Probabilistic Graphical Models

Lecture 1. Introduction

- Better to give a probabilistic distribution than a static decision (this and just this)

problem → represent it (build model → relevant fact and rules?) → (by means of logical theory.) → ask questions. → Inference → yes/no questions; is the question satisfied? Use modus ponens, tollens... (tools)

things are not just true/false. (uncertainty)

with uncertainty (Prob)

problem → representation (define prob. dist. → relevant variables) → questions → Inference → rewrite q's as prob. query; obtain prob. dist. by using prob. inference (tools)

1. $\{C=\{T,F\}, F=\{T,F\}, I=\{T,F\}\}$ → $P(C,F,I)$ → $P(I|C=T)$? → prob. dist. over I using inference.
   - C: cough, F: fever, I: flu (infected)
2. Also, $D=\{C, F, None\}$ → $P(C,F,M,D)$ → $\text{argmax}_d\ P(D=d|C=T, M=T)$?
   - D: disease, C: cancer, F: flu; M: hemoptysis $=\{T,F\}$; which disease is more likely?

- Types of queries: 1. Conditional probabilities queries: find marginal distribution → $(P(I|C=T))$
  2. MAP queries: find value of that maximizes a prob distribution

PGMs get involved when: we don't have an expert to know the probabilistic distributions: just data

problem → representation (set of possible prob. dist. of variables given data) → question → learning: choose most likely prob dist.

- bayesian approach: initial beliefs (prior) → refine.

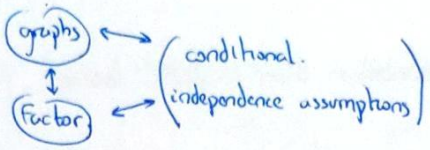
vs.
- Frequentist: coin HTTTTT: $\theta = 5/6$
- Bayesian: initial belief $(1/2)$ + ↑ $p(\theta)$

- Probabilistic approach to Data Science: Assumptions → Identify Patterns → Predict & Evaluate → Criticize model → refine

- Big/Huge Data: we can't reason on a joint prob. dist. over all variables at same time → PGM.
  - encode prob. dist. over large number of variables in a compact way
  - learn from available data
  - efficiently answer q's

  PGM → use graphs! → easy to viz, math adq.
  - nodes: var
  - edges: possible relations.

- PGMs: graphs + set of possible prob. dist
  - types: directed acyclic, undirected, bipartite
  - it is a factorization of the prob. dists.



- Independent random variables: $P(X,Y) = P(X)P(Y)$ $\quad X \perp\!\!\!\perp Y$
  - marginal independence
  $P(X|Y) = P(X)$ or $P(Y|X) = P(Y)$

- Conditional independence: given Z: $P(X,Y|Z) = P(X|Z)P(Y|Z)$ $\quad X \perp\!\!\!\perp Y \,|\, Z$
  - when two variables have common origin, they generally are not marginally independent.
  - if we know the source, we could model the intermediates, making it a conditional independence scenario.

  - independence enhances simplicity: $P(X)$ with $X = Y \cup Z$, $Y \perp\!\!\!\perp Z$
    - $P(X)$: X with 10 binary var
    - $Y \perp\!\!\!\perp Z$: each has 2 bin vars
    - $\Rightarrow$ 2^{10} (1024) $\ggg$ $2 \cdot 2^5$ (64) parameters.
    efficient.

- Structural Statistical Model: two variables X, Y $\xrightarrow{\text{two poss.}}$
  1. $M_1 = \emptyset$ → they are not independent.
  2. $M_2 = \{X \perp\!\!\!\perp Y\}$

  } generalize to n.

  - A distribution respects model M if it satisfies all independencies included in M.

- Factorization: reexpression into product of simpler factors.

prob. dist. ↓ $P(X_1, \ldots, X_{10}) = P(X_1, \ldots, X_5)\, P(X_6, \ldots, X_{10})$ using independence.

  - also: we can move to smaller spaces: $P(X,Y,Z) = P(X|Z)P(Y|Z)P(Z)$

- Inference: queries, prob. based operations.
  - exact: marginalization, conditioning, belief propagation...
  - approx: random sampling

- Learning
  - parametric
  - structural (NP)

  For robust results: tradeoff ⇒ model complexity ↔ available data.

Lecture 2. Probability Overview.

