Learning

- Parameters

- Structure

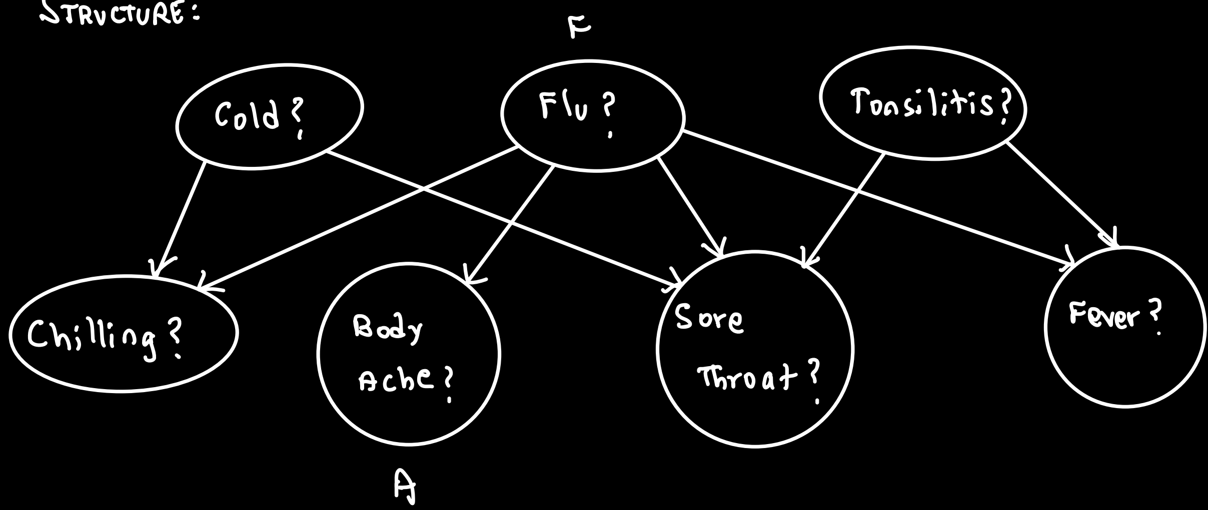
Supervised vs Unsupervised

↳ Query-oriented

↳ model-oriented

LEARNING PARAMETERS:

STRUCTURE:



CPTs can also be estimated from medical records of previous patients

Case	*Cold?*	*Flu?*	*Tonsillitis?*	*Chilling?*	*Bodyache?*	*Sorethroat?*	*Fever?*
1	true	false	?	true	false	false	false
2	false	true	false	true	true	false	true
3	?	?	true	false	?	true	false
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

→ Incomplete example

Complete example

examples

Data: Complete, not complete

F	A	
t	t	$\theta_{a\|f}$
t	f	$\theta_{\bar{a}\|f}$
f	t	$\theta_{a\|\bar{f}}$
f	f	$\theta_{\bar{a}\|\bar{f}}$

$\theta_{a|f}$: Pr (a|f)

$\theta_{\bar{a}|\bar{f}}$: Pr $(\bar{a}|\bar{f})$

Parameters

MAXIMUM LIKELIHOOD:

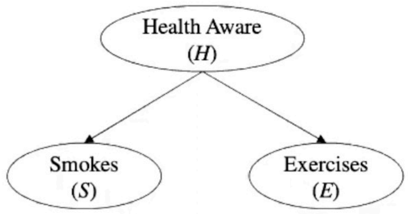
Parameters 1 => BN 1 => Pr_1 (data) = Pr (data | parameters$_1$)

Parameters 2 => BN2 => Pr_2 (data) = Pr (data | parameters$_2$)

$e_1 \; e_2 \ldots \; e_n$

LIKELIHOOD => $Pr_1(e_1)\, Pr_1(e_2) \ldots Pr_1(e_n)$ Score 1

=> $Pr_2(e_1)\, Pr_2(e_2) \ldots Pr_2(e_n)$ Score 2



Case	H	S	E
1	T	F	T
2	T	F	T
3	F	T	F
4	F	F	T
5	T	F	F
6	T	F	T
7	F	F	F
8	T	F	T
9	T	F	T
10	F	F	T
11	T	F	T
12	T	T	T
13	T	F	T
14	T	T	T
15	T	F	T
16	T	F	T

H	S	E	$Pr_{\mathcal{D}}(.)$
T	T	T	2/16
T	T	F	0/16
T	F	T	9/16
T	F	F	1/16
F	T	T	0/16
F	T	F	1/16
F	F	T	2/16
F	F	F	1/16

