

Today

- Reviewing “Marginalization”
- Reviewing “Junction Tree”
- ML Estimation in Graphical Models
- ML Estimation in decomposable graphs
- ML Estimation in arbitrary graphs
- ML Estimation with hidden variables (EM algorithm)

Marginalization: kids

Example: you have 3 kids. What is the probability that the first kid is a boy?

$$P(1st = B) = \sum_{2nd} \sum_{3rd} P(1st = B, 2nd, 3rd)$$

$$P(1st = B) = \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} = \frac{1}{2}$$

1st	2nd	3rd	P
G	G	G	1/8
G	G	B	1/8
G	B	G	1/8
G	B	B	1/8
B	G	G	1/8
B	G	B	1/8
B	B	G	1/8
B	B	B	1/8

Marginalization: Tennis

What's the probability that I win against each?

$$P(F = W) = \sum_{Surya} \sum_{Antonio} P(Surya, Antonio, Federer = W)$$

$$P(F = W) = 0.0 + 0.0 + 0.0 + 0.0 = 0.00$$

$$P(S = W) = \sum_{Antonio} \sum_{Federer} P(Surya = W, Antonio, Federer)$$

$$P(S = W) = 0.0 + 0.2 + 0.0 + 0.2 = 0.4$$

$$P(F = W) = \sum_{Surya} \sum_{Federer} P(Surya, Antonio = W)$$

$$P(A = W) = 0.0 + 0.2 + 0.0 + 0.2 = 0.4$$

Surya	Antonio	Federer	P
W	W	W	0.00
W	W	L	0.2
W	L	W	0.00
W	L	L	0.2
L	W	W	0.00
L	W	L	0.2
L	L	W	0.00
L	L	L	0.4

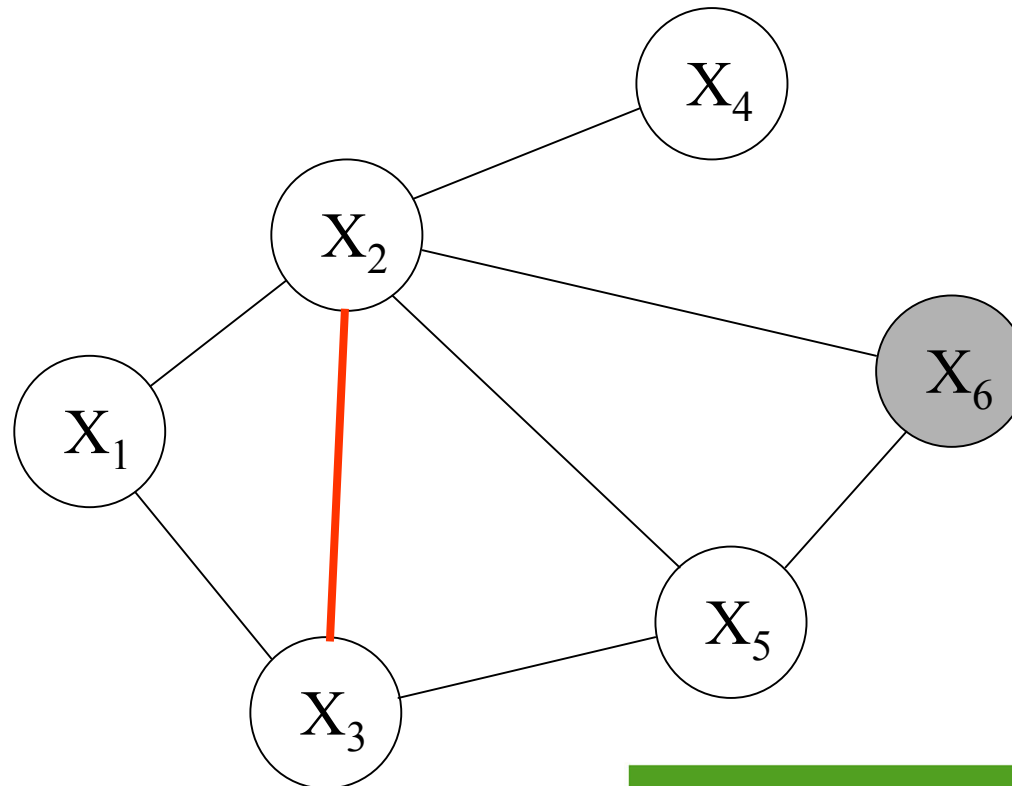
The Junction Tree Algorithm

The **Junction Tree algorithm** provides the definitive answer on how to perform exact inference in Graphical Models