- Learn $P(Y|X)$ instead of $F: X \to Y$
- Joint distribution of M variables: 2^M combinations → learn a value per each (probabilities)
  - ↳ $P(E) = \sum_{r} P(r)$, r = rows that match E

x	y	z	prob
0	0	0	0.1
0	1	0	
0	0	1	
0	1	1	

$$P(E_1|E_2) = \frac{P(E_1 \wedge E_2)}{P(E_2)} = \frac{\sum \text{rows that match both}}{\sum \text{rows that match } E_2}$$

- learn joint distribution from train → use it to calculate using test data.

problem: sometimes data is not enough. → # rows in table
samples needed to learn faithfully
uncommon combinations (not enough observed)

solutions:
smart representations (Bayesian networks, PGM)
estimate prob. from sparse data (maximum likelihood/a posteriori estimates)

- IF $p(X_A|X_B, X_C) = p(X_A|X_B)$, X_A is independent from X_B given X_C.

not: $X_A \perp\!\!\!\perp X_B | X_C$

not: uppercase = variables, lowercase = value

- prop: $X_A \perp\!\!\!\perp X_B | X_C \Rightarrow p(x_A, x_B|x_C) = p(x_A|x_C) \cdot p(x_B|x_C)$
- Chain rule: $p(X, Y, Z) = p(X|Y, Z) \cdot P(Y|Z) \cdot P(Z)$

↳ allows to simplify the factorization

$X = X_1, \dots, X_5$ with $3 \perp\!\!\!\perp 4 | 1, 5$ ⇒ order Z Y X → 1, 4, 5, 3, 2

$$p(X) = p(X_{1,4,5})\, p(X_3|X_{1,4,5})\, p(X_2|X_{1,3,4,5})$$

4 is redundant. (due to independency)

- Let A, B be a partition of V. Then, $\forall x_A, x_B,\ p(x_A) = \sum_{x_B} p(x_A, x_B)$

We can have a marginal dist for each value $X_B = x_B$: $\sum_{x_A} p(x_A|x_B) = 1$

Also, $p(x_A|x_B) = \frac{p(x_A, x_B)}{p(x_B)} = \frac{p(x_A, x_B)}{\sum_{x_A} p(x_A, x_B)}$

↓ post definition.

} collaborates by reducing number free params.

$p(x) = r_x - 1$
$p(x,y) = r_x \cdot r_y - 1$
$p(x|y) = (r_x - 1) \cdot r_y$

} compute number of free params.

- Count parameters without ind.

A \ B	+	−
+	a	b
−	c	1−a−b−c

⇒ 3

with ind:

A \ B	+	−
+		b
−		(1−b)

⇒ 2 ⇒

A \ B	+	−
+	a·b	b(1−a)
−	a(1−b)	(1−a)·(1−b)

Less complexity, avoiding modelling irrelevant params. linked to soft conditional dependencies, helps manage trade-off complexity ↔ data
↓
overfitting

> Exercise:

We want $P(H=+ \mid V=-)$

We have $P(V \mid H=+)\begin{cases} +=0.43 \\ -=0.57 \end{cases}$

$P(V)\begin{cases} +=0.84 \\ -=0.16 \end{cases}$

- Bayes' rule: $P(X \mid Y) = \dfrac{P(Y \mid X) \cdot P(X)}{P(Y)}$ (likelihood, prior)

$\downarrow$

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

$\downarrow$ chain rule by one side — chain rule by the other $\downarrow$

$$P(H=+ \mid V=-) = \frac{P(V=- \mid H=+)\,P(H=+)}{P(V=-)}$$ ($P(H=+)$: missing)

STILL, we can tell the increased risk by not getting vaxxed:

$$\frac{P(H=+ \mid V=-)}{P(H=+ \mid V=+)} \rightarrow P(H=+) \text{ disappears}$$

$\parallel$

7 (17 times more)

Lecture 3. Bayesian Networks.

- Directed acyclic graph $G=(V,E)$ + parameters θ
  - V: variables
  - E: chain rule
  - θ: table of cond. prob. dist.
- Ancestral set: todos los antepasados de un nodo.
- Ancestral ordering: from older to younger (once you visit one variable, all ancestors were visited)
- Moral: convert to undirected graph (remove directions + unite nodes with common children)

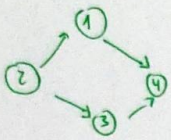
> Exercise.

1. Yes. No
2. parents of X_3: X_5
   children: X_2 / X_6
   parents of X_5: $\emptyset$
   children: X_1, X_3
3. $X_5\ X_3\ X_2\ X_1\ X_6\ X_4$

- Factorization in Bay. Netw/k:

$$P_M(x) = \prod_{i=1}^{n} p(x_i \mid pa_i, \theta_i)$$

pa_i: set of variables pointing towards X_i



$$p(X) = p(X_4 \mid X_1, X_3)\, p(X_1 \mid X_2)\, p(X_3 \mid X_2)\, p(X_2)$$

(ancestral ordering)

(Chain rule)

$$p(X) = p(X_4 \mid X_1, X_2, X_3)\, p(X_1 \mid X_2, X_3)\, p(X_2) \Rightarrow \begin{matrix} X_1 \perp\!\!\!\perp X_3 \mid X_2 \\ X_4 \perp\!\!\!\perp X_2 \mid X_1, X_3 \end{matrix}$$

} independence of the rest given the parents.

