CS 461: Machine Learning Principles

Class 21: Nov. 18

EM: **E**xpectation and **M**aximization Algorithm

& Learning a Bayesian Network

Instructor: Diana Kim

Outline

1. Indirect Inference: MCMC to compute $P[X|e]$ (posterior)

2. Learning a Bayesian Network:
   a network structure is known and data is fully observed
   a network structure is known and data is partially observed

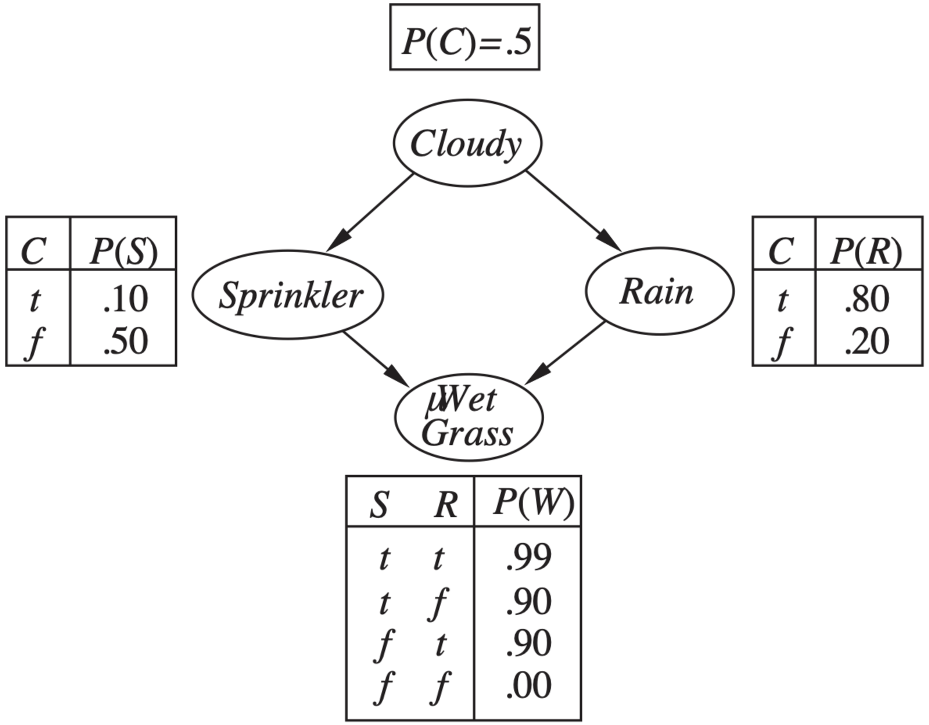
3. EM Algorithm (**E**xpectation-**M**aximization)
   This is MLE (**M**aximum **L**ikelihood **E**stimation) when data is partially observed.
   Example: Learning a Gaussian Mixture Model

3. Learning a Bayesian Network
   a network structure is known and data is partially observed.

4. Application of a Bayesian Network: Bonaparte (DNA-matching software)

MCMC (Markov Chain Monte Carlo) Simulation
Another name: Gibbs Sampling
(Approximate Inference)

Q: $P[\text{Rain} \mid \text{Sprinkler}: +, \text{WetGrass}: +]$?



- [Method 1: Variable Elimination]

$$P[R|S+,W+] = \alpha \sum_C P[R,S+,W+,C]$$

$$= \alpha \sum_C P(C)\cdot P(S+|C)\cdot P(R|C)\cdot P(W+|S+R)$$

$$= \alpha P(W+|S+R) \sum_C P(C)\cdot P(S+|C)\cdot P(R|C)$$

Figure 14.12 From the book “AI: A Modern Approach”

Q: P[Rain | Sprinkler: + , WetGrass: +]? • [Method 2: Sampling]

$$\frac{\#samples(Rain{:}+, \text{Sprinkler: } + , \text{WetGrass: } +)}{\#\ samples\ (\text{Sprinkler: } + , \text{WetGrass: } +)}$$

We need the samples.

# data	Rain	Cloudy	Sprinkler	WetGrass
1	+	+	+	+
2	-	-	+	+
3	+	+	+	+
4	+	-	+	+
5	-	+	+	+
6	-	-	+	+
…	-	-	+	+

MCMC (Markov Chain Monte Carlo) Sampling / Gibbs Sampling

Gibbs Sampling for Q: $P[\text{Rain} \mid \text{Sprinkler:} +, \text{WetGrass:}\ +]$

(1) Gibbs sampling for Bayesian network starts with an arbitrary state
(with the evidence variable fixed at their observed values)
suppose the initial state is

$$\underbrace{\text{Rain: + Cloud: -}}_{\text{+ will vary}}\ \underbrace{\textbf{Sprinkler: +, WetGrass: +}}_{\text{+ fixed as evidence}}$$

Gibbs Sampling for $P[\text{Rain} \mid \text{Sprinkler:} +, \text{WetGrass:} +]$

(1) Gibbs sampling for Bayesian network starts with an arbitrary state
(with the evidence variable fixed at their observed values)
suppose the initial state is
Rain: + Cloud: - Sprinkler: +, WetGrass: +

(2) Randomly sample a value for one of non evidence variable.
Rain or Cloud (in an arbitrary order)

Gibbs Sampling for P[Rain | Sprinkler: + , WetGrass: +]

(1) Gibbs sampling for Bayesian network starts with an arbitrary state
(with the evidence variable fixed at their observed values)
suppose the initial state is
Rain: + Cloud: - Sprinkler: +, WetGrass: +

(2) Randomly sample a value for one of non evidence variable.
Rain or Cloud (in an arbitrary order)

(3) Cloud is sampled by P[Cloud| Sprinkler: + and Rain: +]
given the current value in the Markov Blanket Variables of "Cloud".
Q: what is Markov Blanket?

+ Markov blanket of a R.V **A** is the the R.Vs that, when conditioned upon, makes **A** conditionally independent of all other R.Vs.

Gibbs Sampling for $P[\text{Rain} \mid \text{Sprinkler: +} , \text{WetGrass: } +]$

(1) Gibbs sampling for Bayesian network starts with an arbitrary state
(with the evidence variable fixed at their observed values)
suppose the initial state is
Rain: + Cloud: - Sprinkler: +, WetGrass: +

(2) Randomly sample a value for one of non evidence variable.
Rain or Cloud (in an arbitrary order)

(3) Cloud is sampled by P[Cloud| Sprinkler: + and Rain: +]
suppose Cloud Sample was Cloud: +
then the state will be changed to
Rain: + Cloud: + Sprinkler: +, WetGrass: +

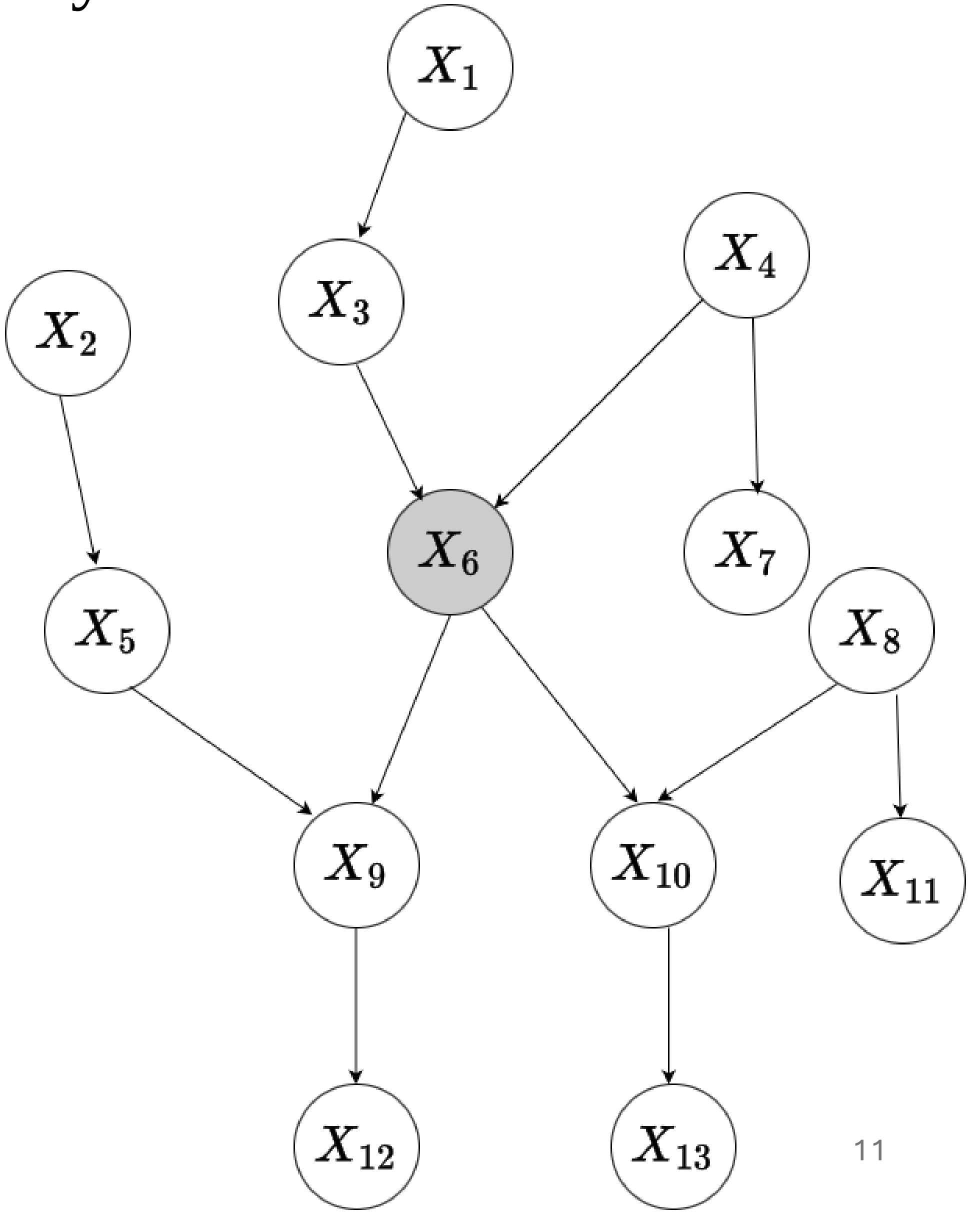
(4) Rain is sampled by P[Rain| Cloud: +, Sprinkler: +, WetGrass: +]
then the state will be changed to
Rain: - Cloud: + Sprinkler: +, WetGrass: +

