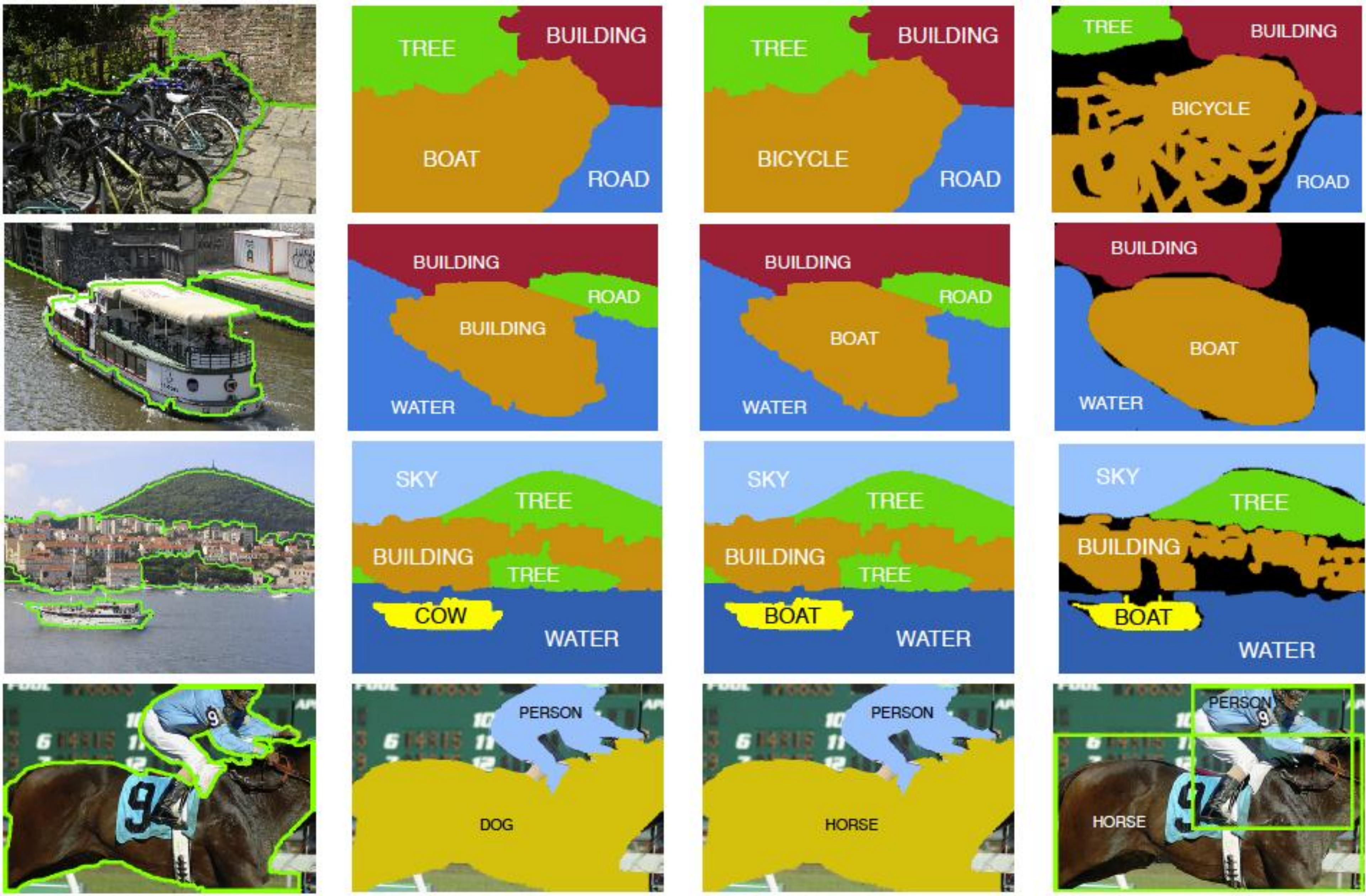
$$P(l_1, .., l_N) = \prod \varphi(l_i, l_j)$$

$\varphi(l_i, l_j)$ can be estimated from co-occurrence statistics from training data