↑ DATASET (Complete)

↑ Empirical Distribution

H	
t	θ_h
f	$\theta_{\bar{h}}$

H	S	
t	t	$\theta_{s\|h}$
t	f	$\theta_{\bar{s}\|h}$
f	t	$\theta_{s\|\bar{h}}$
f	f	$\theta_{\bar{s}\|\bar{h}}$

Similarly for H, E

(BAYES CONDITIONING)

$$\theta_{\bar{s}|h} = Pr(\bar{s}|h) = \frac{Pr(\bar{s}, h)}{Pr(h)}$$

$$= \frac{Pr_D(\bar{s}, h)}{Pr_D(h)}$$

$$= \frac{10/16}{12/16} = 5/6$$

PARAMETER ESTIMATE.

WHAT IF INCOMPLETE DATA?

H	S	E
⋮		
T	F	?
⋮		

EM: [EXPECTATION MAXIMIZATION]

$CPT_1 \rightsquigarrow BN_1 \rightarrow Pr_1(\cdot)$

T F ?

T F T

T F F

$x = Pr_1(E = T \mid H = T, S = F)$

$y = Pr_1(E = F \mid H = T, S = F)$

Fill the x, y in

Recompute CPT

$CPT_2 \rightarrow BN_2 \rightarrow Pr_2(\cdot)$

⋮

$CPT_3 \rightarrow BN_3 \rightarrow Pr_3(\cdot)$

⋮

Converges

Likelihood never decreases.

↳ Increases or remains same.

Convergence speed $\propto \dfrac{1}{n_{\text{missing data}}}$

So θ_1 Inference

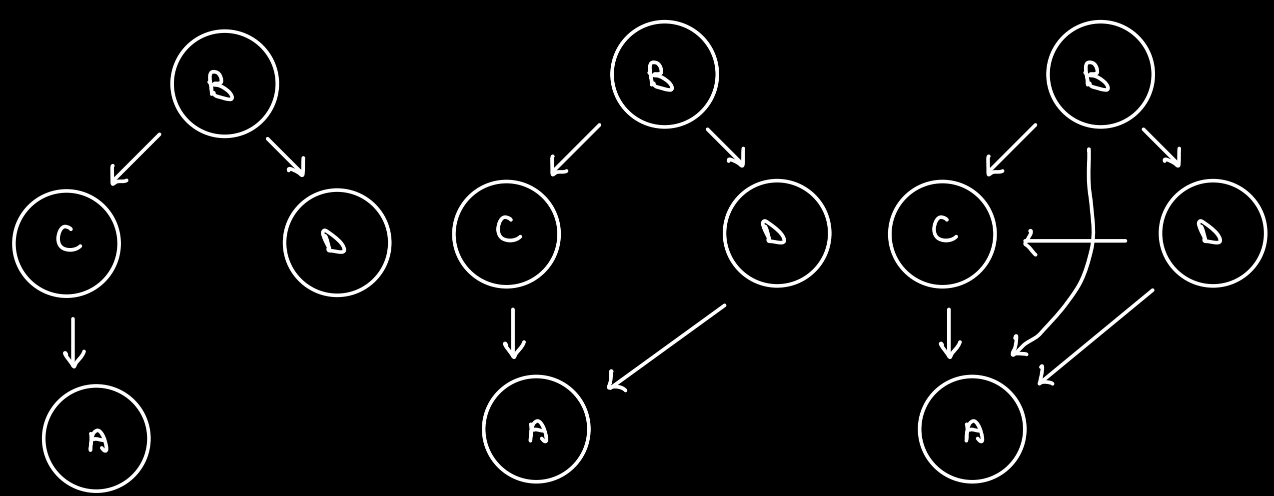
θ_2

θ_3

$\vdots$

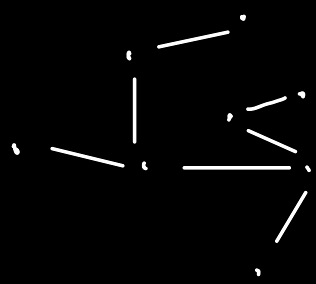
θ_n

LEARNING STRUCTURE:



Choose what gives best score.