Markov Blanket of a Variable

The variable is conditionally independent of every other nodes in the graph given its Markov blanket.

Q: Markov Blanket of X_6?

+ parents: X3 and X4
+ children: X9 and X10
+ coparents: X5 and X8

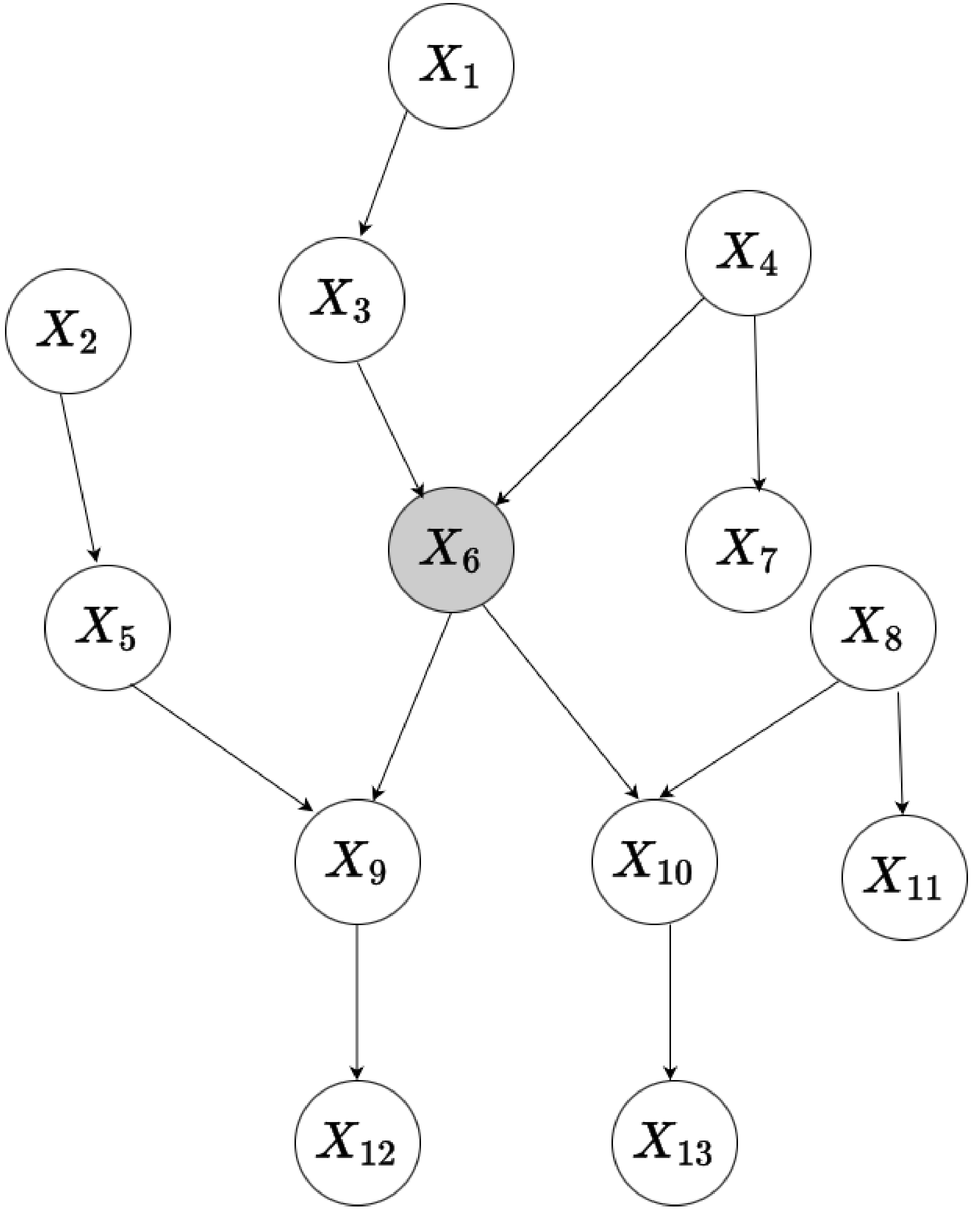


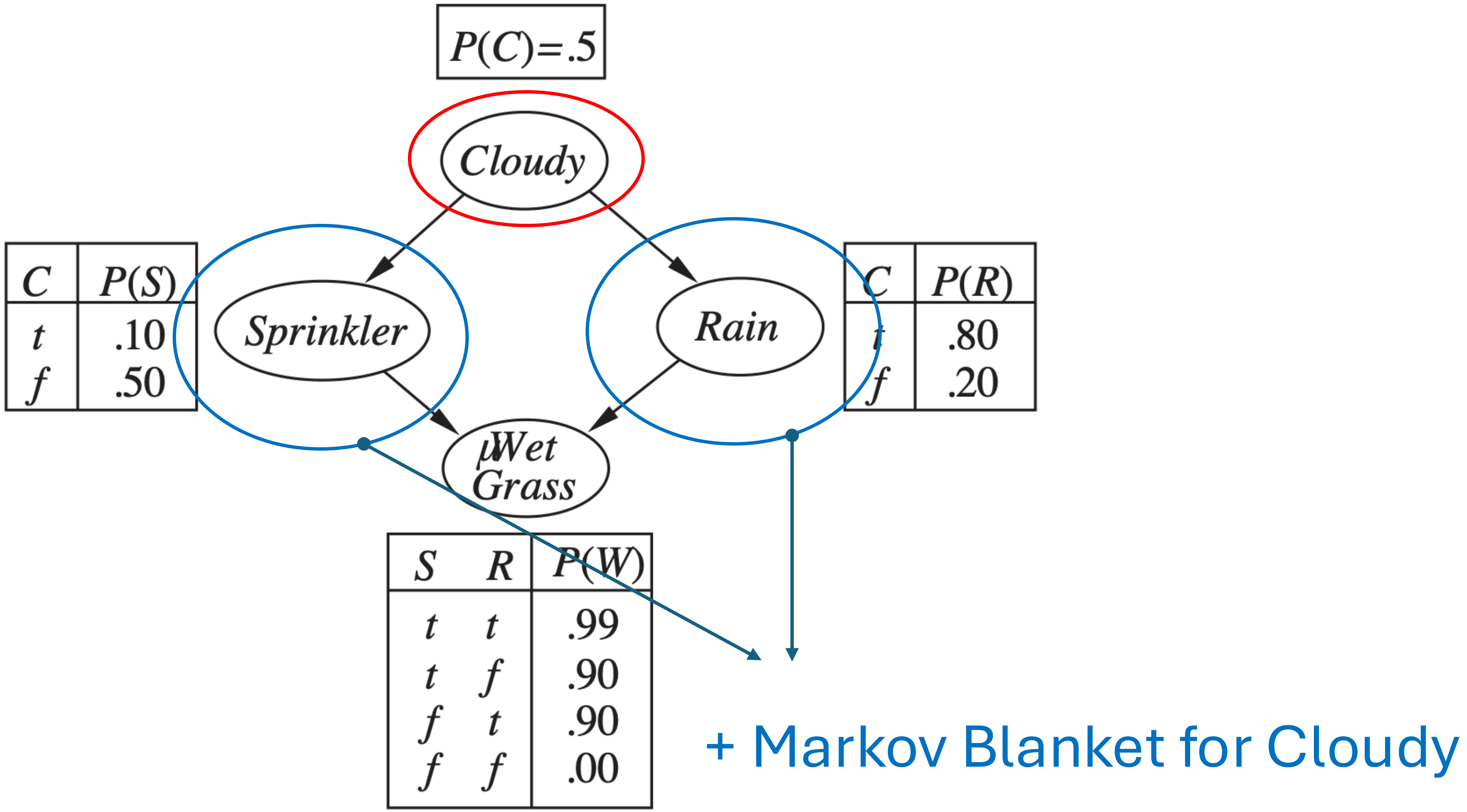
We need the density below to generate the samples.
How can we compute it?
Q: $P[X_6 \mid X_3, X_4, X_5, X_8, X_9, X_{10}]$?

