

# Lecture 20. PGM Representation

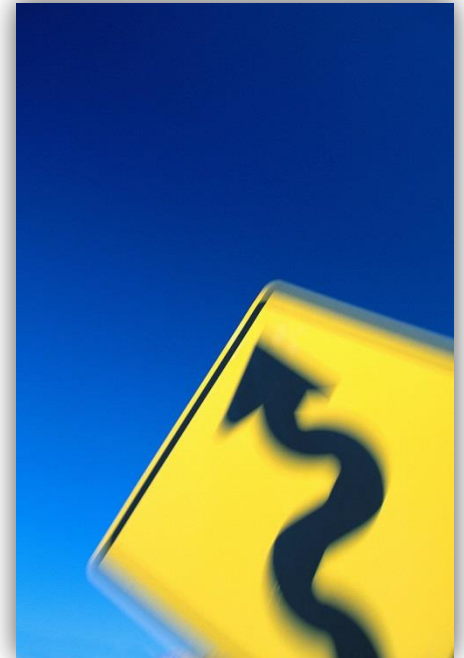
COMP90051 Statistical Machine Learning

Lecturer: Ben Rubinstein



# Next Lectures

- Representation of joint distributions
- Conditional/marginal independence
  - \* Directed vs undirected
- Probabilistic inference
  - \* Computing other distributions from joint
- Statistical inference
  - \* Learn parameters from (missing) data
- Examples



# This lecture

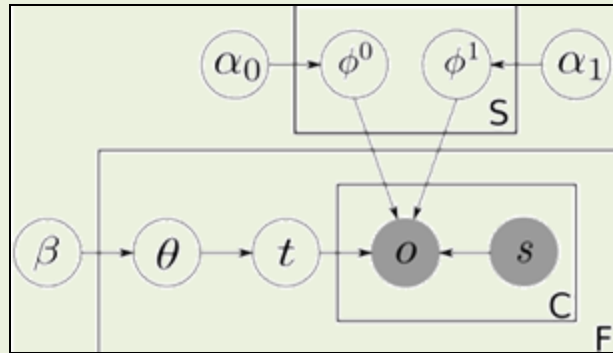
- (Directed) probabilistic graphical models
  - \* Motivations: applications, unifies algorithms
  - \* Motivation: ideal tool for Bayesians
  - \* Independence lowers computational/model complexity
    - Conditional independence
  - \* PGMs: compact representation of factorised joints
- Undirected PGMs and conversion from D-PGMs
- Example PGMs, applications

# Probabilistic Graphical Models

*Marriage of graph theory and probability theory.  
Tool of choice for Bayesian statistical learning.*

*We'll stick with easier discrete case,  
ideas generalise to continuous.*

# Motivation by practical importance



- **Many applications**

- \* Phylogenetic trees
- \* Pedigrees, Linkage analysis
- \* Error-control codes
- \* Speech recognition
- \* Document topic models
- \* Probabilistic parsing
- \* Image segmentation
- \* ...

- **discovered algorithms**

- \* HMMs
- \* Kalman filters
- \* Mixture models
- \* LDA
- \* MRFs
- \* CRF
- \* Logistic, linear regression
- \* ...

# Motivation by way of comparison

## Bayesian statistical learning

- Model joint distribution of  $X$ 's,  $Y$  and parameter r.v.'s
  - \* “Priors”: marginals on parameters
- Training: update prior to posterior using observed data
- Prediction: output posterior, or some function of it (MAP)

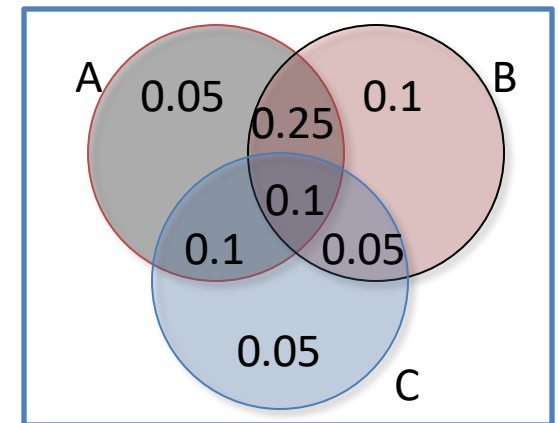
## PGMs aka “Bayes Nets”

- Efficient joint representation
  - \* Independence made explicit
  - \* Trade-off between expressiveness and need for data, easy to make
  - \* Easy for practitioners to model
- Algorithms to fit parameters, compute marginals, posterior

# Everything Starts at the Joint Distribution

- All joint distributions on discrete r.v.'s can be represented as tables
- #rows grows exponentially with #r.v.'s
- Example: Truth Tables
  - \*  $M$  Boolean r.v.'s require  $2^M - 1$  rows
  - \* Table assigns probability per row

A	B	C	Prob
0	0	0	0.30
0	0	1	0.05
0	1	0	0.10
0	1	1	0.05
1	0	0	0.05
1	0	1	0.10
1	1	0	0.25
1	1	1	?