1. LOCAL SEARCH METHODS:



Approximate

Fast

Add, remove, reverse an edge

2. SYSTEMATIC SEARCH METHODS:

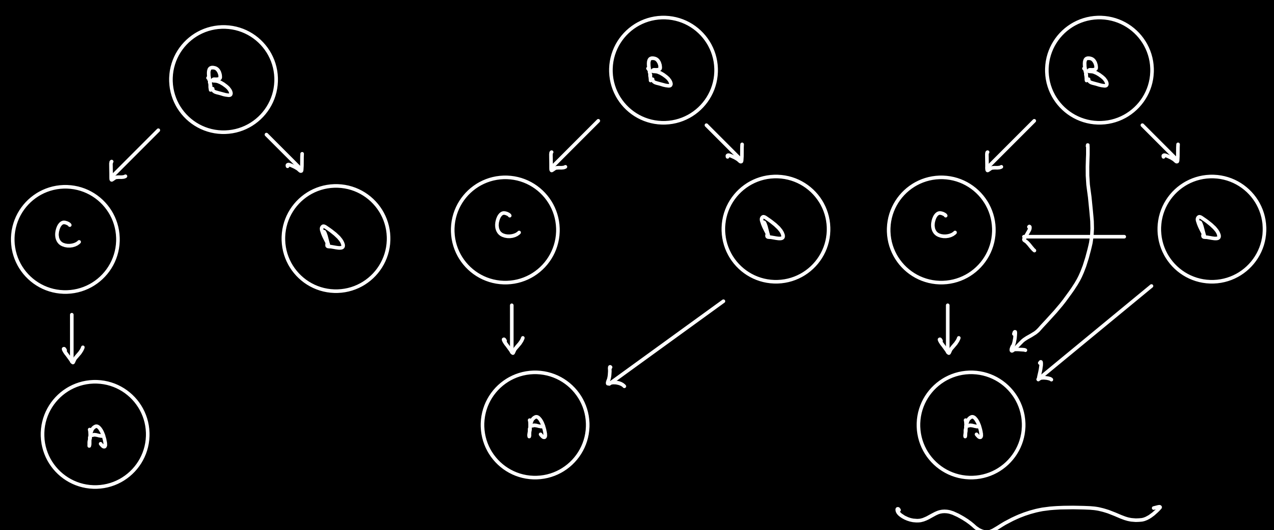
A* search

Guaranteed

Slower

WHY NOT USE MAXIMUM LIKELIHOOD?

Overfitting



always chose

Complete DAG

$$\begin{tikzcd} B \arrow[r] \arrow[rrr, bend left] \arrow[rr, bend right] & D \arrow[r] \arrow[rr, bend left] & C \arrow[r] & A \end{tikzcd}$$

X	Y
1	1.1
5	4.5
10	11
15	14.5
20	22

$y = ax + b$

Parameters

to fit perfectly,

we need

$y = ax^4 + bx^3 + cx^2 + dx + e$

OVERFITTING

* Not generalizing

SCORE : Likelihood + Penalty term

MDL Score

↳ depends on

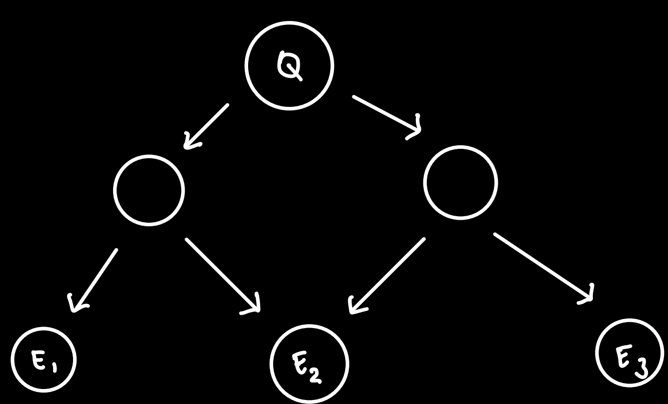
→ number of independent parameters

→ Size of dataset

MODEL - ORIENTED vs QUERY - ORIENTED LEARNING :

Unsupervised vs Supervised Learning

Un-labeled vs labeled data



Model - Oriented:

Learn Structure (can answer any query)

[What we did till now]

Query - Oriented:

Specific to one query.

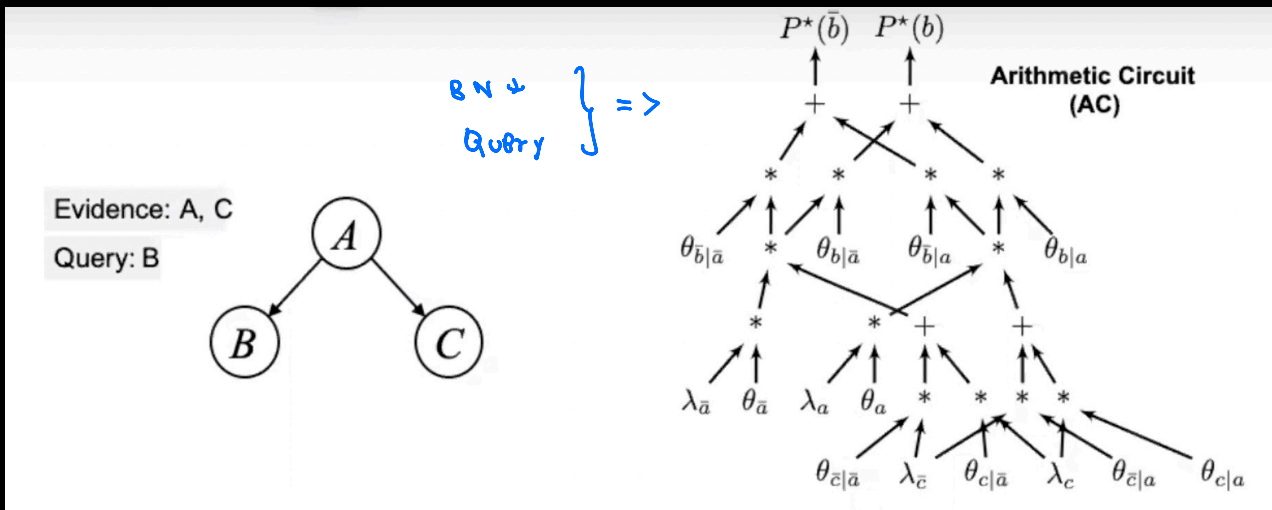
$$P(Q \mid E_1, E_2, E_3)$$

Query

QUERY - ORIENTED:

E_1	E_2	E_3	Q

labels



Boolean formula
+
Weights
$\Downarrow$
NNF Circuit
$\downarrow$
Convert NNF circuit to AC

$P \rightarrow$ Distribution on B (Query)

$\theta \rightarrow$ BN Parameters

A	λ_a	$\lambda_{\bar{a}}$
T	1	0
F	0	1
?	1	1

Evidence (Input)

$A = T, C = F$

$\lambda_a = 1$

$\lambda_{\bar{a}} = 0$

$\lambda_c = 0$

$\lambda_{\bar{c}} = 0$

$A = F$

$\lambda_a = 0$

$\lambda_{\bar{a}} = 1$

$\lambda_c = 1$

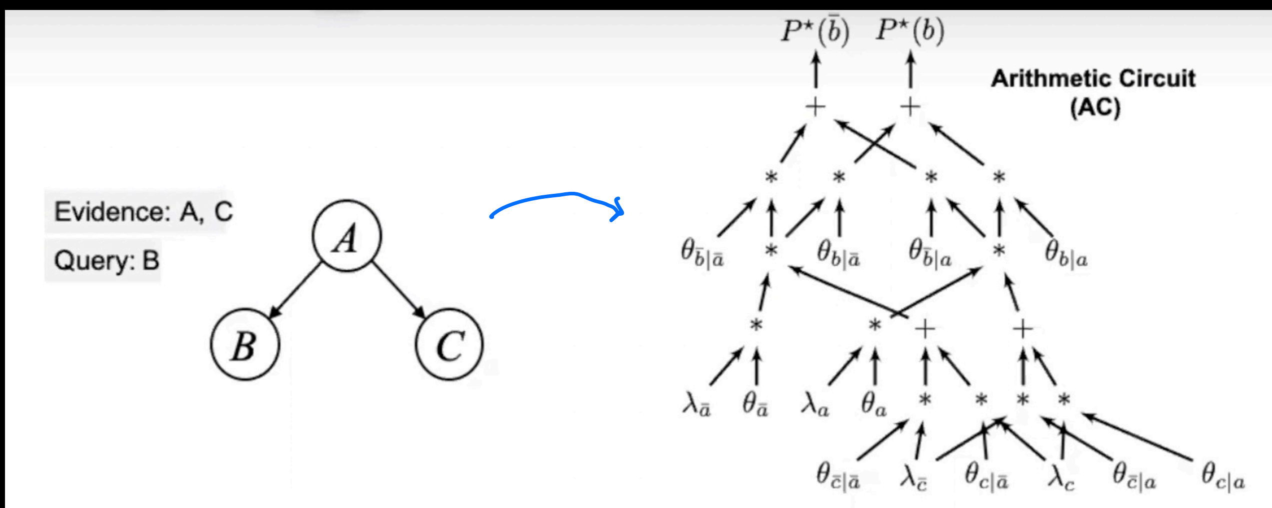
$\lambda_{\bar{c}} = 1$

n variables

d # values

ω tree width

$O(n \cdot d^{\omega})$



LABELED DATA

Input		Output
A	C	B
T	T	F
F	T	F
⋮	⋮	⋮
T	T	F

Loss function

(cross entropy)

$\underbrace{P(x) \quad Q(x)}$

number (how close they are)

model $\left\{ A = T,\ C = T \right. \Rightarrow Pr(B)$

$\boxed{P}$

Prediction

from data $\left\{ A = T,\ C = T \right. \Rightarrow$ Label

$\boxed{Q}$

B	
T	0
F	1

one - hot distribution

GRADIENT DESCENT is used to optimize loss function.

* TensorFlow
* Pytorch

CROSS ENTROPY:

$P(x)$: predictions

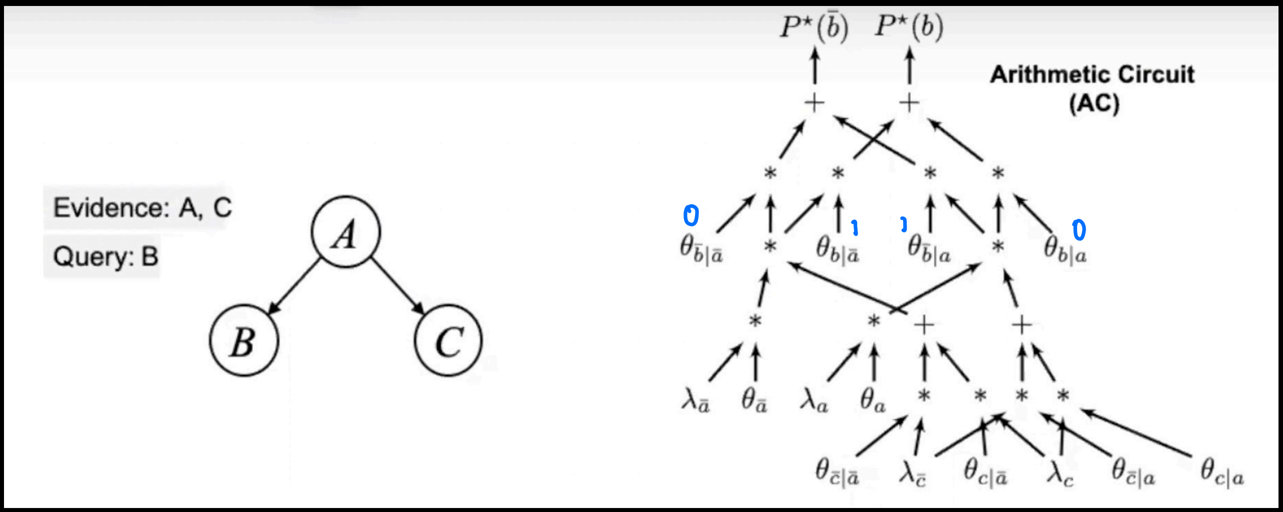
distribution

$Q(x)$: labels

$$CE: \sum_{x} Q(x) \cdot \log_2 (P(x))$$

LOSS FUNCTION

BACKGROUND KNOWLEDGE:



B = T iff A = F

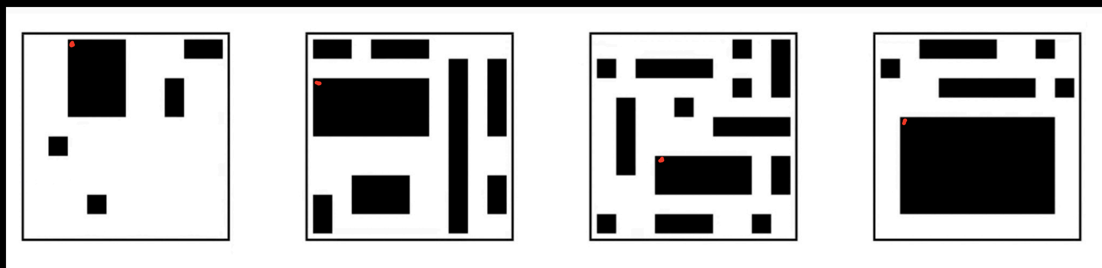
A	B	
t	t	0
t	f	1
f	t	1
f	f	0

Background Knowledge

FIND THE LARGEST RECTANGLE:

n

n



Rectangle:

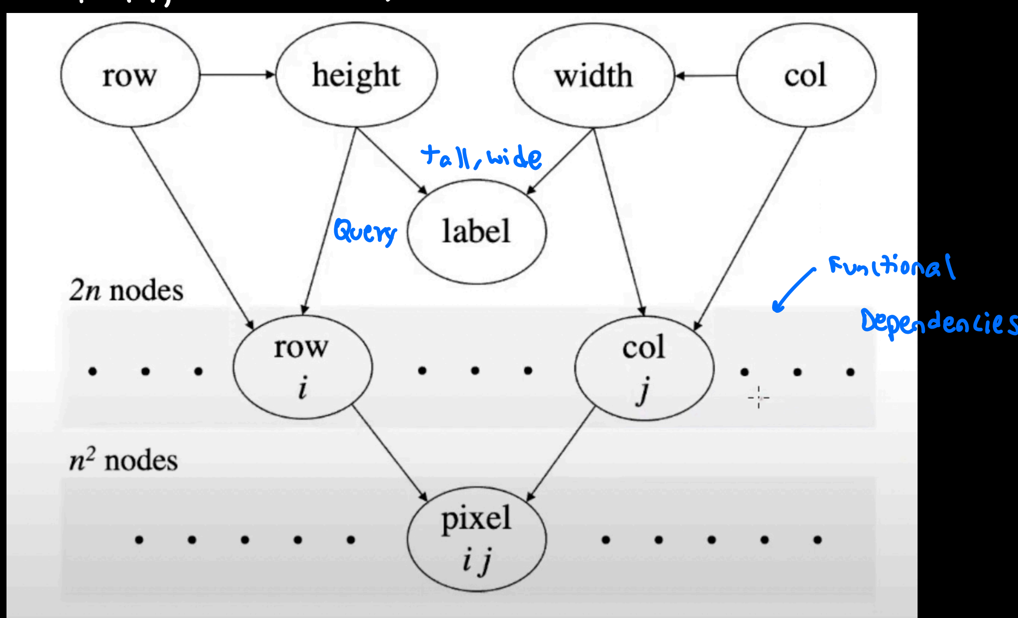
Upper left : row , col

Height

Width

Label : Tall or wide

1... n 1... n 1... n 1... n

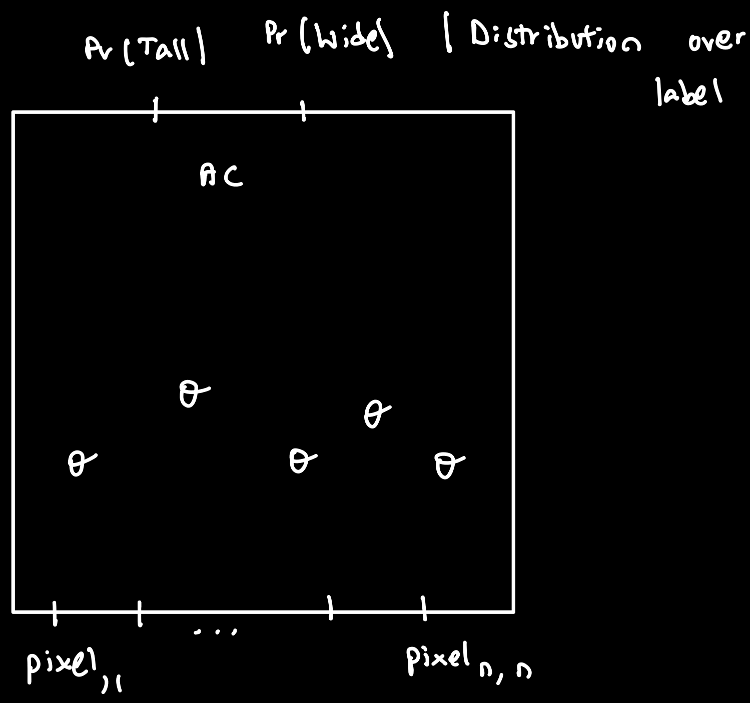


Evidence:

$row_1, row_2 \dots row_n$: True , False

$col_1 , col_2 \dots col_n$: True / False

$pixel_{1,1}$
$\quad\;\; {}_{row, col}$


Pr(Tall)
Pr(Wide)
| Distribution over label
AC
θ
θ
θ
θ
θ
...
pixel 1,1
pixel n,n