A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora and S. Belongie. Objects in Context. ICCV 2007

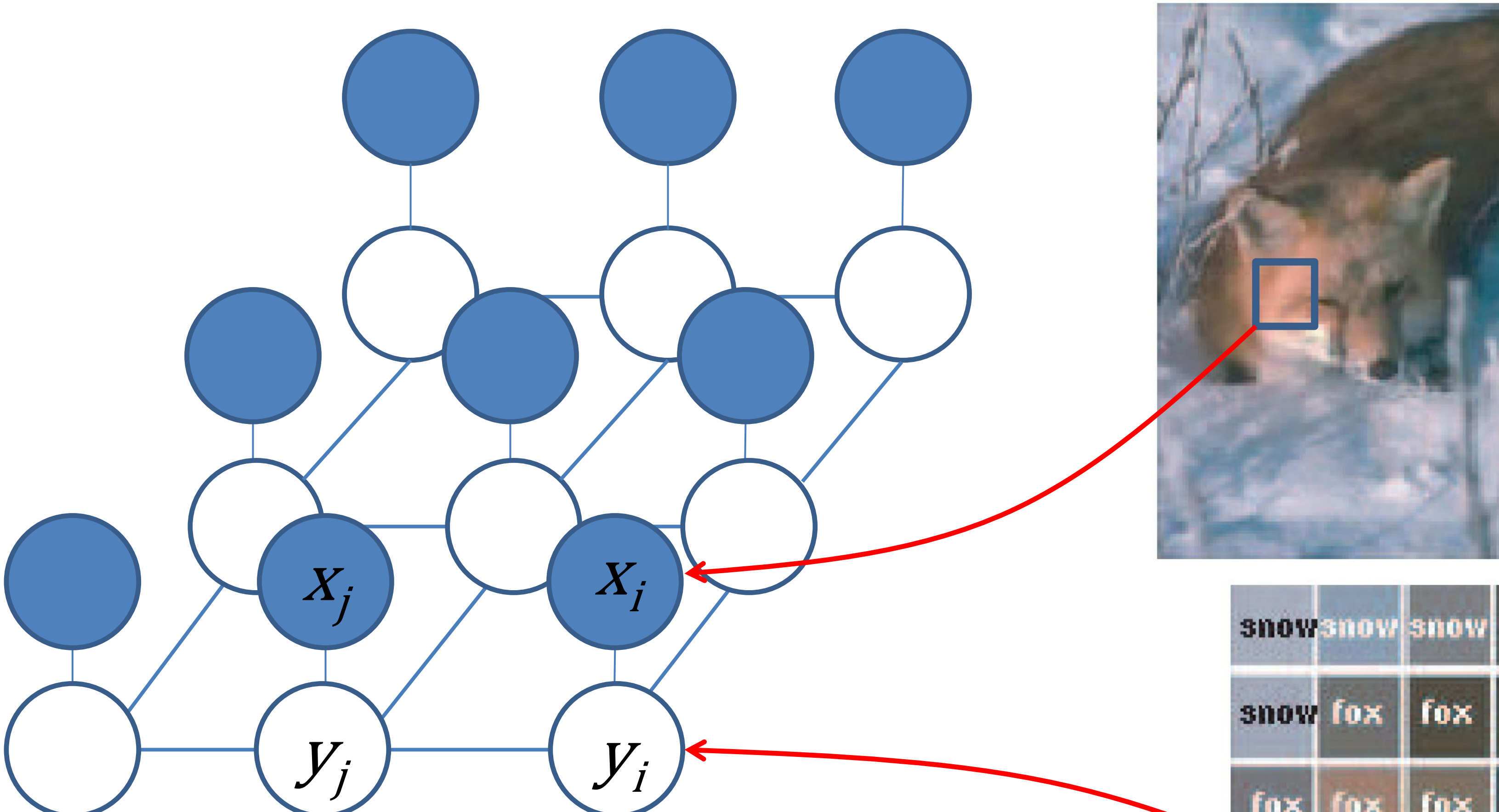


A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora and S. Belongie. Objects in Context. ICCV 2007