# The Good: What we can do with the joint

- **Probabilistic inference** from joint on r.v.'s
  - \* Computing any other distributions involving our r.v.'s
- Pattern: want a distribution, have joint; use:  
**Bayes rule** + **marginalisation**
- Example: **naïve Bayes classifier**
  - \* Predict class  $y$  of instance  $\mathbf{x}$  by maximising

$$\Pr(Y = y | \mathbf{X} = \mathbf{x}) = \frac{\Pr(Y=y, \mathbf{X}=\mathbf{x})}{\Pr(\mathbf{X}=\mathbf{x})} = \frac{\Pr(Y=y, \mathbf{X}=\mathbf{x})}{\sum_y \Pr(\mathbf{X}=\mathbf{x}, Y=y)}$$

Recall: *integration (over parameters)* continuous equivalent of sum (both referred to as marginalisation)



# The Bad & Ugly: Tables *waaaaay* too large!!

- **The Bad:** Computational complexity
  - \* Tables have exponential number of rows in number of r.v.'s
  - \* Therefore → poor space & time to marginalise
- **The Ugly:** Model complexity
  - \* Way too flexible
  - \* Way too many parameters to fit  
→ need lots of data OR will overfit
- Antidote: assume independence!

A	B	C	Prob
0	0	0	0.30
0	0	1	0.05
0	1	0	0.10
0	1	1	0.05
1	0	0	0.05
1	0	1	0.10
1	1	0	0.25
1	1	1	?

# Example: You're late!

- Modeling a tardy lecturer. Boolean r.v.'s
  - \*  $T$ : Ben teaches the class
  - \*  $S$ : It is sunny (o.w. bad weather)
  - \*  $L$ : The lecturer arrives late (o.w. on time)
- Assume: Ben sometimes delayed by bad weather, Ben more likely late than other lecturers
  - \*  $\Pr(S|T) = \Pr(S)$ ,  $\Pr(S) = 0.3$   $\Pr(T) = 0.6$
- Lateness not independent on weather, lecturer
  - \* Need  $\Pr(L|T=t, S=s)$  for all combinations
- Need just 6 parameters



		T	
		False	True
S	False	0.1	0.2
	True	0.05	0.1

$$\text{Joint: } \Pr(T, S, L) = \Pr(L|T, S) \Pr(T|S) \Pr(S) \Pr(T)$$

# Independence: not a dirty word

Lazy Lecturer Model	Model details	# params
Our model with $S, T$ independence	$\Pr(S, T)$ factors to $\Pr(S) \Pr(T)$	2
	$\Pr(L T, S)$ modelled in full	4
Assumption-free model	$\Pr(L, T, S)$ modelled in full	7

- Independence assumptions
  - \* Can be reasonable in light of domain expertise
  - \* Allow us to factor  $\rightarrow$  Key to tractable models

# Factoring Joint Distributions

- **Chain Rule:** for any ordering of r.v.'s can always factor:

$$\Pr(X_1, X_2, \dots, X_k) = \prod_{i=1}^k \Pr(X_i | X_{i+1}, \dots, X_k)$$

- Model's independence assumptions correspond to
  - Dropping conditioning r.v.'s in the factors!
  - Example **unconditional indep.**:  $\Pr(X_1 | X_2) = \Pr(X_1)$
  - Example **conditional indep.**:  $\Pr(X_1 | X_2, X_3) = \Pr(X_1 | X_2)$
- Example: independent r.v.'s  $\Pr(X_1, \dots, X_k) = \prod_{i=1}^k \Pr(X_i)$
- Simpler factors: **speed up inference** and **avoid overfitting**

# Mini Summary

- Joint distributions
- Probabilistic inference: Bayes rule & marginalisation
- Direct representation of joints
  - \* Probabilistic inference: Computationally costly
  - \* Statistical inference: Requires more data
- Factoring joints and conditional independence

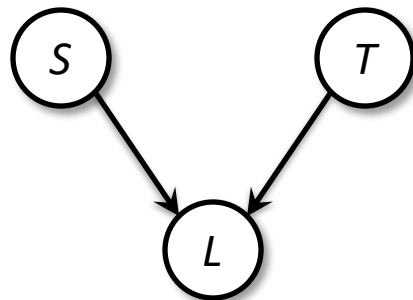
Next: Directed probabilistic graphical models

# Directed PGM

- Nodes
- Edges (acyclic)
- Random variables
- Conditional dependence
  - \* **Node table:**  $\Pr(\text{child}|\text{parents})$
  - \* Child directly depends on parents
- Joint factorisation