> Exercise 1. $X_2, X_1, X_5, X_6, X_4, X_3$

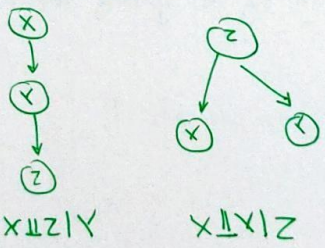
2. $P(X) = P(X_3 \mid X_4, X_5)\, P(X_4 \mid X_6, X_1)\, P(X_6 \mid X_2)\, P(X_5 \mid X_1)\, P(X_1 \mid X_2)\, P(X_2)$

↳ prob(each node | parents)

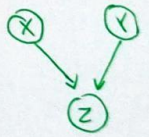
3. c)

d) also, but with way too many repeating parameters.

- Influence flow is stopped



Influence starts if Z is observed.



$X \not\perp\!\!\!\perp Y \mid Z$

v-structure

(while $X \perp\!\!\!\perp Y$)

$\Rightarrow X \not\perp\!\!\!\perp Y$

not gross example.

> Exercise: FF T F F F

T: ↓ independent cond. dependent

F: ↓ reposor

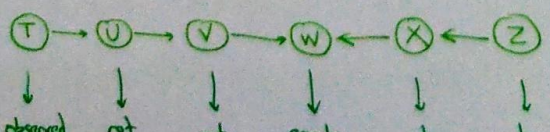
F: ↓ v-struct

F: ↓ v-struct w/ extra steps.

Lecture 4

- Active trail: trail $X_1 \rightleftharpoons \dots \rightleftharpoons X_n$ of probabilistic influence flow

it is active when



T: ↓ observed

U: ↓ not obs.

V: ↓ not obs.

W: ↓ can be observed (bottom)

X: ↓ not obs.

Z: ↓ obs.

↳ this observation activates the flow.

only time you can have an observations in middle of an active track (except v-struct) (*)

- d-separation: generalization of conditional independence to set of variables
  - ↳ independent if there is no activations. (
  - how to check: 1) identify ancestors of X, Y, Z
    2) remove the rest, moralize graph ↰ generates new paths from v-structures. (by moralizing you can see all the trails)
    3) does Z block all the paths from X to Y?
       - (*) → observation ≡ no activation ⇒ independence holds.
         no obs. ≡ active trail

- Markov's blanket: minimal set of variables you need to observe in order to make a variable independent from the rest. have enough to update the prob. dist.
  - ↳ parents, children and children's parents. useful to avoid useless observations → ML efficiency.
  - Block all the paths + the influence of v-structs.

- Redundancy: two DAGs produce same dependance model if they have the same edges (mod. direction) and same v-struct.

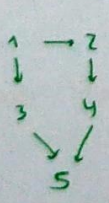
- I(G) is the set of independence statements that hold on G.
  I(P) " " " " " " P
  ↓ the world.
  } G is an I-map of P if $I(G) \subseteq I(P)$
  I-map: Independency maps.

  The other way around may not be true (wasted params.)
  ↓ assume dependencies that are not real

> I-map quiz: d)

- Given a Bay. Net. G, a prob. dist P.

  P factorizes according to G ⟺ P factorizes.

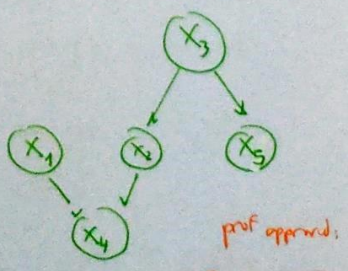
- Factorization of 

  $\prod P(X_i \mid \text{parents of } X_i)$

  $P(X_1)P(X_2|X_1)P(X_3|X_1)P(X_4|X_2)P(X_5|X_3,X_4)$
  → ancestral ordering.

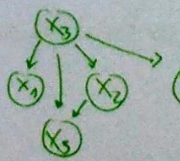
- $I = \{X_3 \perp\!\!\!\perp X_1 | X_2;\ X_2 \perp\!\!\!\perp X_1,\ X_5|X_3;\ X_3 \perp\!\!\!\perp X_4;\ X_4 \perp\!\!\!\perp X_3, X_5 | X_1, X_2\}$

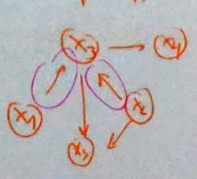
  chain rule ↓ $P(X_4|X_1,X_2,X_3,X_5)P(X_2|X_1,X_3,X_5)\cdot P(X_5|X_1,X_3)P(X_1|X_3)P(X_3)$
  ↓ start from largest ↳ deleted due to independencies.

- $I = \{X_4 \perp\!\!\!\perp X_2, X_1 | X_3;\ X_2 \perp\!\!\!\perp X_1;\ X_5 \perp\!\!\!\perp X_1, X_4 | X_2, X_3\}$

  $P(X_5|X_1,X_2,X_3,X_4)P(X_4|X_1,X_2,X_3)P(X_2|X_1,X_3)P(X_1|X_3)P(X_3)$

  diff →
  prof approved:
  

For $P(E)$ ($E \in 0,1,2$) we need ≥ 2 parameters.

↳ number of params in a BayNetwork.

Lecture 5. Markov Networks

Some scenarios are not representable by a Bayesian Network.

- Markov Networks:
  - undirected graph
  - Factors over cliques (instead of dist. per variable + parents)
  - requires normalization: partition function
  - independence determined by separation → instead of d-sep.
  - $F \overset{\text{always}}{\underset{\text{only for pos. dist.}}{\rightleftarrows}} I$ vs Bay. always / always
  - Markov blanket ≡ neighbours

Lack of v-structures.

- ϕ is not a probability distribution; partition function

$\phi_1(A,B)$ "strength"

A	B	
0	0	30
0	1	5
1	0	1
1	1	100

graph undirected → factorization

$$P(x) = \frac{1}{Z} \prod_{i=1}^{F} \phi_i(x_{\phi_i})$$

hard to compute

Factorization constant: $Z = \sum_x$

Factors

ϕ ← partition func.

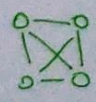
directed.

(instead of $P(x) = \prod_{i=1}^{v} p(x_i | pa_i)$)

prob. dist for each value of the parents.

X_4	X_2	$P(X_4 \mid X_2)$
a	b	0'3
b	b	0'7

a	a	0'6
b	a	0'4

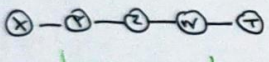
- Clique: set of variables where all (vertexes) of them are connected to each other.
  - 1-vertex: all single variables.
  - 2- : connected variables
  - 3- : 
  - 4- : 

- maximal clique: a clique not contained by any other clique.

> Factorization in MNs Exercise

1. Valid (is over cliques)
2. Non-valid
3. Valid (over maximal cliques)
4. Valid
5. Valid (all pair-wise cliques)

> 1. Y
2. N: A,B,D not a clique
3. N:
4. N

- Independence: if there is no path between the variables.
- Conditional indepence: if the condition variable blocks any connection between the two variables.
- Active trail 

if none is observed.

> Independence Ex

None

> Conditional Ind. Ex.

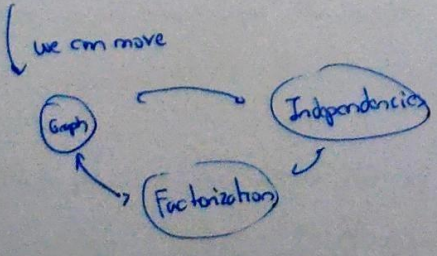
a) F
b) T
c) T
d) F

- Markov blankets: neighbours of a node (because we don't have v-structs)
(if these are observed, observing any other doesn't influence at all)

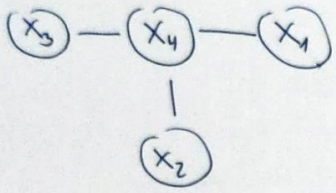
- Hammersley Clifford Theorem: Markov structure (network) $\mathcal{H}$, P prob. dist' I.

P factorizes over $\mathcal{H}$ $\Rightarrow$ P satisfies independences in $\mathcal{H}$

P satisfies ind. in $\mathcal{H}$
P is positive
$\Rightarrow$ P factorizes.



> Exercise



▷ T $(X_1 \perp X_3 | X_4)$

▷ F $(X_2 \perp X_3 | X_1)$

▷ F $(X_2, X_3 \perp X_1)$

> Exercise (provide a factorization)

$P \propto \phi(2,6,4,7)\,\phi(2,3,4)\,\phi(1,2,3)\,\phi(3,5)$ → maximal cliques

• prop: A BN without V-structs ≡ MN

(normalization)

• Number of parameters for a Markov Networks

(Bayesian): $\sum_{v} \Omega_{PA}(\Omega_{x_v} - 1)$

v: all vars; PA: all parent values

X_4	X_2	$P(X_4, X_2)$
a	a	0,4
a	b	0,6
b	a	0,2
b	b	0,8

need 2 params.

MN:

would need 4.

$\sum_{F} \Omega_{S_\phi}$ □ → nothing else! It is not a prob. dist.

F: factors with scope S_ϕ; Ω_{S_ϕ}: all poss. values in the scope

• Applications: - denoising: one observed variable, one hidden variable } for each pixel → input image, → models real values.

- segmentation (similar)

• Factor Algebra: - product

X	Y	ϕ
0	0	3

Y	Z	ψ
0	0	5
0	1	4

→

X	Y	Z	$\phi \times \psi$
0	0	0	15
0	0	1	12

always larger or equal table

commutative, associative, ∃ neutral element

- reduction: observe a set of variables (reduce table by fixing variables, and selecting rows where values apply)

multiples values ≡ AND.

- prop: $(\phi_1 \times \phi_2)[U=u] = \phi_1[U=u] \times \phi_2[U=u]$

uses smaller tables.

- marginalization: sum the rows that have the specified values

X	Y	$\sum_{z_i} \phi$
0	0	sum of 000 and 001

- prop: $\sum_X (\phi_1 \times \phi_2) = \phi_1 \times \sum_X \phi_2$

if $X \notin Scope(\phi_1)$

generalization: $\sum_X (\phi_1 \times \phi_2 \times \phi_3) = \phi_1 \times \sum (\phi_2 \times \phi_3)$

> Reduction and marginalization exercise

$p(X|Y=1)$

reduce:

X	Z	$\phi[Y=1]$
0	0	11
0	1	2
0	2	1
1	0	4
1	1	1
1	2	9

marginalize →

X	$\sum_Z$
0	14
1	14

normalize → (for prob dist)

X	$p(X\|Y=1)$
0	0.5
1	0.5

> 1. Reduce $\phi_1[B=b]$ ...

Reduce $\phi_2[B=b]$.

$$P(A,C|B) = \frac{1}{Z} \phi_1[B=b]\,\phi_2[B=b] \sum_D (\phi_3 \cdot \phi_4)$$

3.

$$P(A|B=b) = \frac{1}{Z} \sum_B \left(\sum_A (\phi_1 \phi_4)\, \phi_2[C=c] \right) \phi_3[C=c]$$

|| → more efficient

$$\frac{1}{Z} \sum_A \left(\phi_4 \cdot \sum_B (\phi_1 \cdot \phi_2[C=c]) \right) \cdot \phi_3[C=c]$$

Lecture 6.

- normalization: it will always result in one variable and a value for all its possible values.

> Exercise "Factors in Markov Network": b), c), e)

product of everything and getting rid of all the variables (summing): normalizing factor to get a prob. dist.

Plate Models.

notation: boxes mean that the variables inside are repeated n times. (also called plate)

shadow variables: those that are usually observed.

> Plate semantics exercise: a), b), d)

> Plate interpretation exercise: b)

> Grounded Plates: b)

- limitations: no connections between copies of the same plates.
  no connections between specific variables depending on another variable.

• Temporal/late models: future and past are independent given the present.
  plate
  time invariance $P(X(t+1)|t)$ always the same $\forall t$

- Dynamic Bayesian Network 1) $X(0)$ uses Bayesian Network
  2) $X(t+1)$ given $X(t)$ → Conditional BN

> Applications of DBNs: a) F → because it's connecting to several connections
  b) T
  c) F
  d) T

> Independencies in DBNs: a)
  b)
  c)
  d)

Exact Inference.

• Conditional Probability Query: Find the distribution $P(Y|E=e)$ given a partition of variables $X=(Y,H,E)$
  ↓ exponential complexity (even if P factorizes to graph $\mathcal{H}$)
  Evidence ↑ Hidden
  not polynomial time except for some graphs (specific)

- Variable Elimination (for 1 query)
- Message Passing (for several queries)

1. Eliminate all variables in Y → a) For all variables that have said variable V in the scope
   b) Keep a copy without them
   c) product between all them and V
   d) Marginalize V
   e) sum b) and d) and return
2. Normalize

> Visualizing VE: start by X_1 or X_8 because they're in only one factor

$$\sum_{X_8} \phi(X_6, X_8) = \tau_7(X_6)$$

↳ Then continue with X_6 ($\text{scope}(X_6) \subseteq \text{scope}(X_4, X_6, X_7)$) → no large intermediate tables.
but also continuing with nodes with least factors! (X_2 step = bad)

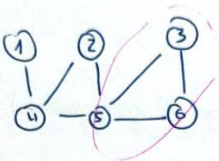
- Induced graph: add those edges that appear during elimination.

  ↳ the more extra edges ≡ the more complex our elimination order was.

  a good heuristic is the size of the biggest clique in the induced graph.

> VE ordering exercise: 1, 3, 6, 2, 8, 7, 5

- with evidence: observe and marginalize.

> Intermediate Factors (Bayesian!) moralize! →  → $P(X_6 \mid X_3, X_5) = f(X_5, X_3, X_6)$

X_5 ← answer. ($f_1(X_4, X_2, X_5)$ and $f(X_3, X_5, X_6)$)

Lecture 7.

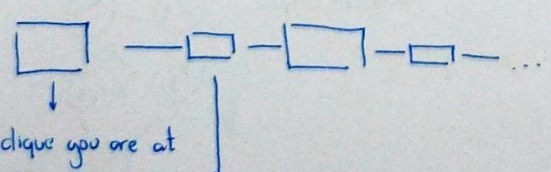
[20 mins late]

- Variable Elimination for MAP (argmax)

- sum of factors: like product but summing
- max-marginal: like marginalization but taking the maximum instead of summing.

- Message Passing:

we are going to use cluster graphs and clique tree

- clique tree: 

} very related to Variable Elimination.

- cluster graph each node is associated with a subset (cluster)

clusters are related to each other with the intersection (sepset) → or $C \cap$

set of beliefs } one factor for each subset β_i / sepset μ → $P \propto \frac{\prod \beta}{\prod \mu}$

now $p(X_j) = \frac{1}{Z} \sum_{C_i \setminus X_j} \beta_i(C_i)$ such that $X_j \in C_i$ (any cluster that contains X_j and marginalize) one only

the transformation is expensive
but the queries are really cheap } Great for many queries

- building clusters: * preservation: all factors ϕ_j need a cluster C_i such that $Scope(\phi_j) \subseteq C_i$.

* running intersection property: if there is a variable in several clusters.

1) There is a unique path among them.

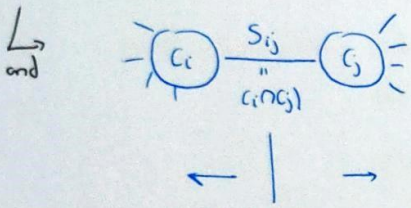
2) The variable is present in all intermediate clusters and sepsets.

> Cluster Graph construction: c), d)

> Family preservation: c) (by definition of preservation)

in clique trees, the sepset is exactly the intersection

and



→ there are no loops in the graph.

both sides are independent given the sepset

> Draw Exercise: a) X_1, X_2, X_3 — X_2 — X_2, X_5 — X_5 — X_4, X_5 — X_4 — X_3, X_4

NO: not satisfying RIP

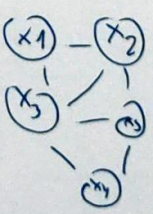
X_1, X_2, X_3 — X_2, X_5 — X_2, X_3, X_4, X_5

too big

↳ problem is 2 and 3 both in too many clusters.

Think of Variable Elimination! Good order: 1, 4, 2, 3, 5

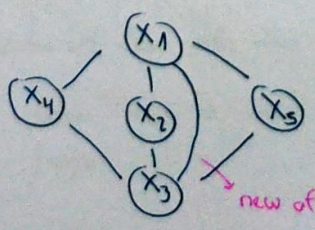
Induced graph:



X_1, X_2, X_3 — X_1, X_3 — X_2, X_3, X_5 — X_3, X_5 — X_3, X_4, X_5

b) Similar. Order: 5, 2, 1, 3, 4

Induced Graph:

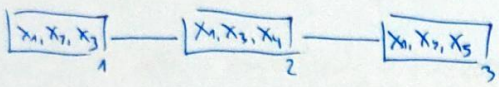


new after eliminating 5

X_1, X_3, X_5 — X_1, X_3 — X_1, X_2, X_3 — X_1, X_3 — X_1, X_3, X_4

- cluster factor: $\psi_i(C_i) = \prod \phi$
  - ϕ: all the factors that were assigned to that cluster.

- message: From one cluster to another



$$\delta_{2\to 3} = \sum \psi_{(2)} \cdot \delta_{1\to 2}$$

- $\sum$: marginalize over all variables not in 2,3 scopes
- $\psi_{(2)}$: cluster factor of sender
- $\delta_{1\to 2}$: all incoming messages of the sender.

belief propagation: select a root

starting from the leaves up to the root: message passing.

select another root until messages are passed in both directions in all edges.

B_i = Factor * all incoming messages.

μ_{ij} = mult. of both messages in an edge.

> Message Ordering Exercise: d)

b, c → not enough messages ×

a → two roots ×

↓

every time we send a message, we need all the incoming msgs. beforehand.

$C_1 \to C_1$, $C_2 \to C_3$, $C_3 \to C_4$, $C_5 \to C_3$ would be wrong.

> Message passing in cluster tree: a) we marginalize getting rid of the variables in 3 not in sepset.

× all incoming msgs.

> b)

- cluster graphs are calibrated when the B_i marginalized for all variable in a sepset is the same for both clusters.

$$\sum_{C_i \setminus S_{ij}} B_i = \sum_{C_j \setminus S_{ij}} B_j$$

↓

not met with loops, etc. (loop = lack of leaves = message passing is not constant) ***

met automatically in clique trees.

- Exact inference:
  1. Represent as clique tree
  2. Calibrate
  3. Answer

} not polynomial.

** alternatively, initialize all messages to 1
pass messages until it is calibrated enough.
it won't be 100% calibrated ⇒ not exact answers. ~~exact~~ inference.

Lecture 8. Approximate Inference

Also sampling

↳ larger M, better estimations.

for conditional distribution: restrict to those samples that satisfy the condition.

> Cat. Dist. c), d), a), d), b, b, b ...

- Forward sampling: follow ancestral ordering
  ↳ for each variable you sample it
  } you get one sample; for answering queries you need to repeat a lot

- Conditional probability query: filter the sample to those that fulfill E=e.
  might have very low prob. ⇒ no samples.

> Rejecting samples: e)

Imagine $P(e) = \frac{1}{10^6}$ very small ⇒ $M = M' \cdot 10^6 = 10^9$ → samples you need to have.
↓
M' = 1000
(samples you want)

instead of generating, we may weigh samples.
these samples will always fulfill evidence E=e.

Instead of summing samples, weighted-sum them to answer queries.

- CPQ for Markov Networks: how to sample? Factors are not prob. dist.
  no ancestor ordering (one can have inner loops) } → other tools.

- Gibbs sampling: 1) pick a sample initial state
  ↓ most used.
  2) keep iterating to update the distributions ← variables are not independent (contrary to assumptions)
  ↓
  only those in the Markov blanket could be needing an update.

then remove samples E≠e. Keep those Y=y

> Gibbs Sampling on BN: c)
not d: active trail (Markov Blanket definition on BNs)

> Gibbs BN (2): F) (children, parents and children's parents.
marginalize on x_{23}'', only unknown value) <-?

- Variational inference: against problematic samplings.

1) Take a simple family of distributions. (Q)
2) Find the closest one to P(Y|E=e)
$Q(Y) \in Q$
3) Use Q to answer queries

~ = will be defined further.

- Family of distribution: "Normal distribution" → depends on μ, σ.
- Closure: through 'cross-entropy' distance find the distribution in the family that is the closest to P

Kullback-Liebler distance: positive, not symmetric, no triangular inequality.

$Q^* = \underset{Q \in \mathcal{Q}}{\operatorname{argmin}}\ KL(Q||P)$ ← instead: maximize Evidence Lower Bound (ELBO)

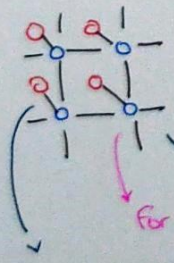
reverse (I-proj) P||Q → M-proj.

- Mean-field algorithm: use Q that factorizes all variables independently: $Q(X) = \prod_{i=1} Q_i(X_i)$ → restrictive; ↳ efficient after all

you can update one marginal at a time (given that the rest are fixed) without updating everything } 'coordinate-descent'

- M-projection: takes care of probability mass.
I-projection: owes the mdd of the distribution. (don't put prob. mass where dist. is not defined ⇒ high penalization)

Lecture 9. Image denoising



o - true value (hidden) → one parameter for each.
o - input value (noisy, observed)
↘ also a parameter for each link between adjacent pixels (smoothness) – } corruption probability.*
for each pixel.
*factors
⇓
prob. dist.
correlation between true and hidden – one parameter each

- energy_diff_if_swap: how will the energy function change if we swap one value.
- swap values until energy function stops improving.

Lecture 10. Probabilistic Learning.

Can we learn from data? Without knowing the structure, the parameters...

- **Model learning**: we have a dataset and maybe domain experts.
  ↳ objective: combine them to infer a model M that approximates P^*

  restrict the set of possible models (needed assumption): PGMs.

  extract info from dataset D to encode the structure and the parameters. } remember overfitting.
  structural + parametric learning.

  trading performance, complexity, dataset.

- problem of freq. approach: you need infinite samples. (empirical distribution, tends to true distribution)
- **laplace smoothing**: to avoid never seen events

$$\frac{N_1 + 1}{N_0 + N_1 + 2}$$ ← number of classes.

- sufficient statistics to get an estimate Coin: N_0 and N_1 / N_0 and N

> Sufficient Statistics Exercise: c)

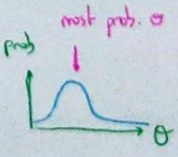
Lecture 11.

> Parametric learning in MNs and BNs: a) → you can have info of a variable in several factors.
not gonna be seen.

- **Maximum likelihood**: θ that maximizes $P(data|\theta)$ ← likelihood
- **Maximum a posteriori**: θ that maximizes $P(\theta|data) = \frac{P(data|\theta)P(\theta)}{P(data)}$

- **Bayesian estimation**: estimated from data but updated with expert knowledge:

Posterior $(\theta|data)$
Prior $(\theta) = P(\theta)$ → beta prior dist
Likelihood: $P(data|\theta)$



→ in terms of hyper params (α, β)

Beta prior is appropriate when our data follows one of these distributions: Bernouilli, Binomial

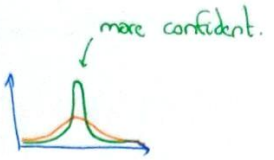
↳ The higher the parameters, the more dense the prob. mass → 

↳ The posterior is also a beta. (conjugate posterior) → with higher hyperparameters.

- map estimate: Formula and def. in slides.

For multinomial dist. it is similar, but with a Dirichlet.

- The hyperparameters represent most prob. value and confidence (in prior)



- non-informative prior: bayesian approach without the frequentist problems.

- Equivalent sample size: both 'weights' sum up to 1 → they are weights (weighting data and prior means)

Posterior mean of Beta distribution → mean_data (as ∞)

(Remember Laplace smoothing was doing the weighted sum with the 'ideal' distribution)

Laplace smoothing = Dirichlet MAP estimate with $\beta_{0,1} = 2$

Lecture 12. Structural Learning

↳ We want to approximate a prob. dist from a dataset.

Removing edges: less params, better generalization, might not learn real distribution.
Adding: the opposite.

> Counting params. ▷

▷ In the Bayesian Network Factorization, only $P(X_4|X_1,X_6)$ changes to $P(X_4|X_1,X_2,X_6)$

$\prod_i P(X_i|PA_i)$ → parents

formula counting params: $(\Gamma_{X_i} - 1) \cdot \Gamma_{PA_i}$ (possible values of X_i; possible comb. of parents val.)

remember in this case, binary variables:

$(2-1) \cdot 2^{(2)}$ vs. $(2-1) \cdot 2^{(3)}$ (parents)

- Why: we need the model to use it
  we may be interested in the link between the variables (this can be the goal: medicine)

- Score based structural learning: 1) Define scoring function (penalized) likelihood → refer to slides.
  2) Find structure that maximizes it Hill-Climbing, genetic alg.

- scoring function reduces to Mutual Information (if we use likelihood): it represents the strength of the dependence relationship $X_i \not\perp\!\!\!\perp PA_i$

Mutual Information is "the part of the variable" that can be explained "with its parents".

- likelihood scoring function: ⊕ pros: • decomposes in one term per variable.
  ⊖ con: • MI is always ≥0, and mostly >0
  ||
  adding a new edge always adds score ⇒ best score is totally complete graph
  overfitting!

> Likelihood score exercise: a) T: we only change one direction (no new edges) without involving v-structs.
b) F: adding one edge ⇒ increases score.
c) T: some reason
d) F: we can not know for all datasets (add one, remove one)

not directly related to score.
but it is relevant to know if cond. ind. statements change or not (the num. of)

- avoid overfitting: explicitly control it (limit)
  penalize complexity in the score (options) → penalized log-like likelihood (BIC-Score)
  (bayesian approach: compute prob. of all possible structures)
  penalize {edges / params.

* BIC Score: in terms of params.
If N grows (number of samples) ⟶ more emphasize on fitting to data.

> BIC Sore: a) T: same parameters and same N if same D
b) F:
c) F: there can be a dataset that balances both parts of formula.
d) F

• Explore options and check which one maximizes:

[..] → 30 mins (find tree or forest algorithm → all variables have ≤1 parent)

> Recovering directionality. d)

• Local search: moving across neighborhoods, choose best subgraph G (it may not be the global best)
- possible modifications: add, remove or reverse an edge → evaluate all one-action possible modification. (Greedy Hill climbing)
costful but use score decomposability

> Calculating Likelihood Differences: remember $\sum_v MI(X_v|PA_v)$

c) ↳ identify the terms that are different in all graphs.

> Hidden variable