It does not repeat computations

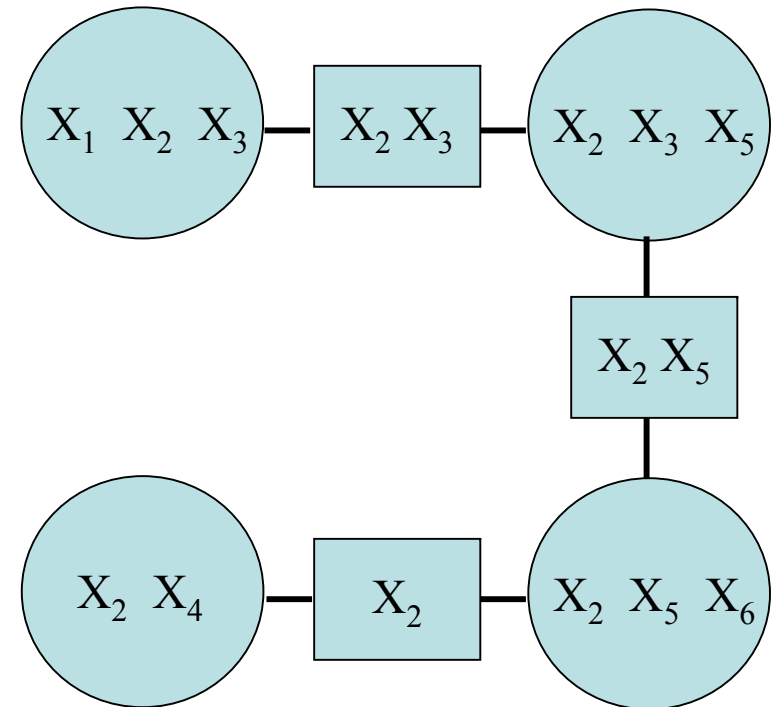
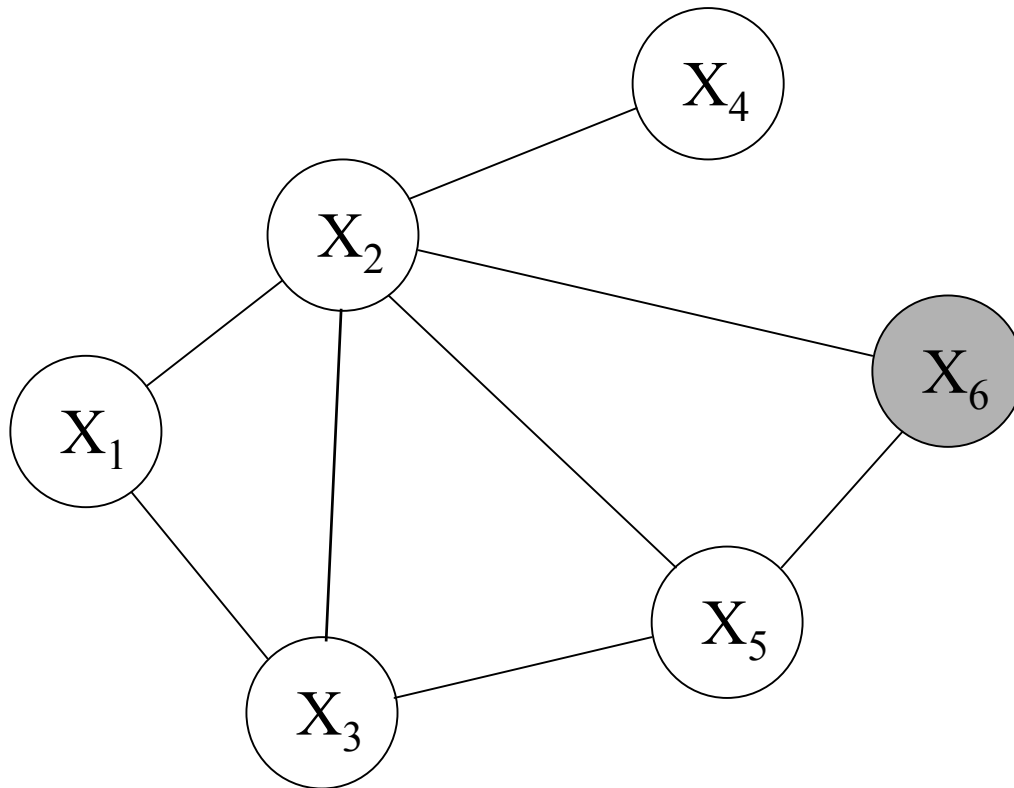
The Junction Tree Algorithm