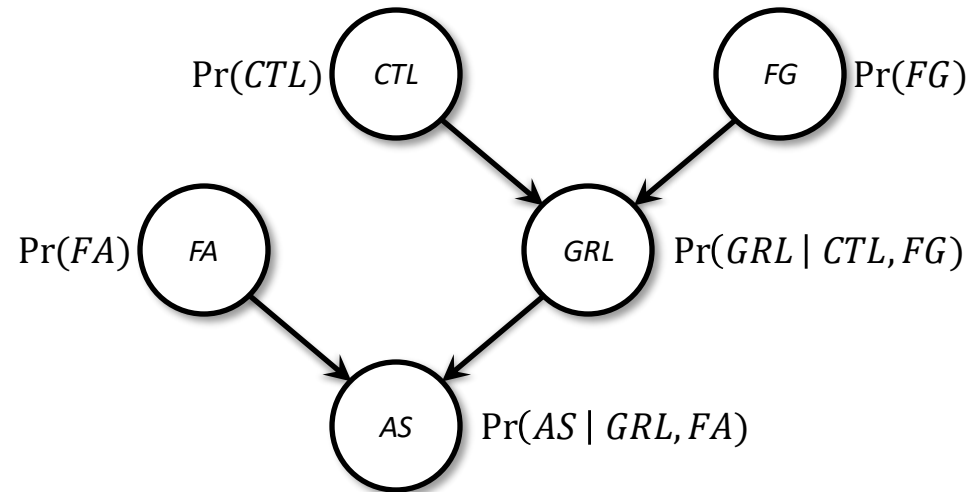
$$\Pr(X_1, X_2, \dots, X_k) = \prod_{i=1}^k \Pr(X_i | X_j \in \text{parents}(X_i))$$

## *Tardy Lecturer Example*

 $\Pr(S)$  $\Pr(T)$  $\Pr(L|S, T)$

# Example: Nuclear power plant

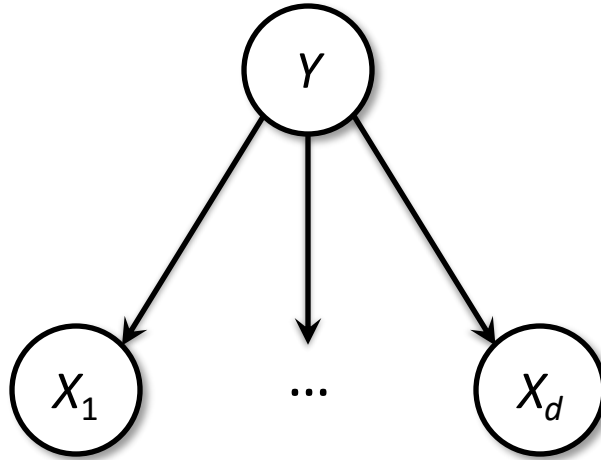
- Core temperature  
→ Temperature Gauge  
→ Alarm
- Model uncertainty in monitoring failure
  - \* GRL: gauge reads low
  - \* CTL: core temperature low
  - \* FG: faulty gauge
  - \* FA: faulty alarm
  - \* AS: alarm sounds
- PGMs to the rescue!



Joint  $\Pr(CTL, FG, FA, GRL, AS)$  given by

$$\Pr(AS|FA, GRL) \Pr(FA) \Pr(GRL|CTL, FG) \Pr(CTL) \Pr(FG)$$

# Naïve Bayes



$$Y \sim \text{Bernoulli}(\theta)$$

*Aside: Bernoulli is just Binomial with count=1*

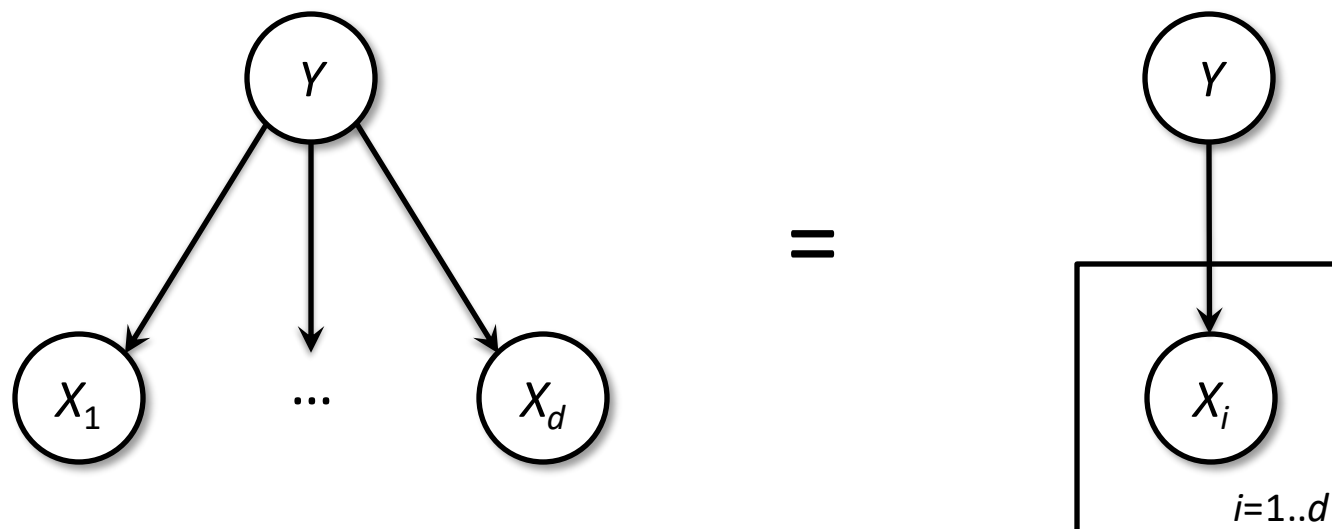
$$X_j|Y \sim \text{Bernoulli}(\theta_{j,Y})$$

$$\begin{aligned}\Pr(Y, X_1, \dots, X_d) &= \Pr(X_1, \dots, X_d, Y) \\ &= \Pr(X_1|Y) \Pr(X_2|X_1, Y) \dots \Pr(X_d|X_1, \dots, X_{d-1}, Y) \Pr(Y) \\ &= \Pr(X_1|Y) \Pr(X_2|Y) \dots \Pr(X_d|Y) \Pr(Y)\end{aligned}$$

Prediction: predict label maximising  $\Pr(Y|X_1, \dots, X_d)$

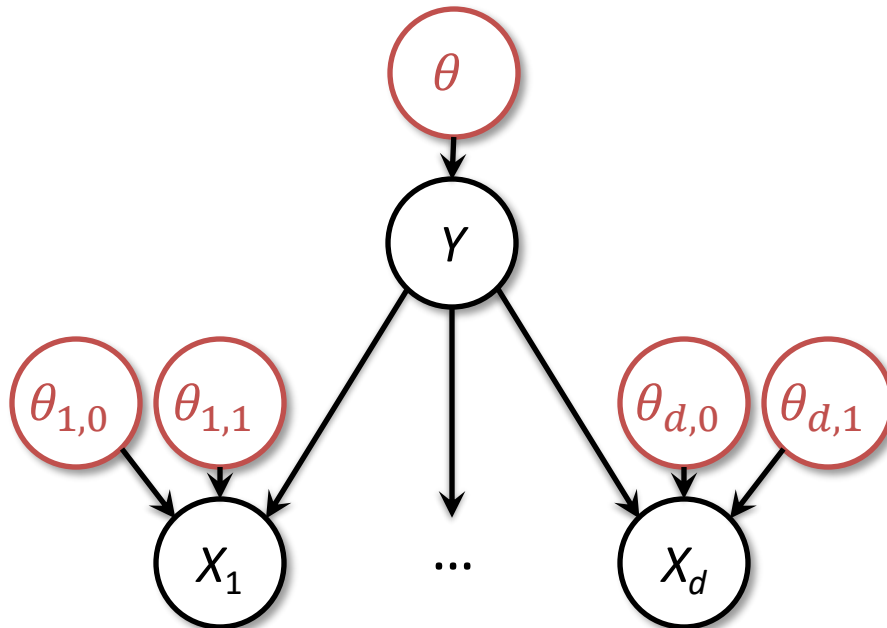


# Short-hand for repeats: Plate notation



# PGMs: frequentist OR Bayesian...

- PGMs represent joints, which are central to Bayes
- Catch is that Bayesians add: **node per parameters**, with table being the parameter's prior



$$Y \sim \text{Bernoulli}(\theta)$$

$$X_j|Y \sim \text{Bernoulli}(\theta_{j,Y})$$

$$\theta's \sim \text{Beta}$$

# Mini Summary

Directed probabilistic graphical models (D-PGMs)

- Definition as graph and conditionals
- Definition as joint distribution factorisation
- Plate notation
- Bayesian D-PGMs

Next: Undirected probabilistic graphical models

# Undirected PGMs

*Undirected variant of PGM, parameterised by arbitrary positive valued functions of the variables, and global normalisation.*

*A.k.a. Markov Random Field.*

# Undirected vs directed

## Undirected PGM

- Graph
  - \* Edges undirected
- Probability
  - \* Each node a r.v.
  - \* Each clique  $C$  has “factor”  
 $\psi_C(X_j: j \in C) \geq 0$
  - \* Joint  $\propto$  product of factors

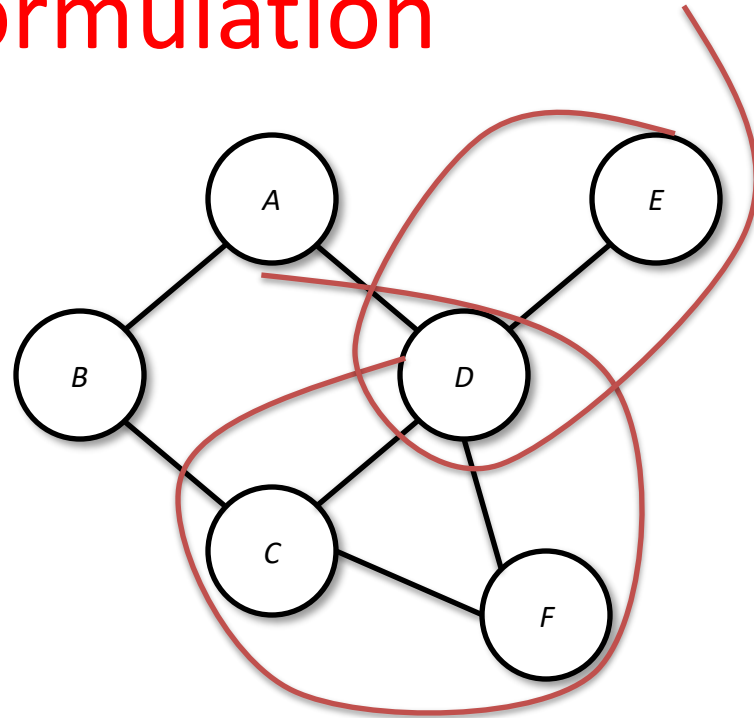
## Directed PGM

- Graph
  - \* Edges directed
- Probability
  - \* Each node a r.v.
  - \* Each node has conditional  
 $p(X_i | X_j \in \text{parents}(X_i))$
  - \* Joint = product of cond'ls

**Key difference = normalisation**

# Undirected PGM formulation

- Based on notion of
  - \* **Clique**: a set of fully connected nodes (e.g., A-D, C-D, C-D-F)
  - \* **Maximal clique**: largest cliques in graph (not C-D, due to C-D-F)
- Joint probability defined as



$$P(a, b, c, d, e, f) = \frac{1}{Z} \psi_1(a, b) \psi_2(b, c) \psi_3(a, d) \psi_4(d, c, f) \psi_5(d, e)$$

- \* where each  $\psi$  is a positive function and Z is the normalising '**partition**' function

$$Z = \sum_{a, b, c, d, e, f} \psi_1(a, b) \psi_2(b, c) \psi_3(a, d) \psi_4(d, c, f) \psi_5(d, e)$$

# Directed to undirected

- Directed PGM formulated as

$$P(X_1, X_2, \dots, X_k) = \prod_{i=1}^k Pr(X_i | X_{\pi_i})$$

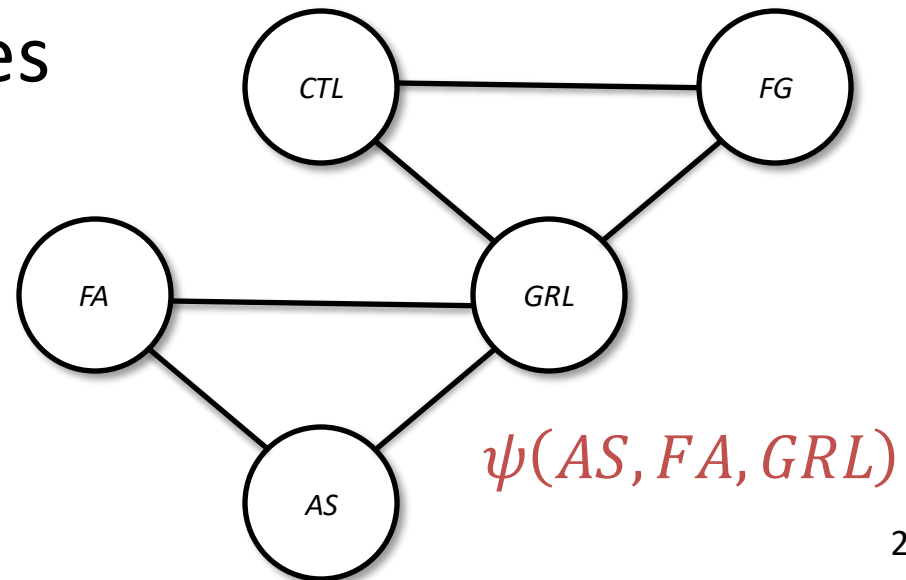
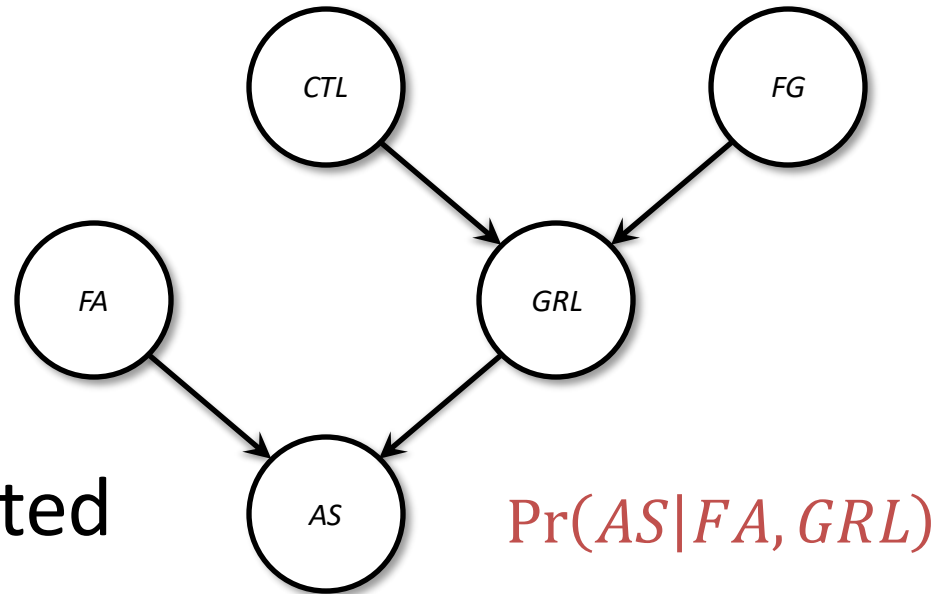
where  $\pi$  indexes parents.

- Equivalent to U-PGM with
  - \* each conditional probability term is included in one factor function,  $\psi_c$
  - \* clique structure links *groups of variables*, i.e.,  $\{\{X_i\} \cup X_{\pi_i}, \forall i\}$
  - \* normalisation term trivial,  $Z = 1$

1. copy nodes

2. copy edges, undirected

3. 'moralise' parent nodes





# Why U-PGM?

- Pros
  - \* generalisation of D-PGM
  - \* simpler means of modelling without the need for per-factor normalisation
  - \* general inference algorithms use U-PGM representation (supporting both types of PGM)
- Cons
  - \* (slightly) weaker independence
  - \* calculating global normalisation term ( $Z$ ) intractable in general (but tractable for chains/trees, e.g., CRFs)

# Mini Summary

Undirected probabilistic graphical models (U-PGMs)

- Definition
- Conversion to D-PGMs
- Pros/Cons over D-PGMs

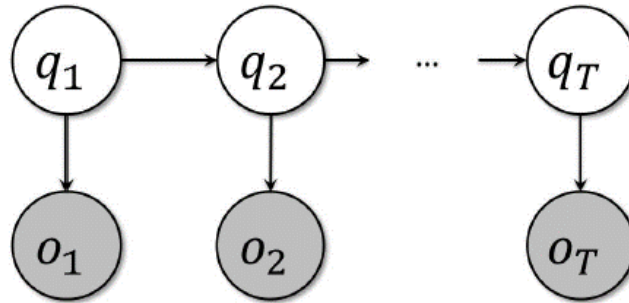
Next: Examples and applications of PGMs

# Example PGMs

*The hidden Markov model (HMM);  
lattice Markov random field (MRF);  
Conditional random field (CRF)*

# The HMM (and Kalman Filter)

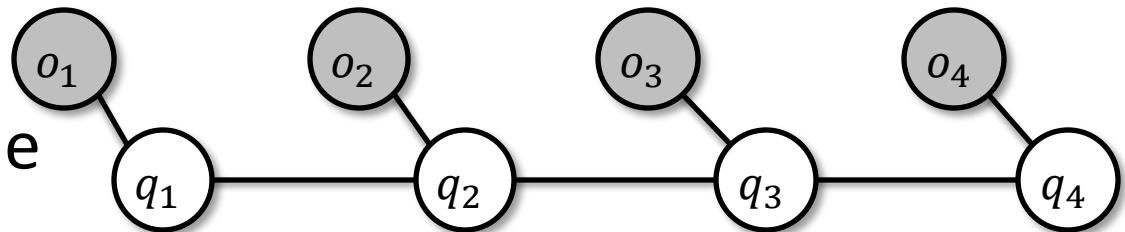
- Sequential observed **outputs** from hidden **state**



$A = \{a_{ij}\}$  transition probability matrix;  $\forall i : \sum_j a_{ij} = 1$   
 $B = \{b_i(o_k)\}$  output probability matrix;  $\forall i : \sum_k b_i(o_k) = 1$   
 $\Pi = \{\pi_i\}$  the initial state distribution;  $\sum_i \pi_i = 1$

- The **Kalman filter** same with continuous Gaussian r.v.'s

- A **CRF** is the undirected analogue

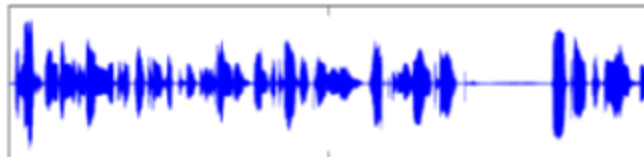


# HMM Applications

- NLP – **part of speech tagging**: given words in sentence, infer hidden parts of speech

“I love Machine Learning” → noun, verb, noun, noun

- **Speech recognition**: given waveform, determine phonemes



- Biological sequences: classification, search, **alignment**
- Computer vision: identify who's walking in video, **tracking**

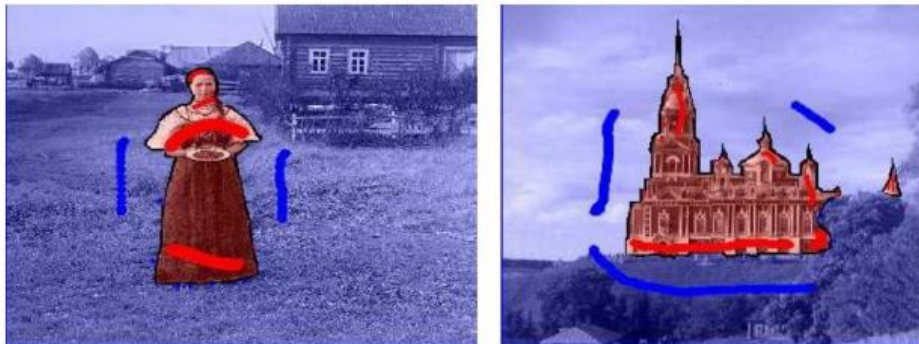
# Fundamental HMM Tasks

HMM Task	PGM Task
<b>Evaluation.</b> Given an HMM $\mu$ and observation sequence $O$ , determine likelihood $\Pr(O \mu)$	Probabilistic inference
<b>Decoding.</b> Given an HMM $\mu$ and observation sequence $O$ , determine most probable hidden state sequence $Q$	MAP point estimate
<b>Learning.</b> Given an observation sequence $O$ and set of states, learn parameters $A, B, \Pi$	Statistical inference

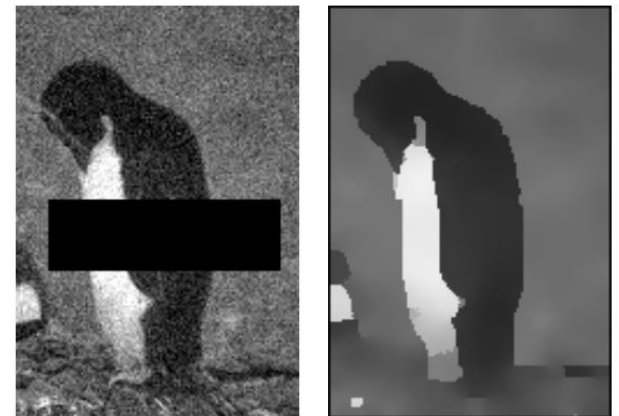
# Pixel labelling tasks in Computer Vision



Semantic labelling (Gould et al. 09)



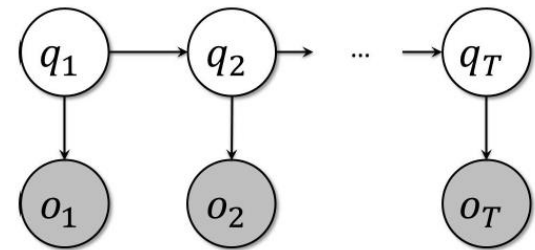
Interactive figure-ground segmentation (Boykov & Jolly 2011)



Denoising (Felzenszwalb & Huttenlocher 04)

# What these tasks have in common

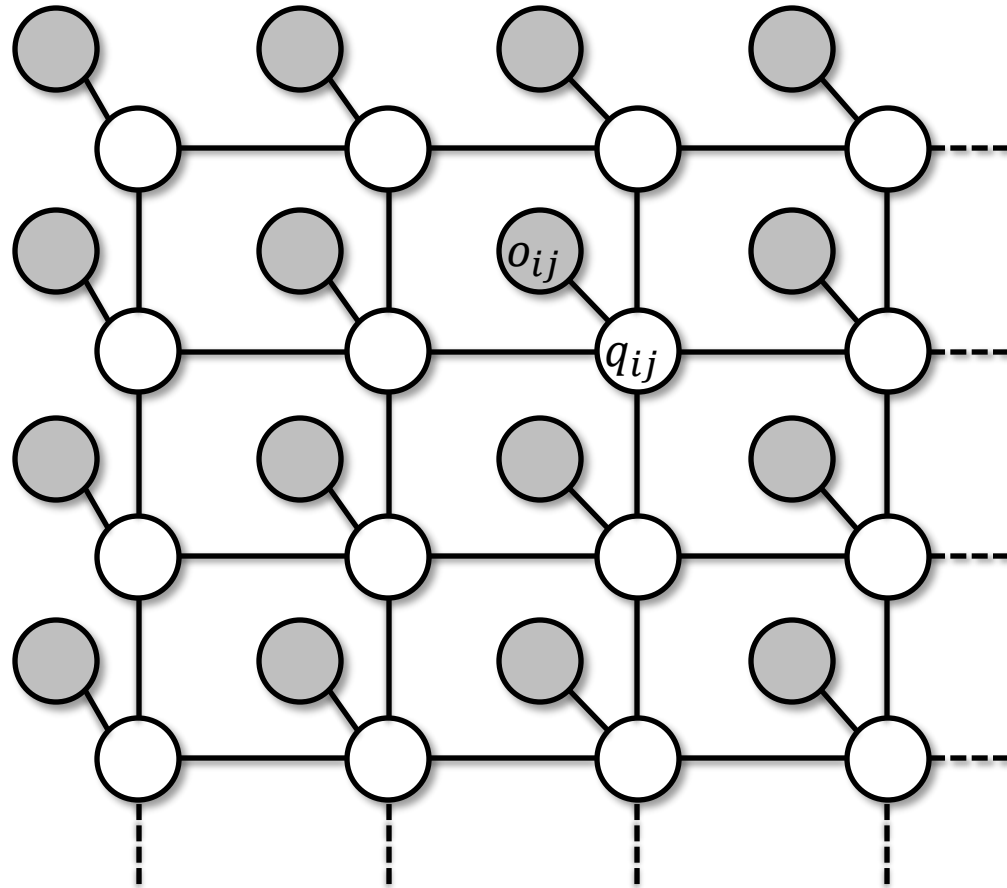
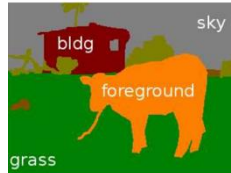
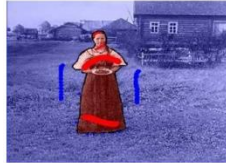
- Hidden state representing semantics of image
  - \* Semantic labelling: Cow vs. tree vs. grass vs. sky vs. house
  - \* Fore-back segment: Figure vs. ground
  - \* Denoising: Clean pixels
- Pixels of image
  - \* What we observe of hidden state
- Remind you of HMMs?





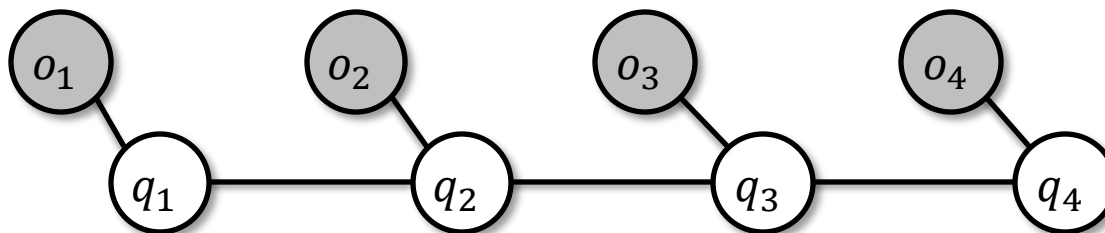
# A hidden square-lattice Markov random field

- **Hidden states:**  
square-lattice model
  - \* Boolean for two-class states
  - \* Discrete for multi-class
  - \* Continuous for denoising
- **Pixels:** observed outputs
  - \* Continuous e.g. Normal



# Application to sequences: CRFs

- Conditional Random Field: Same model applied to sequences
  - \* observed outputs are words, speech, amino acids etc
  - \* states are tags: part-of-speech, phone, alignment...
- CRFs are discriminative, model  $P(Q/O)$ 
  - \* versus HMM's which are generative,  $P(Q,O)$
  - \* undirected PGM more general and expressive



# Summary

- Probabilistic graphical models
  - \* Motivation: applications, unifies algorithms
  - \* Motivation: ideal tool for Bayesians
  - \* Independence lowers computational/model complexity
  - \* PGMs: compact representation of factorised joints
  - \* U-PGMs
- Example PGMs and applications

**Next time:** elimination for probabilistic inference