We can imagine a new network consisting of the R.Vs of X3, X4 , X5, X6, X8, X9, X10.

Then P[X6|X3,X4,X5,X8, X9, X10]
= αP[X6, X3,X4,X5,X8, X9, X10]
= αP[X3, X4, X5, X8] P[X6| X3, X4] P[X9, X10|X6]
= αP[X6| X3, X4] P[X9, X10|X6]

The posterior query conditioned by Markov blanket
Is proportional to the product:
P[target|parents]P[children|target]





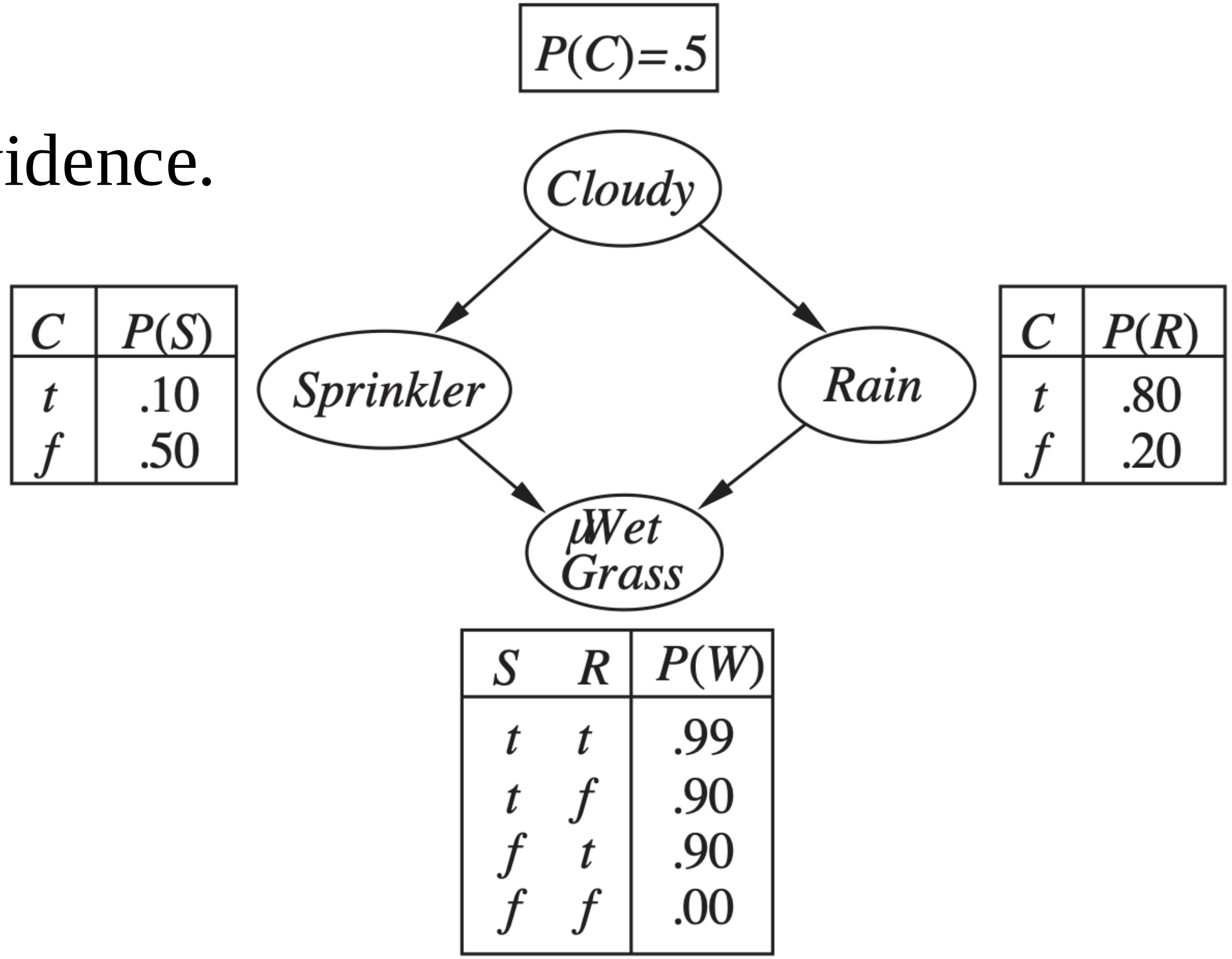
Q: What is the Markov Blanket for Rain? + cloudy, wet grass, sprinkler

**How to compute P[Target | Markov Blanket]?

Given Markov Blanket, the target variable is independent from other R.Vs, so we can imagine a new graph only with the target and blanket variables.

P[Target | Markov Blanket] = α P[Target, Markov Blanket]
= α P[parents] P[co-parents | parents]
P[target | parent]P[children | coparents and target]
= α P[target | parent]P[children | coparents and target]

Example of Computing Posterior, given Markov Blanket Variables as Evidence.



P[Cloud| Sprinkler: + & Rain: +]?

$$\alpha \begin{bmatrix} P[C+]P[Sprinkler+ \mid C+]\ P[Rain+ \mid C+] \\ P[C-]P[Sprinkler+ \mid C-]\ P[Rain+ \mid C-] \end{bmatrix} = \alpha \begin{bmatrix} 0.04 \\ 0.05 \end{bmatrix} \approx \begin{bmatrix} 0.44 \\ 0.56 \end{bmatrix}$$

Learning a Bayesian Network

: we are going to focus on the case where a structure is given.

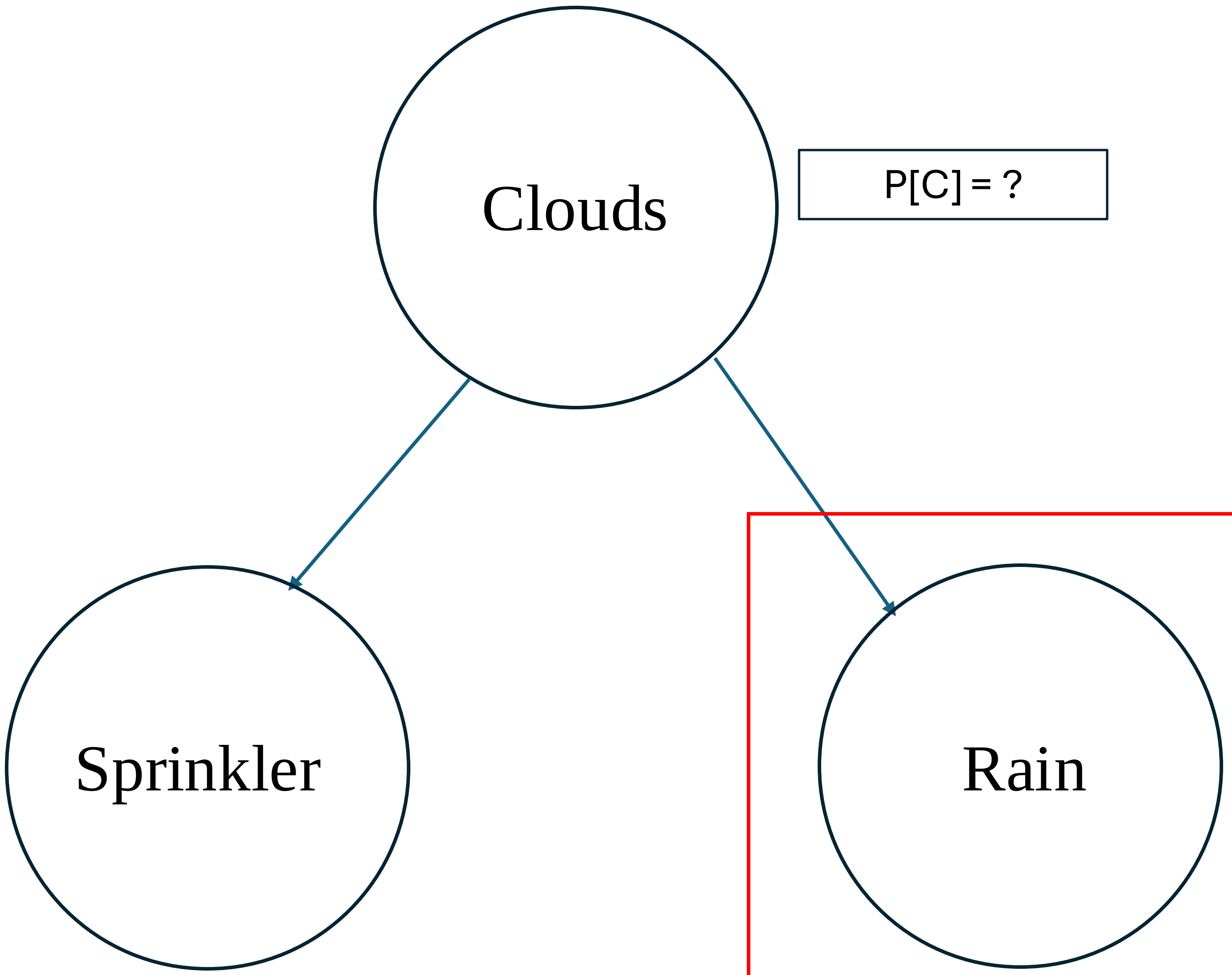
(a Bayes net can encode a causal relationship based on prior knowledge.)