(1) *Triangulate* the graph (if it's not triangulated)



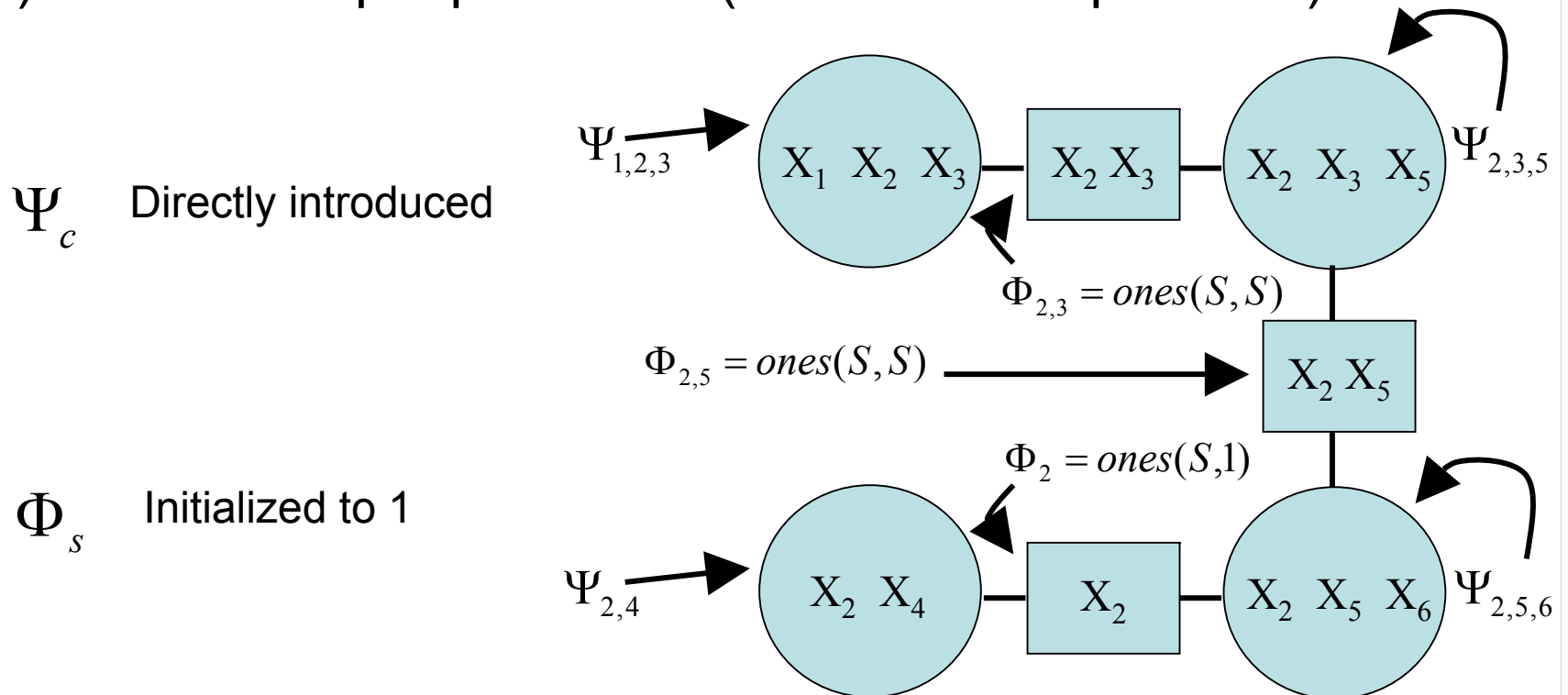
The Junction Tree Algorithm

(2) Create a Junction Tree



The Junction Tree Algorithm

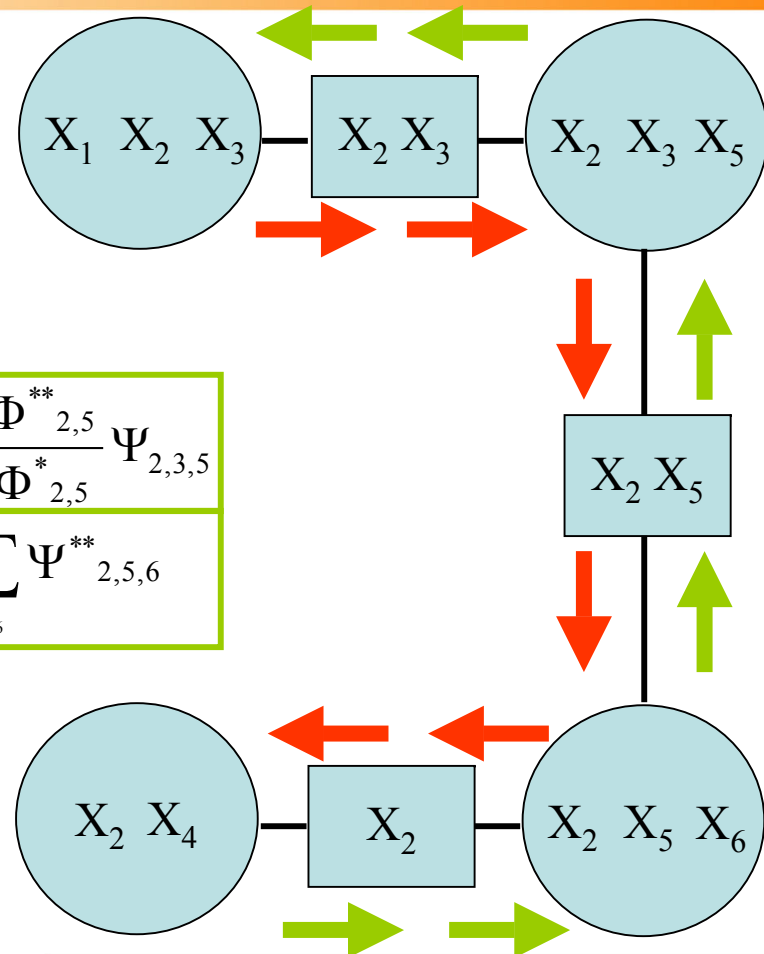
(3) Initialize clique potentials (nodes and separators)



The Junction Tree Algorithm

(4) Message passing

$\Psi^{**}_{1,2,3} = \frac{\Phi^{**}_{2,3}}{\Phi^*_{2,3}} \Psi_{1,2,3}$	$\Phi^{**}_{2,3} = \sum_{x_5} \Psi^{**}_{2,3,5}$
$\Phi^*_{2,3} = \sum_{x_1} \Psi_{1,2,3}$	$\Psi^*_{2,3,5} = \frac{\Phi^*_{2,3}}{\Phi_{2,3}} \Psi_{2,3,5}$
	$\Phi^*_{2,5} = \sum_{x_3} \Psi^*_{2,3,5}$
	$\Psi^{**}_{2,3,5} = \frac{\Phi^{**}_{2,5}}{\Phi^*_{2,5}} \Psi_{2,3,5}$
	$\Psi^*_{2,5,6} = \frac{\Phi^*_{2,5}}{\Phi_{2,5}} \Psi_{2,5,6}$
	$\Phi^{**}_{2,5} = \sum_{x_6} \Psi^{**}_{2,5,6}$
$\Psi^*_{2,4} = \frac{\Phi^*_2}{\Phi_2} \Psi_{2,4}$	$\Phi^*_2 = \sum_{x_5, x_6} \Psi^*_{2,5,6}$
$\Phi^{**}_2 = \sum_{x_4} \Psi^*_{2,4}$	$\Psi^*_{2,5,6} = \frac{\Phi^{**}_2}{\Phi^*_2} \Psi_{2,5,6}$



The Junction Tree Algorithm

Once the algorithm has finished:

potential in each clique node is equal to the marginal in that node

Marginals for singletons can then be computed by brute force

ML Estimation

So far we have seen how to perform **INFERENCE** in Graphical Models, i.e. compute **marginal** and **conditional** probabilities given the potential functions