A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora and S. Belongie. Objects in Context. ICCV 2007

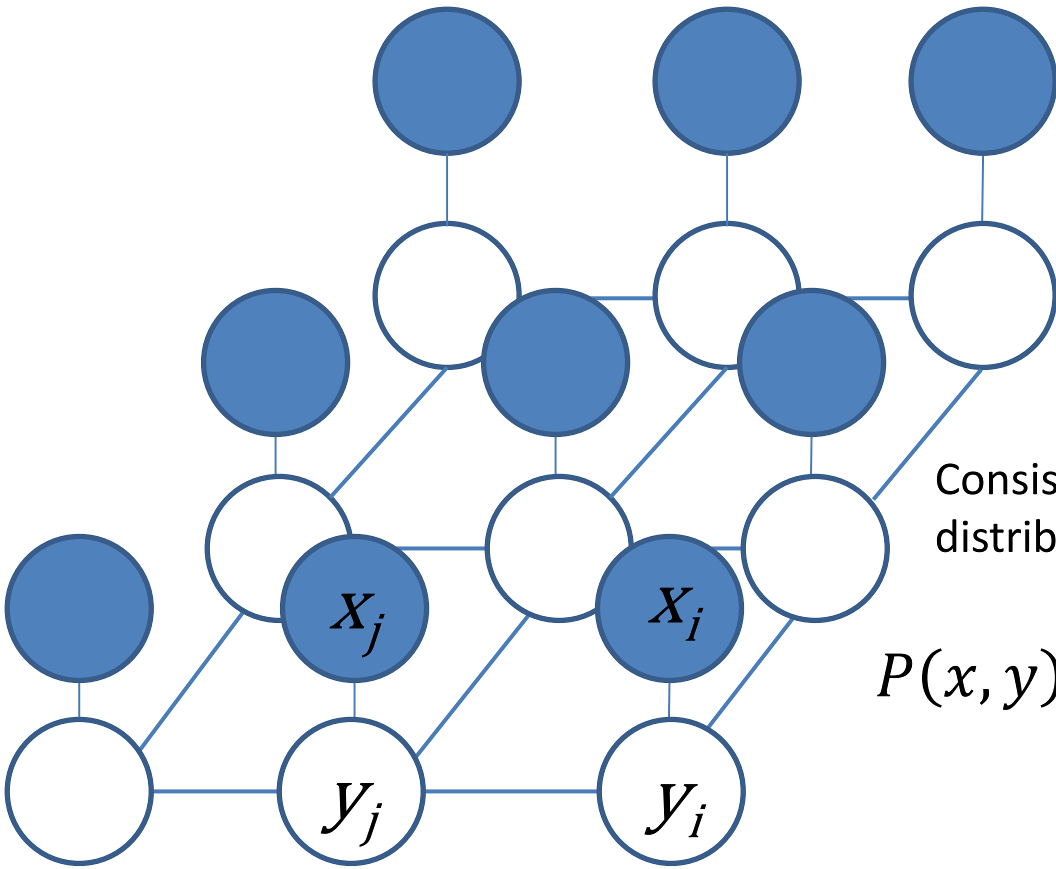
Example: MRF for image labeling



- y = class label (e.g., road, car, etc...)
- x = image data at local patch

Example from Carbonetto, de Freitas & Barnard, ECCV'04

Example: MRF for image labeling



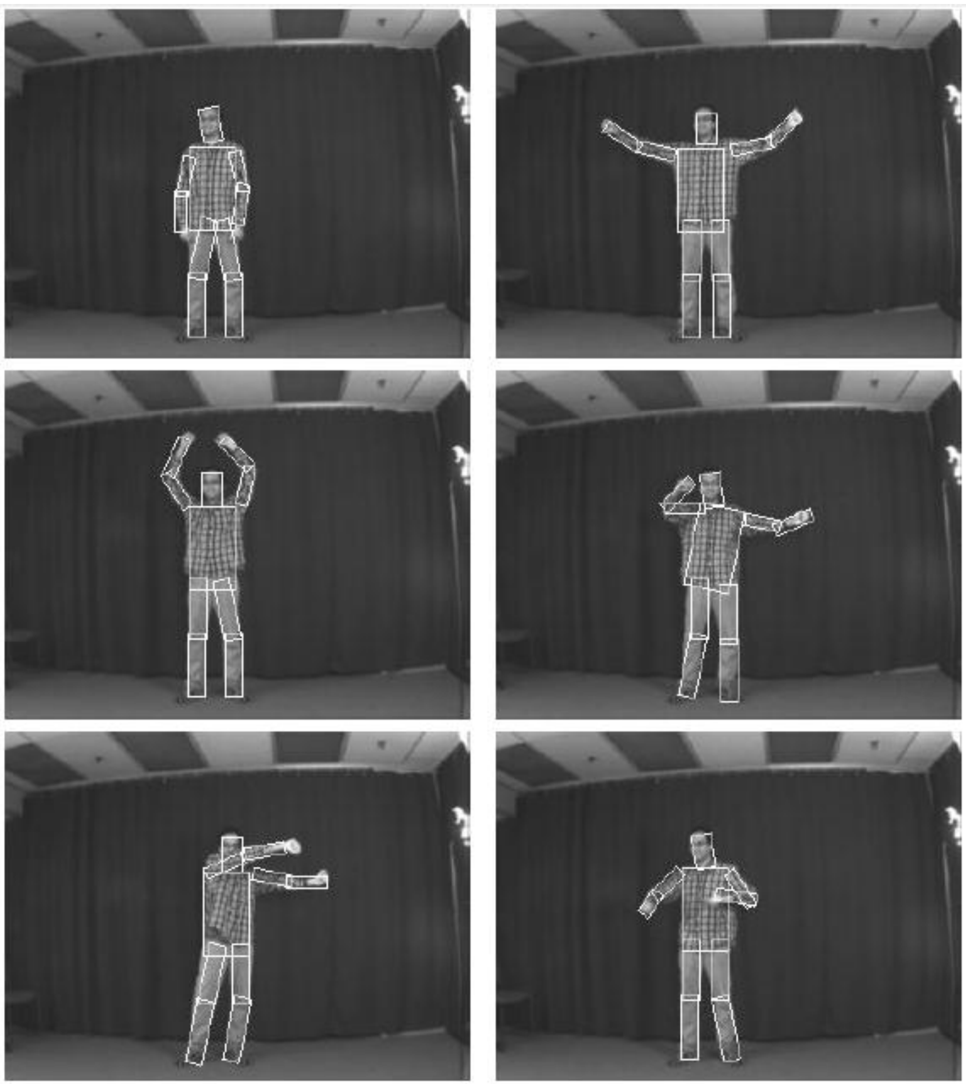
Consistency on spatial distribution of labels

Agreement of label vs. input data

$$P(x,y)\ \alpha \prod_{i,j} \varphi_{ij}(y_i, y_j) \prod_i \varphi_i(x_i, y_i)$$

- y = class label (e.g., road, car, etc…)
- x = image data at local patch

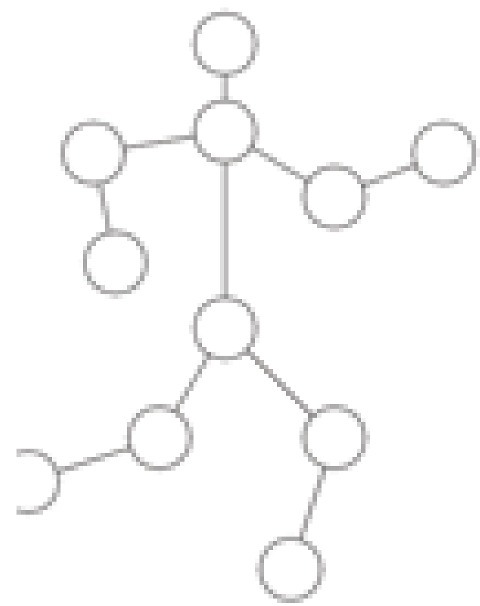
Example: Inferring human poses



Example from Felzenszwalb'04

x_i = Input image data at limb i

y_i = Pose (location and orientation) of limb i



- Note: Efficient because tree-structured