If you are interested in learning a structure,
Chow-Liu Algorithm find the optimal T (first / second order dependence)
T = $argmin_T \; KL\,(P||\,T)$, where P is a true distribution.

Q: what is the true distribution? how can we compute this?

+ a true density can be represented by the network encoding the full dependence like
P[X1] P[X2|X1] P[X3|X2,X1] ....P[Xn|X1X2,...Xn-1]

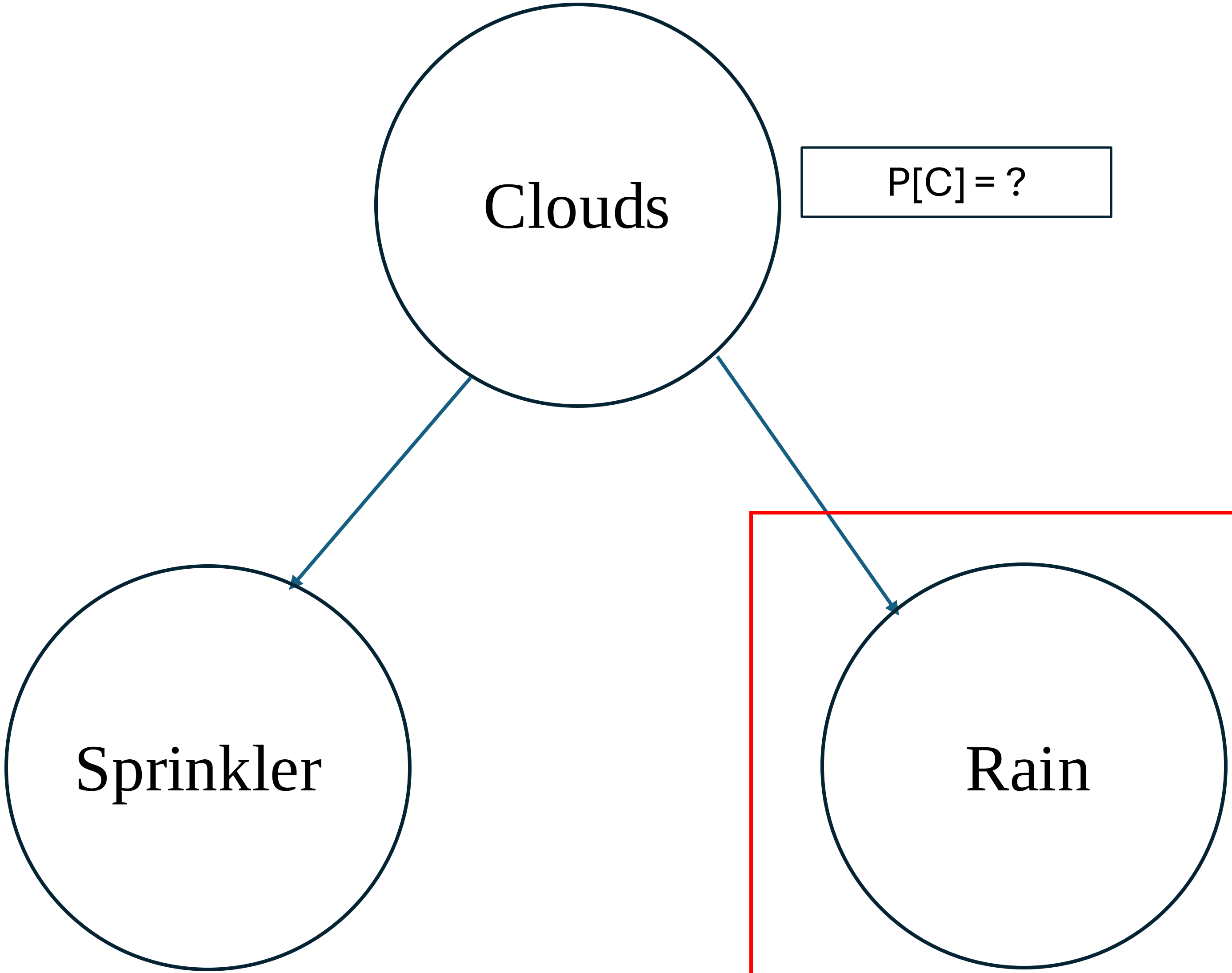
When a structure is known and data is fully observed,
we need to estimate CPTs (Filling up the Conditional Probability Table).



C	P[S|C]
T	?
F	?

C	P[R|C]
T	?
F	?

When a structure is known and data is fully observed,
we need to estimate CPTs (Conditional Probability Table).



P[C] = ?

C	P[S]
T	?
F	?

C	P[R]
T	? p
F	? q

MLE estimation for p:
when data is fully observed.

$$p(D) = \prod_{n=1}^{N} P(x_n)$$

$$\log P(D) = \sum_{n=1}^{N} \log P(x_n)$$

$$= \sum_{n=1}^{N} \log P[C(x_n)] + \log P[S(x_n)|C(x_n)] \boxed{+ \log P[R(x_n)|C(x_n)]}$$

the four possible cases of R and C
: $p\ 1-p, q, 1-q$

$$\frac{\partial \log P(D)}{\partial p} = N_p \cdot 1/P - N_n \cdot 1/(1-P) = 0$$

$$\boxed{P = \frac{N_p}{N_p + N_n}}$$

- N_P: # samples cloud + & rain +
- N_n: # samples cloud − & rain +

This is just a sample mean (Bernoulli R.V) what if data is partially observed?

Suppose we forgot to collect **cloud** information.

Can we still estimate p *or* q : P[rain| cloud +] or P[rain| cloud -]?

- MLE when data is partially observed.

$$\log P(D|\theta, p, q) = \sum_{n=1}^{N} \log \sum_{C_i} P(S_i, R_i, C_i|\theta, p, q)$$

marginalization over unobserved variables

- MLE when data is fully observed (complete MLE)

$$\log P(D|\theta, p, q)) = \sum_{n=1}^{N} \log P(S_i, R_i, C_i|\theta, p, q)$$

Q: which one is easier in computing MLE ?

$$\log P(D|\theta, p, q)) = \sum_{n=1}^{N} \log P(S_i, R_i, C_i|\theta, p, q)$$
$$= \sum_{n=1}^{N} \log P(C_i) + \log P(S_i|C_i) + \log P(R_i|C_i)$$

$$\log P(D|\theta, p, q) = \sum_{n=1}^{N} \log \sum_{C_i} P(S_i, R_i, C_i|\theta, p, q)$$
$$= \sum_{n=1}^{N} \log(P(C_i+) + P(S_i|C_i+) + P(R_i|C_i+)$$
$$+ P(C_i-) + P(S_i|C_i-) + P(R_i|C_i-))$$

hard to optimize (parameters are interdependent)!

It’s hard to get a closed form to compute MLE,
when there are missing variables.
(log \sum is not favorable; can we change it to \sum log like the complete case?)

Jensen's Inequality: (both expectation are finite)

$E[\log X] \leq \log E[X]$

$$\log P(D|\theta,p,q) = \sum_{n=1}^{N} \log \sum_{C_i} P(S_i, R_i, C_i|\theta,p,q)$$

- hard to optimize!

$$= \sum_{n=1}^{N} \log \sum_{C_i} \frac{P(S_i, R_i, C_i|\theta,p,q)q(C_i)}{q(C_i)}$$

$$= \sum_{n=1}^{N} \log E[\frac{P(S_i, R_i, C_i|\theta,p,q)}{q(C_i)}]$$

$$\geq \sum_{n=1}^{N} E[\log(\frac{P(S_i, R_i, C_i|\theta,p,q)}{q(C_i)})]$$

- by Jensen's Inequality

$$\geq \sum_{n=1}^{N} \underline{E[\log P(S_i, R_i, C_i|\theta,p,q)]} + H(q(C_i))$$

- easy to optimize!

expectation over

$q(C_i)$: *any arbitrary density for* C_i

gives a lower bound.

Q: how can we make the lower bound tight?

+ q(Ci) ~ P (Ci | Si,Ri, θ, p, q) *then* the lower bound gets tight:

Q: What does the expectation of likelihood mean?
Likelihood is not a probability. It is a function of the parameters.
In the example below, the likelihood with one point data x_i is a function of $L(\mu, \sigma)$.

$$\mathcal{N}(x_i|\mu, \sigma^2) = \frac{1}{\sqrt{(2\pi\sigma^2)}} \exp\{\frac{-1}{2\sigma^2}(x_i - \mu)^2\}$$

What if x_i is unobserved and instead we know the density $x_i \sim$ p(x_i)?
we can measure the expectation treating μ and σ as constants.

$$E[f(X_i|\mu, \sigma)] = \sum_{x_i} f(x_i|\mu, \sigma)p(x_i)$$

Q: What density makes the lower bound tight?

$$\sum_{n=1}^{N} E[\log P(S_i, R_i, C_i|\theta, p, q)] + H(q(C_i))$$

$$= \sum_{n=1}^{N} \sum_{C_i} q(C_i) \log(\frac{P(S_i, R_i, C_i|\theta, p, q)}{q(C_i)})$$

$$= \sum_{n=1}^{N} \sum_{C_i} q(C_i) \log(\frac{P(C_i|S_i, R_i, \theta, p, q) P(S_i, R_i|\theta, p, q)}{q(C_i)})$$

$$= \sum_{n=1}^{N} (\sum_{C_i} q(C_i) \log \frac{P(C_i|S_i, R_i, \theta, p, q)}{q(C_i)} + \sum_{C_i} q(C_i) \log P(S_i, R_i|\theta, p, q))$$

$$\boxed{= \sum_{n=1}^{N} -KL(q(C_i)||P(C_i|S_i, R_i, \theta, p, q))} + \log P(S_i, R_i|\theta, p, q)$$

when $\mathrm{q(C_i)} = P(C_i|S_i, R_i, \theta, p, q)$

the lower bound becomes equal to log $\mathrm{P}(S_i, R_i|\theta, p, q)$

$$\log P(D|\theta,p,q) = \sum_{n=1}^{N} \log \sum_{C_i} P(S_i, R_i, C_i|\theta,p,q)$$

$$= \sum_{n=1}^{N} \log \sum_{C_i} \frac{P(S_i, R_i, C_i|\theta,p,q)q(C_i)}{q(C_i)}$$

$$= \sum_{n=1}^{N} \log E[\frac{P(S_i, R_i, C_i|\theta,p,q)}{q(C_i)}]$$

$$\geq \sum_{n=1}^{N} E[\log(\frac{P(S_i, R_i, C_i|\theta,p,q)}{q(C_i)})]$$

$$\geq \sum_{n=1}^{N} \underline{E[\log P(S_i, R_i, C_i|\theta,p,q)]} + H(q(C_i))$$

+ The final goal is to maximize log P[D| θ, p, q].

By maximizing lower bound we approaches to the maximal point of P[D| θ, p, q] step by step.

expectation over $q(C_i)$: $P(C_i | S_i, R_i, \theta, p, q)$

what is the next ?

we do maximization over θ, p, q. Why?

EM Algorithm
to compute $argmax_{\theta,p,q} P(D|\theta, p, q)$ for the example of cloud, rain, sprinkler.

1. start with arbitrary parameters: $\theta, p,$ q
2. compute $P(C_i \,|S_i, R_i, \theta, p, q)$ for $\forall\, i$
3. **E step**: compute $\sum_{n=1}^{N} E[\log P(S_i, R_i, C_i|\theta, p, q)]$
4. **M step**: Update $\theta, p, q = \; argmax_{\theta,p,q} \; \sum_{n=1}^{N} E[\log P(S_i, R_i, C_i|\theta, p, q)]$
5. go back to step 2.

EM Algorithm
to compute $argmax_{\theta,p,q} \log P(D|\theta,p,q)$ for the example of cloud, rain, sprinkler.

1. start with arbitrary parameters: θ, p, q
2. compute $P(C_i\,|S_i, R_i, \theta, p, q)$ for $\forall\, i$
3. E step: compute $\sum_{n=1}^{N} E[\log P(S_i, R_i, C_i|\theta, p, q)] \quad = \mathrm{Q}\,(\theta, \theta^t)$

   Auxiliary Function

   + $\mathrm{Q}\,(\theta, \theta^t)$ is the expectation of log likelihood.
   the expectation is computed based on the density
   $P(C_i\,|S_i, R_i, \theta, p, q)$
4. S step: Update $\theta, p, q = argmax_{\theta,p,q} \sum_{n=1}^{N} E[\log P(S_i, R_i, C_i|\theta, p, q)]$
5. go back to step 2.

Auxiliary Function $Q\,(\theta, \theta^t) = \sum_{n=1}^{N} E[\log P(S_i, R_i, C_i | \underbrace{\theta, p, q}_{\theta})]$

expectation $P(C_i\,|S_i, R_i, \theta^t)$

- $Q\,(\theta^t, \theta^t) = \sum_i \log P[S_i, R_i\,|\theta^t]$
- $Q\,(\theta^{t+1}, \theta^t) \geq Q\,(\theta^t, \theta^t)$
- $Q\,(\theta^{t+1}, \theta^{t+1}) \geq Q\,(\theta^{t+1}, \theta^t)$
- $Q\,(\theta^{t+1}, \theta^{t+1}) = \sum_i \log P[S_i, R_i\,|\theta^{t+1}]$

EM monotonically increases the observed data log likelihood!

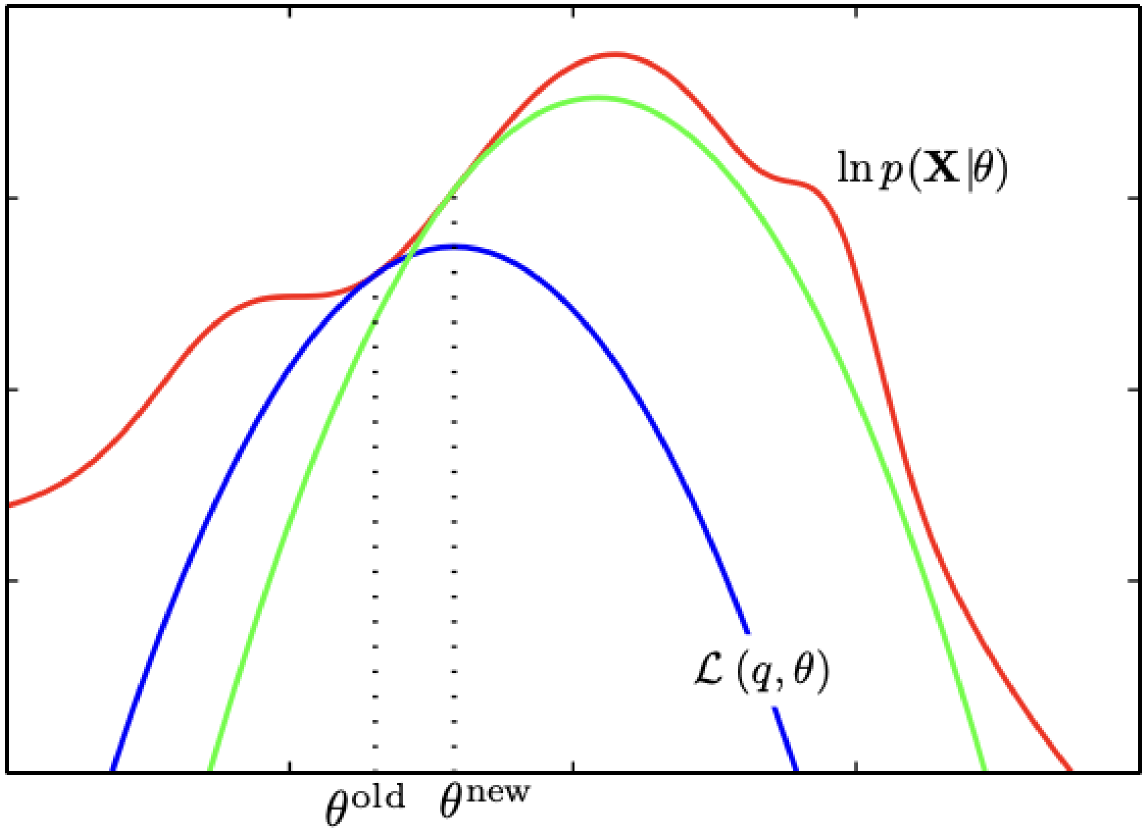
- $\sum_i \log P[S_i, R_i\,|\theta^t] \leq (\theta^{t+1}, \theta^t) \leq Q\,(\theta^{t+1}, \theta^{t+1}) = \sum_i \log P[S_i, R_i\,|\theta^{t+1}]$

EM algorithm finds a local minimum.

$\log P(X|\theta)$ is often non – convex or non – concave.

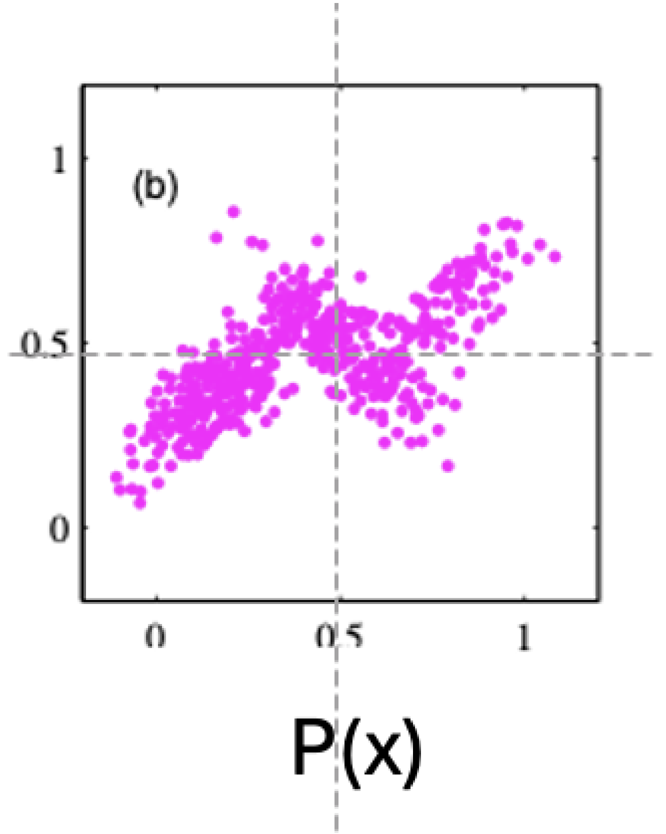
Figure 9.14 The EM algorithm involves alternately computing a lower bound on the log likelihood for the current parameter values and then maximizing this bound to obtain the new parameter values. See the text for a full discussion.

From textbook Bishop

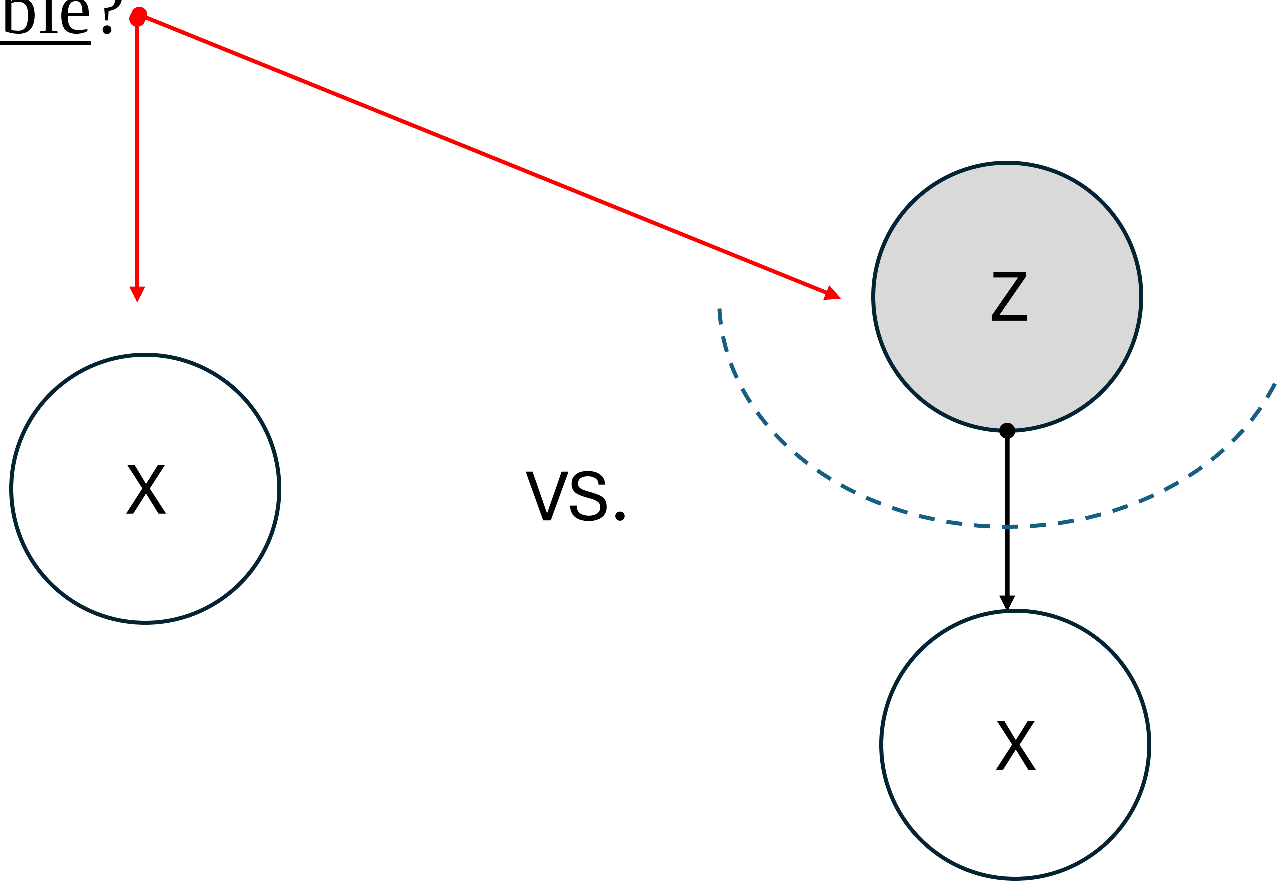


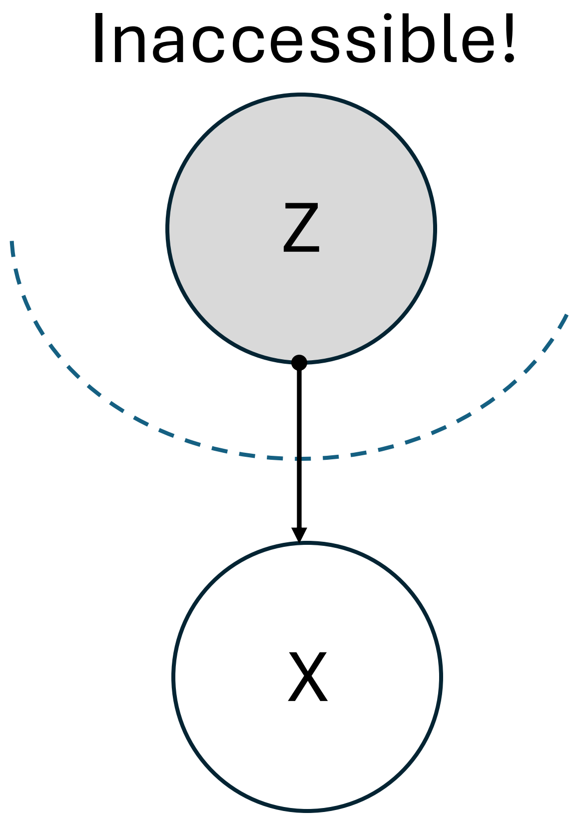
EM for Gaussian Mixture Models (GMM)

Data often has multiple modalities.
Which model is efficient and scalable?
Is Z accessible?



From textbook Bishop Fig. 9.5



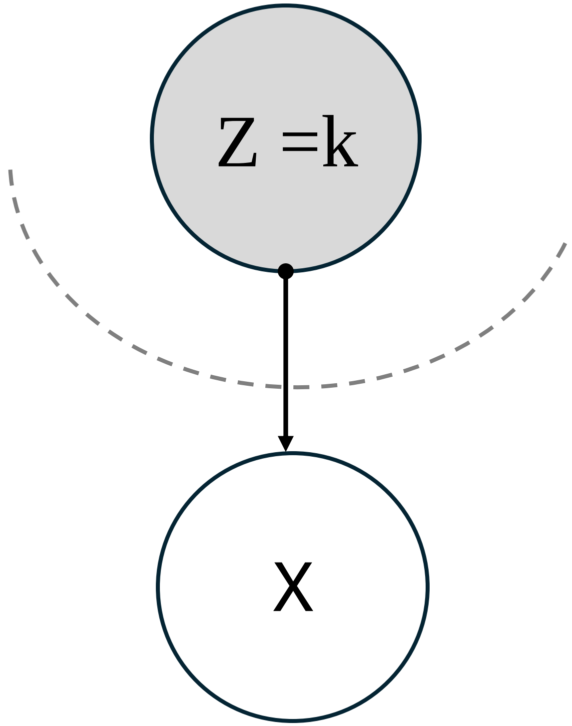


Q: What parameters do we need to define the model?

Q: How could we learn the parameters from data?

Gaussian Mixture Modeling (GMM)

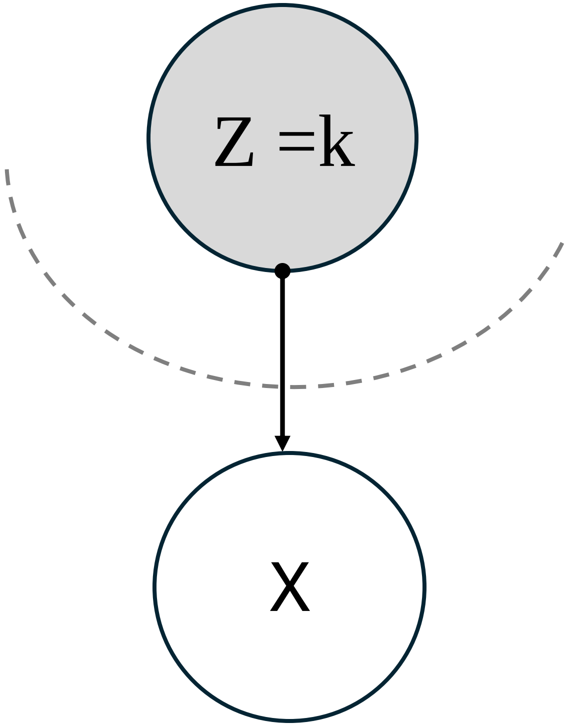
Bayesian Network Representation of GMM



- $P(Z = k) = \pi_k$
- $P(X|Z = k) = \mathcal{N}(\mu_k, \Sigma_k)$

Q: $P[X]$?

Bayesian Network Representation of GMM

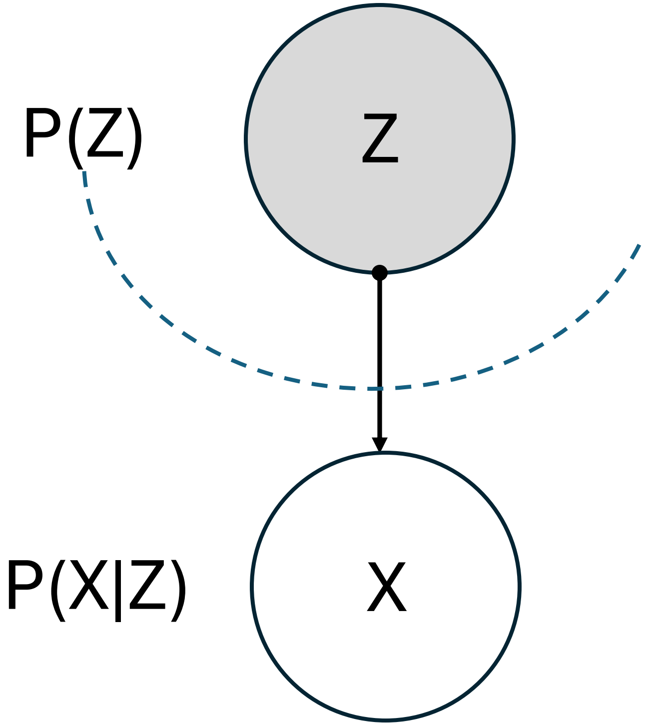


- $P(Z = k) = \pi_k$

- $P(X|Z = k) = \mathcal{N}(\mu_k, \Sigma_k)$

$$P(X) = \sum_{Z=k} P(X, Z) = \sum_{Z=k} P(Z = k)P(X|Z = k)$$
$$= \sum_{Z=k} \pi_j \mathcal{N}(\mu_k, \Sigma_k)$$

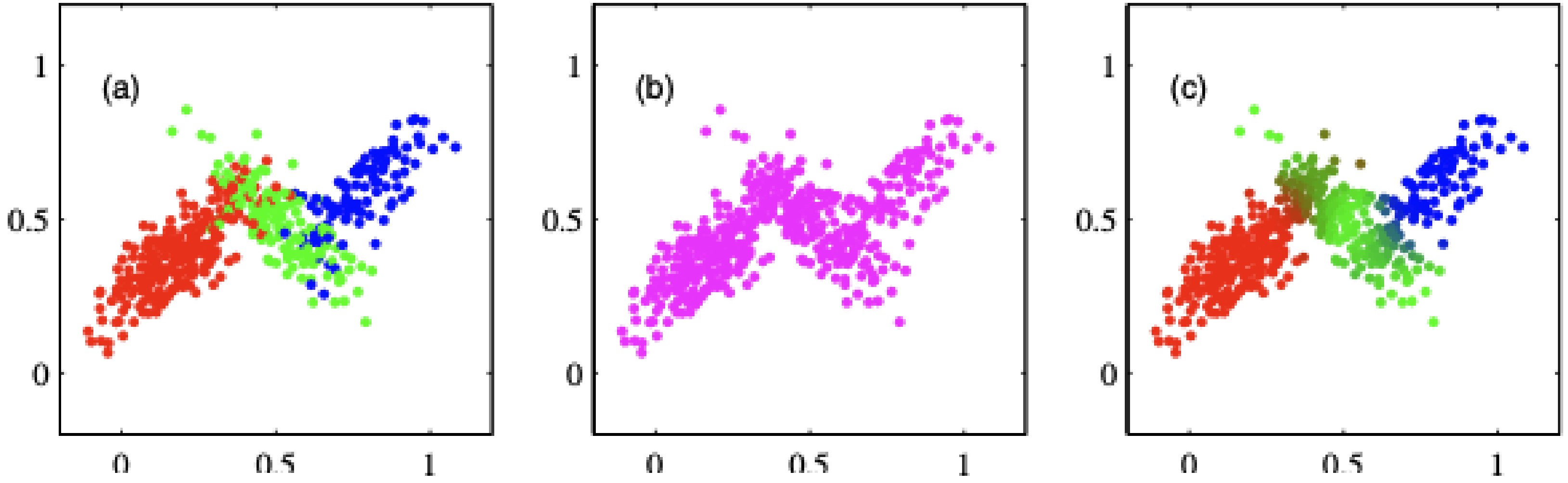
[Bayesian Network Representation of Mixture Model]



Once we set this model,
we could compute the followings.

- P(X)
- P(Z=k |X) * valuable information
  * prediction about invisible factors
  * useful for clustering.

In reality, we observed only x, but once we learn the parameters we we can <u>simulate</u> data $P(Z,X)$ or compute the posterior $P[Z|X]$.



P(z,x) P(x) P(z|x)

From textbook Bishop Fig. 9.5

Learning the parameters : π, μ, Σ
We cannot access Z, how can we learn the parameters?
What method?

EM for GMM

$$\sum_{n=1}^{N} \log P(x_n|\theta) = \sum_{n=1}^{N} \log \sum_{z_n=k} P(x_n|z_{n=k}, \theta) P(z_n = k|\theta)$$

$$\geq \sum_{n=1}^{N} E[\log P(z_n, x_n|\theta)] + H(P(z_n|x_n, \theta))$$

[1] E step: at the current parameters $\theta(t)$: $\pi(t)$, $\mu(t), \Sigma(t)$

we compute the posterior

$$\gamma_{nk}(\theta_t) = P[z_n = k|x_n, \theta_t] = \frac{P(z_n = k, x_n|\theta_t)}{P(x_n|\theta_t)} = \frac{P(x_n|z_n = k, \theta_t) P(z_n = k|\theta_t)}{\sum_{z_n=k} P(x_n|z_n = k, \theta_t) P(z_n = k|\theta_t)}$$

- **responsibility** that cluster K takes for data point n

EM for GMM

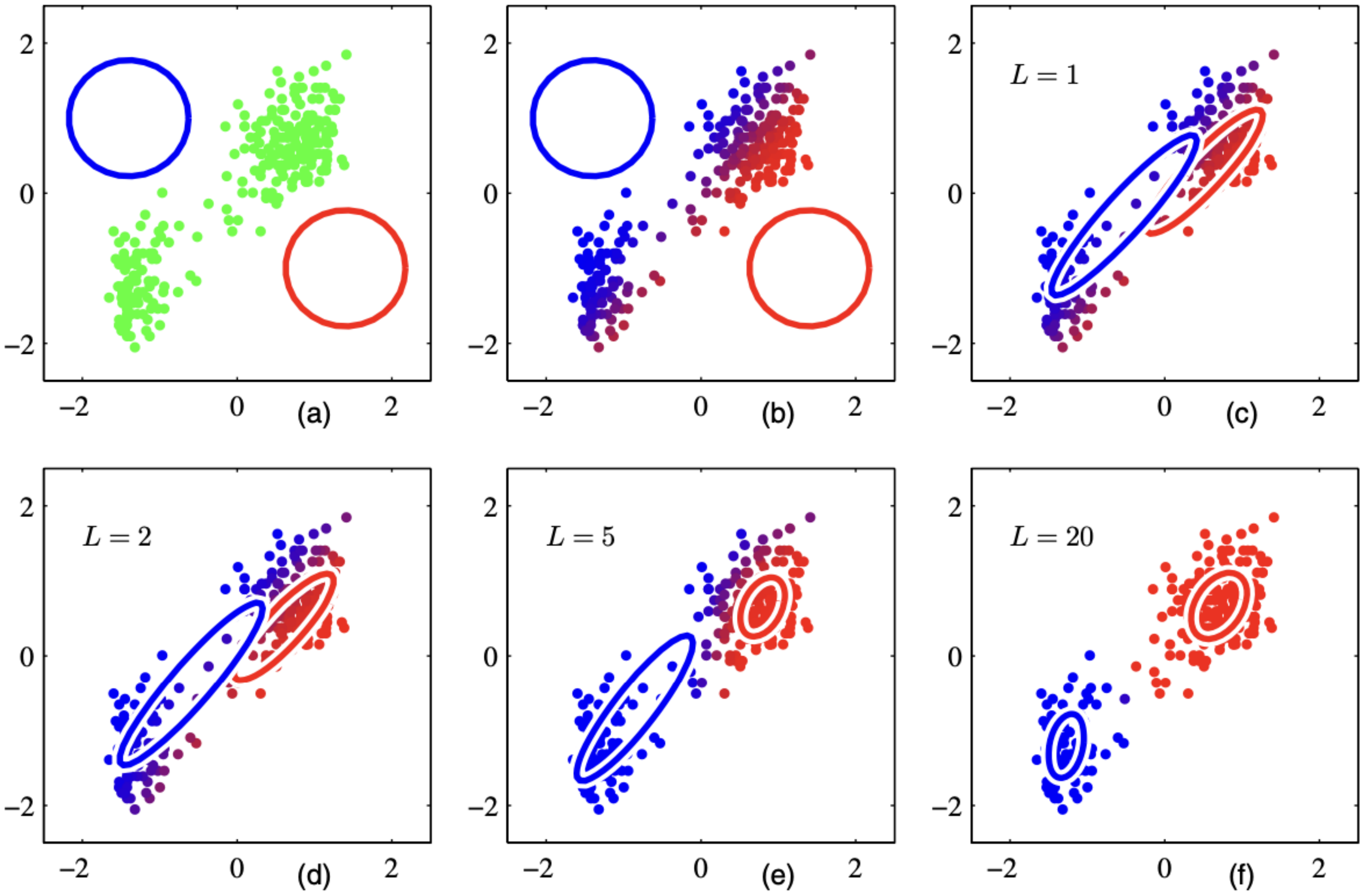
[1] E step: at current $\theta(t)$: $\pi(t)$, $\mu(t), \Sigma(t)$
compute $\gamma_{nk}(\theta(t))$

[2] M step: $\theta(t+1) = \arg\max_{\theta} \sum_{n=1}^{N} E[\log P(z_n, x_n|\theta)]$, when E over $\gamma_{nk}(\theta)$

- $\pi_k(t+1) = \frac{\sum_n P(z_n = k|x_n, \theta_t)}{N} = \frac{\sum_n \gamma_{nk}(\theta_t)}{N} = \frac{\gamma_k(\theta_t)}{N}$
- $\mu_k(t+1) = \frac{\sum_n P(z_n = k|x_n, \theta_t) \cdot x_n}{\sum_n P(z_n = k|x_n, \theta_t)} = \frac{\sum_n \gamma_{nk} \cdot x_n}{\gamma_k(\theta_t)}$
- $\Sigma(t+1) = \frac{\sum_n \gamma_{nk}(x_n - \mu_k(t+1))(x_n - \mu_k(t+1))^t}{\gamma_k(\theta_t)}$

Illustration of EM algorithm for GMM

P[Z=k | X, θ]



From textbook Bishop Fig. 9.8

Let' s go back to the problem of
Learning the parameters of Bayesian network.

what if data is partially observed?
Suppose we forgot to collect cloud information.
Can we still estimate p *or* q : P[rain| cloud +] or P[rain| cloud -]?

EM for learning a Bayesian Network (The cloud information is unknown.)

- E step : compute $\gamma_{nk}\,(t) = \; P[Cloud\ (n) = k \mid Rain\ (n)\ \ and\ S\ (n)\ ]$

- M step: update the parameters

- $P(C)(t+1) = \dfrac{\sum_{n=1}^{N} \gamma_{nk}(t)}{N}$

- $P(S:+, R:+|C:+)(t+1) = \dfrac{\sum_{n=1}^{N} \gamma_{nk}\delta(x_n = S:+, R:+)}{\sum_{n=1}^{N} \gamma_{nk}}$

Application of Bayesian Net
“Bonaparte”

(2010 air disaster in Tripoli / Malaysia Airlines flight MH17 in the Ukraine in 2014)

Example of Bayesian Network in the Real World

- Bonaparte DNA matching Software

  A pedigree of the family is converged to a Bayesian Network.

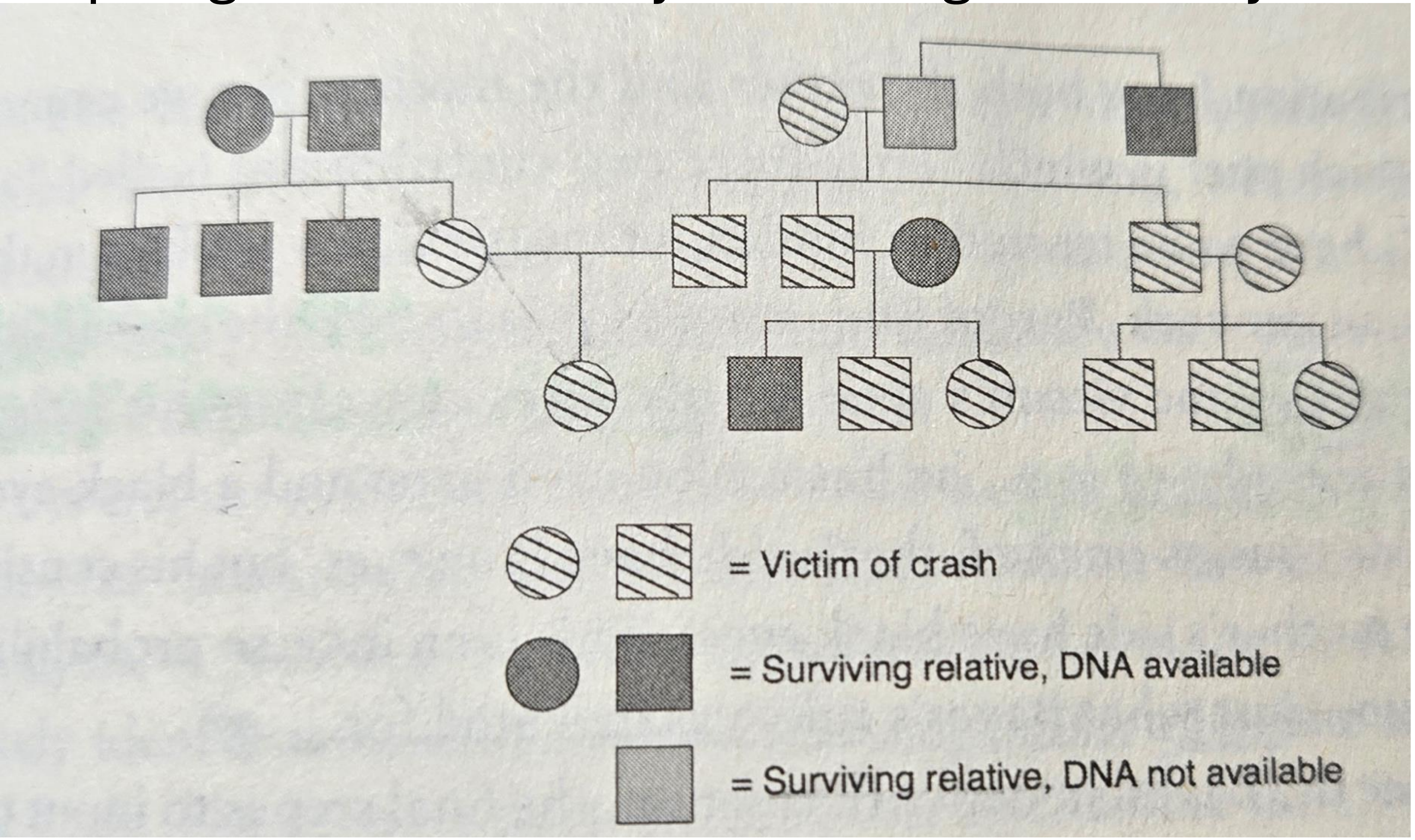


Figure 3.7 from "The book of why" by Judea Peral

+ By building a Bayesian Network and CPT, genetic contexts can be considered effectively from ancestors to descendants.

Q: what will be the benefit to use a Bayesian network for DAN matching instead of using a direct comparison?