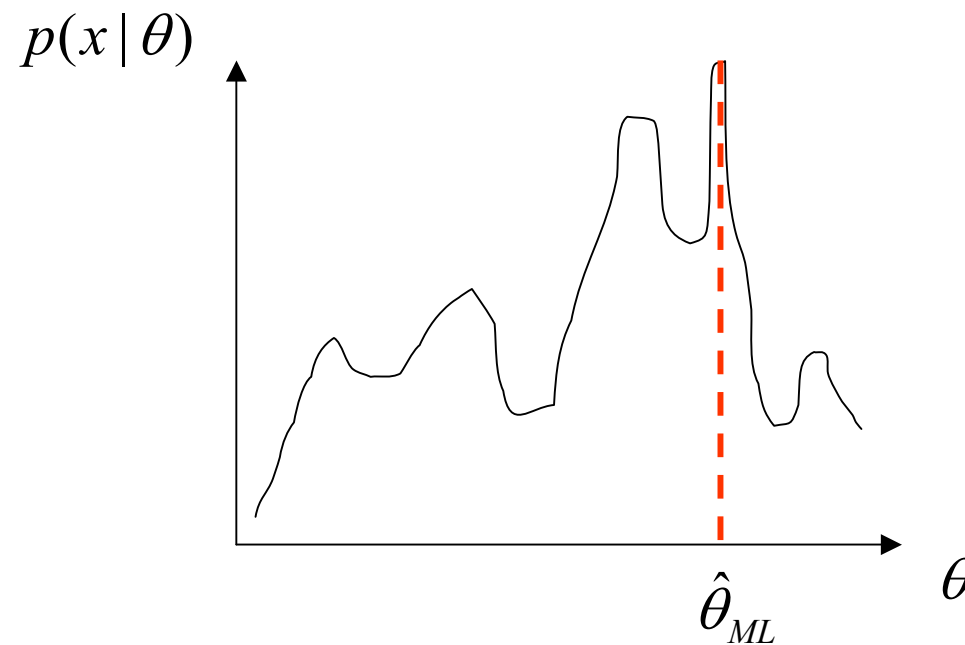
But, how can we **learn** the potential functions from data?

This is the problem of **Estimation** or **Learning** in Graphical Models

ML Estimation

Maximum-likelihood estimation:

Given data X , what is the value of θ that maximizes $p(x|\theta)$?



ML Estimation

θ_c is the part of Ψ_c that needs to be estimated

$$\Psi_c(x_c) = \Psi_c(x_c; \theta_c)$$

Example:

$$\Psi_c(x_c; \theta_c) = \frac{1}{\sqrt{(2\pi)^n \det(\Sigma_c)}} \exp\left(-\frac{1}{2}(x_c - \mu_c)^T \Sigma_c^{-1} (x_c - \mu_c)\right)$$

$$\theta_c = (\mu_c, \Sigma_c)$$

ML Estimation

Solution:

Write down the whole dataset:

$$D = (x_{V,1}, x_{V,2}, \dots, x_{V,N})$$

Where V demotes the index for all the nodes and N is the number or samples

ML Estimation

Solution:

The joint probability is written as

$$p(x_V | \theta) = \frac{1}{Z} \prod_C \Psi_C(x_C)$$

x_V

Marginal

Total # observations

$$m(x_V) = \sum_n \delta(x_V, x_{V,n})$$

$$m(x_C) = \sum_{x_{VC}} m(x_V)$$

$$N = \sum_{x_V} m(x_V)$$

ML Estimation

Solution:

The log-likelihood for the n^{th} sample is

$$p(x_{V,n} | \theta) = \frac{1}{Z} \prod_{x_V} p(x_V | \theta)^{\delta(x_{V,n}, x_V)}$$

The probability of all the observed data is then:

$$p(D | \theta) = \prod_n \prod_{x_V} p(x_V | \theta)^{\delta(x_{V,n}, x_V)}$$

ML Estimation

Solution:

Taking logarithms, one obtains that the log-likelihood is:

$$l(\theta; D) = \sum_C \sum_{x_C} m(x_C) \log \Psi_C(x_C) - N \log Z$$

Our task then is to maximize this with respect to $\Psi_C(x_C)$

ML Estimation

Solution:

Take derivatives and obtain:

$$\hat{p}_{ML}(x_C) = \frac{1}{N} m(x_C)$$

Which is exactly the empirical mean $\tilde{p}(x_C)$. So,

$$\hat{p}_{ML}(x_C) = \tilde{p}(x_C)$$

ML Estimation

First – How to learn the potential functions when we have **observed data for all variables** in the model?

Second – How to learn the potential functions when **there are latent (hidden) variables**, i.e., we do not observe data for them?

Completely Observed GMs

Completely observed Graphical Models: