Bias, Variance and Error

Bias and Variance

given algorithm that outputs estimate $\hat{\theta}$ for θ , we define:

the <u>bias</u> of the estimator: $E[\hat{\theta}] - \theta$

the <u>variance</u> of estimator: $E[\ (\hat{\theta} - E[\hat{\theta}])^2\ ]$

e.g. $\hat{\theta}^{MLE}$ estimator for probability θ of heads, based on n independent coin flips

$$\hat{\theta}^{MLE} = \frac{\alpha_1}{\alpha_1 + \alpha_0}$$

what is its bias?

variance?

$$Var(\hat{\theta}^{MLE}) = \frac{\theta(1-\theta)}{n}$$

Bias and Variance

given algorithm that outputs estimate $\hat{\theta}$ for θ , we define:

the <u>bias</u> of the estimator: $E[\hat{\theta}] - \theta$

the <u>variance</u> of estimator: $E[\ (\hat{\theta} - E[\hat{\theta}])^2\ ]$

which estimator has higher bias? higher variance?

$$\hat{\theta}^{MLE} = \frac{\alpha_1}{\alpha_1 + \alpha_0}$$

$1/n$ $\hat{\theta}_n^{MLE}$ bias = 0

Var = $\frac{(1-\theta)\theta}{n}$

$$\hat{\theta}^{MAP} = \frac{\alpha_1 + \beta_1 - 1}{(\alpha_1 + \beta_1 - 1) + (\alpha_0 + \beta_0 - 1)}$$

$\hat{\theta}_n^{MAP}$ bias > 0 for finite n

var is less

$P(|X - E[X]| \geq t) = \frac{Var(X)}{t^2}$

Bias – Variance decomposition of error

Reading: Bishop chapter 9.1, 9.2

- Consider simple regression problem f:X→Y

$$y = f(x) + \varepsilon$$

$h(x) = y = w_0 + w_1 x_1$

deterministic

noise $N(0,\sigma)$

Define the expected prediction error:

$$E_D \left[\int_y \int_x (h(x) - f(x))^2 p(y|x) p(x) dy dx \right]$$

expectation over training D

learned estimate of f(x)

Sources of error

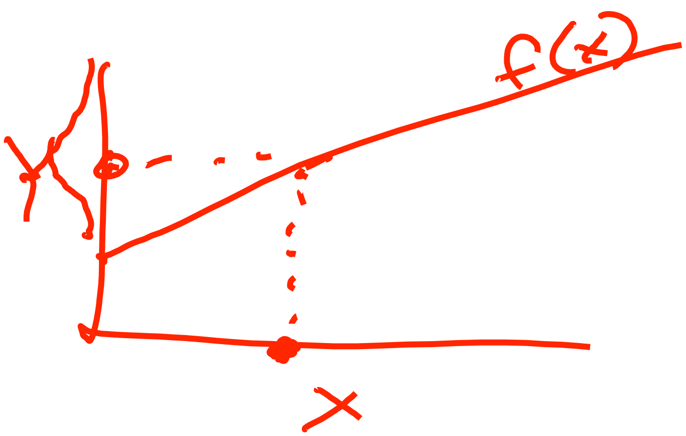
$$y = f(x) + \varepsilon$$

deterministic

noise $N(0,\sigma)$

What if we have perfect learner, infinite data?

- Our learned h(x) satisfies h(x)=f(x)
- Still have remaining, *unavoidable error*



σ^2

Sources of error

- What if we have only n training examples?
- What is our expected error
  - Taken over random training sets of size n, drawn from distribution $D=p(x,y)$

$$E_D\left[\int_y \int_x (h(x) - f(x))^2 p(y|x)p(x)dydx\right]$$

Sources of error

$y = f(x) + \varepsilon$

deterministic

noise $N(0,\sigma)$

$$E_D\left[\int_y \int_x (h(x) - f(x))^2 p(y|x)p(x)dydx\right]$$

$$= unavoidableError + bias^2 + variance$$

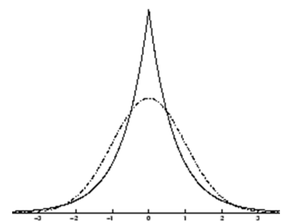
$$bias^2 = \int (E_D[h(x)] - f(x))^2 p(x)dx$$

$$variance = \int E_D[(h(x) - E_D[h(x)])^2]p(x)dx$$

L2 vs. L1 Regularization

$$W = \arg\max_W \ \ln P(W) + \sum_l \ln(P(Y^l|X^l; W))$$

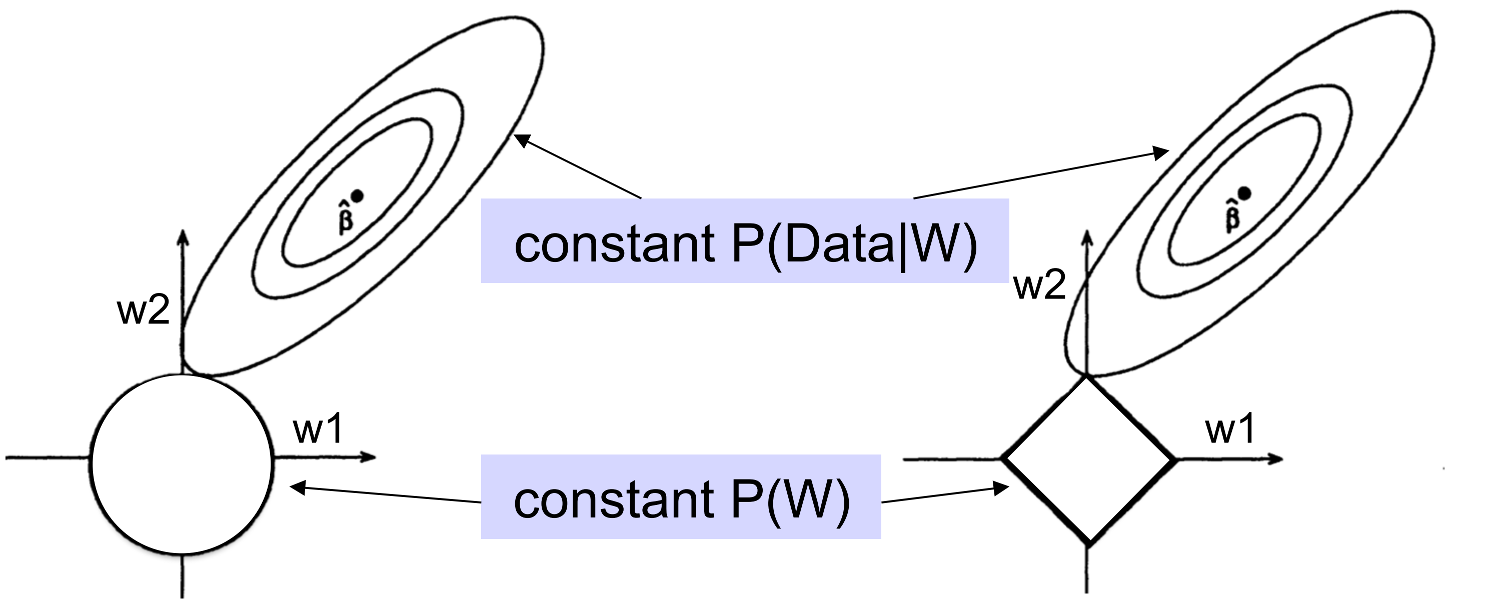
Gaussian P(W)
→ L2 regularization



Laplace P(W)
→ L1 regularization

$$\ln P(W) \propto \sum_i w_i^2$$

$$\ln P(W) \propto \sum_i |w_i|$$



Summary

- Bias of parameter estimators
- Variance of parameter estimators
- We can define analogous notions for estimators (learners) of functions
- Expected error in learned functions comes from
  - unavoidable error (invariant of training set size, due to noise)
  - bias (can be caused by incorrect modeling assumptions)
  - variance (decreases with training set size)
- MAP estimates generally more biased than MLE
  - but bias vanishes as training set size → ∞
- Regularization corresponds to producing MAP estimates
  - L2 / Gaussian prior / leads to smaller weights
  - L1 / Laplace prior / leads to fewer non-zero weights

Machine Learning 10-601

Tom M. Mitchell
Machine Learning Department
Carnegie Mellon University

February 18, 2015

Today:

- Graphical models
- Bayes Nets:
  - Representing distributions
  - Conditional independencies
  - Simple inference
  - Simple learning

Readings:

- Bishop chapter 8, through 8.2

Graphical Models

- Key Idea:
  - Conditional independence assumptions useful
  - but Naïve Bayes is extreme!
  - Graphical models express sets of conditional independence assumptions via graph structure
  - Graph structure plus associated parameters define *joint probability distribution over set of variables*

10-601

- Two types of graphical models:
  - Directed graphs (aka Bayesian Networks)
  - Undirected graphs (aka Markov Random Fields)

Graphical Models – Why Care?

- Among most important ML developments of the decade

- Graphical models allow combining:
  - Prior knowledge in form of dependencies/independencies
  - Prior knowledge in form of priors over parameters
  - Observed training data

- Principled and ~general methods for
  - Probabilistic inference
  - Learning

- Useful in practice
  - Diagnosis, help systems, text analysis, time series models, ...

Conditional Independence

Definition: X is conditionally independent of Y given Z, if the probability distribution governing X is independent of the value of Y, given the value of Z

$$(\forall i, j, k) P(X = x_i | Y = y_j, Z = z_k) = P(X = x_i | Z = z_k)$$

Which we often write $P(X|Y, Z) = P(X|Z)$

E.g., $P(Thunder|Rain, Lightning) = P(Thunder|Lightning)$

Marginal Independence

Definition: X is marginally independent of Y if

$$(\forall i, j) P(X = x_i, Y = y_j) = P(X = x_i) P(Y = y_j)$$

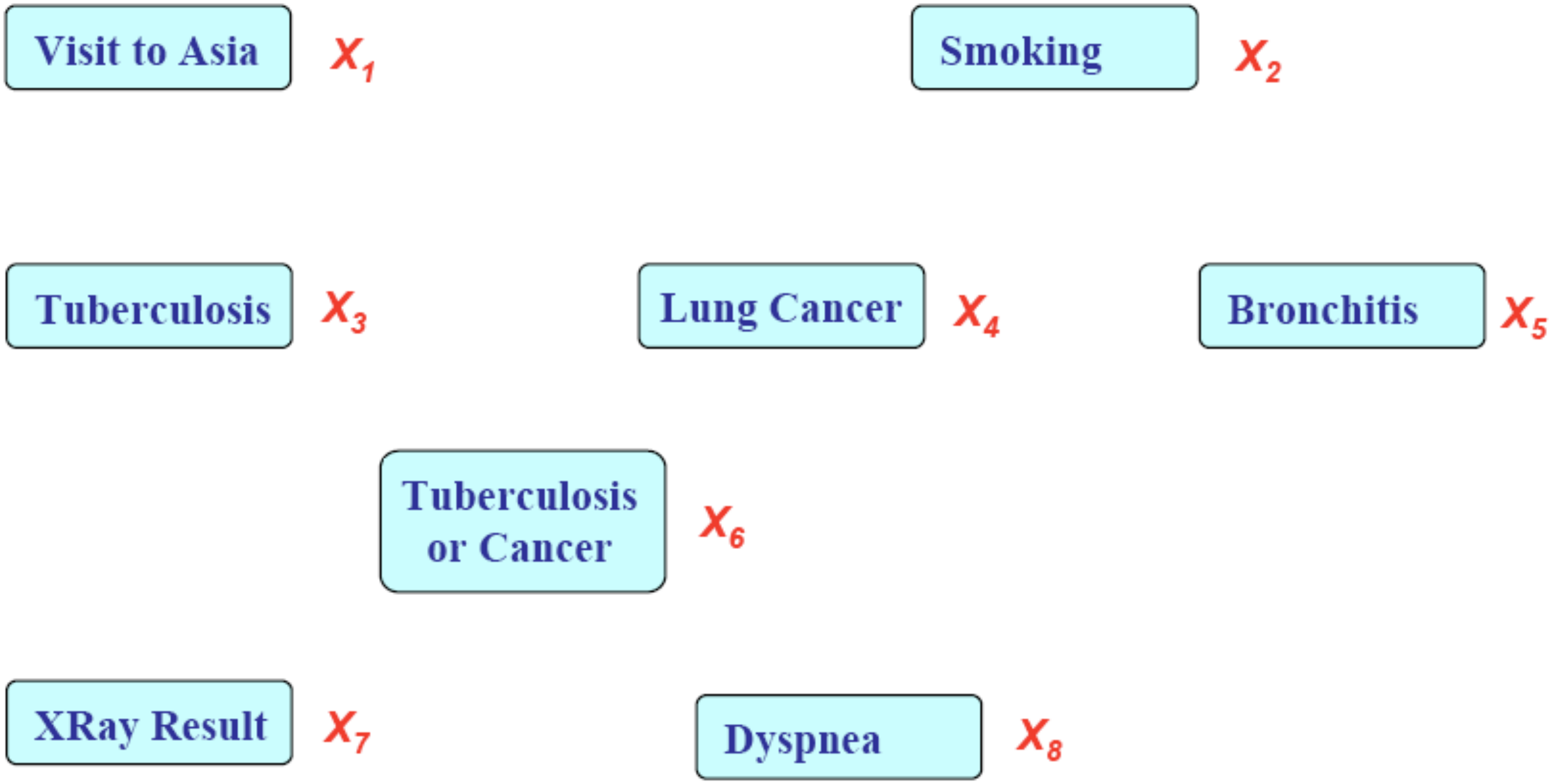
Equivalently, if

$$(\forall i, j) P(X = x_i | Y = y_j) = P(X = x_i)$$

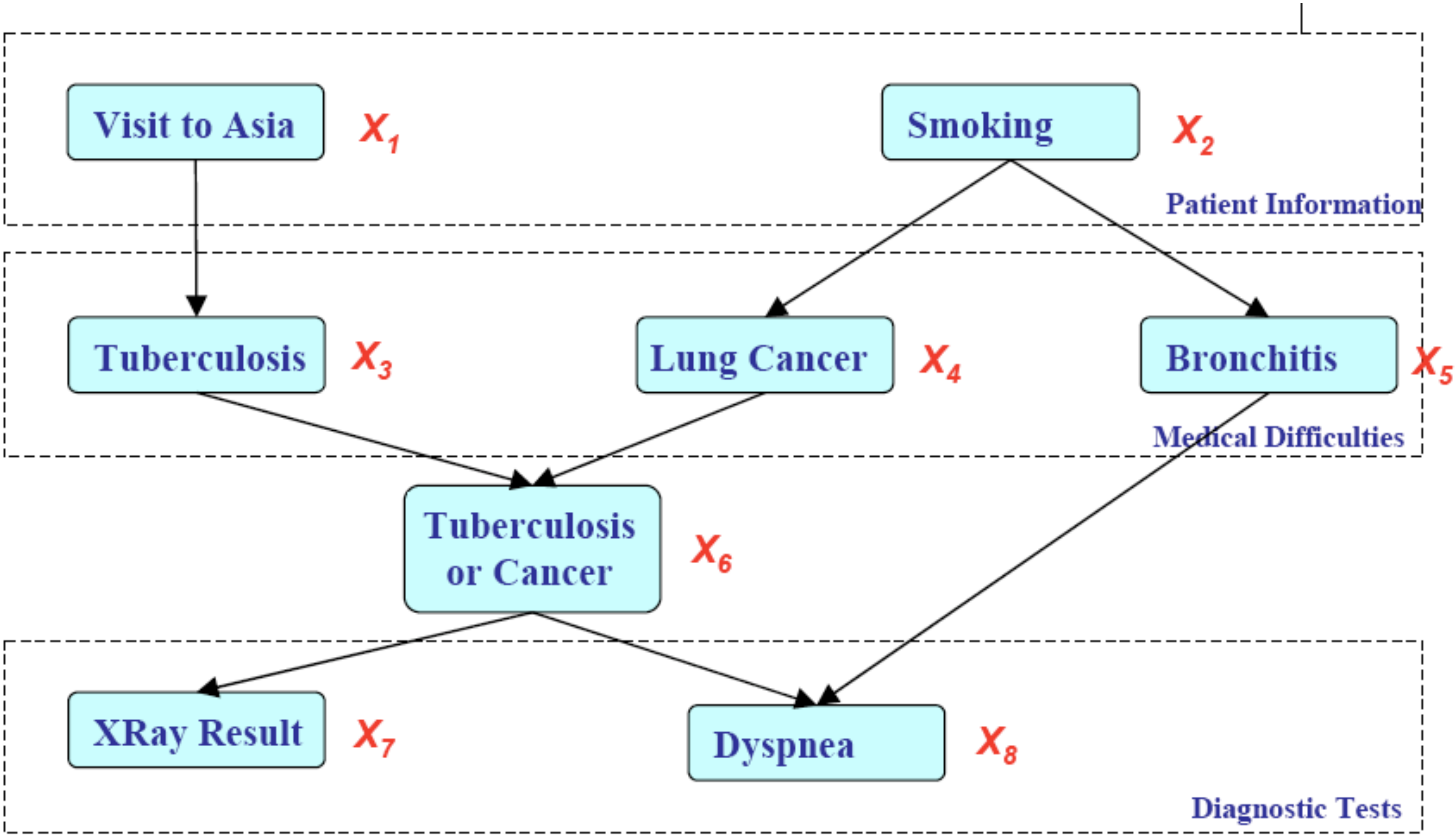
Equivalently, if

$$(\forall i, j) P(Y = y_i | X = x_j) = P(Y = y_i)$$

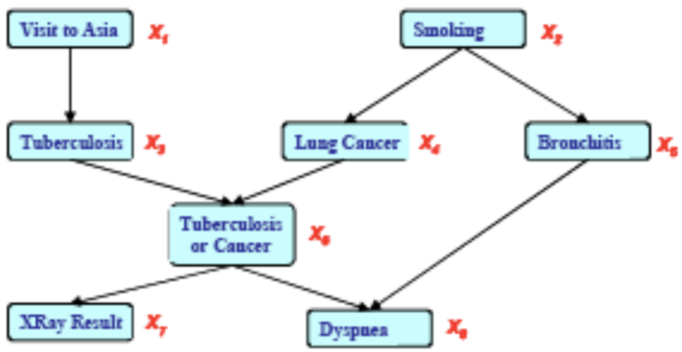
Represent Joint Probability Distribution over Variables



Describe network of dependencies



Bayes Nets define Joint Probability Distribution in terms of this graph, plus parameters

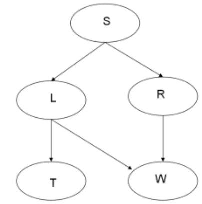


$$P(X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8)$$
$$= P(X_1)\, P(X_2)\, P(X_3 | X_1)\, P(X_4 | X_2)\, P(X_5 | X_2)\, P(X_6 | X_3, X_4)\, P(X_7 | X_6)\, P(X_8 | X_5, X_6)$$

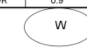
Benefits of Bayes Nets:

- Represent the full joint distribution in fewer parameters, using prior knowledge about dependencies
- Algorithms for inference and learning

Bayesian Networks Definition



Parents	P(W|Pa)	P(¬W|Pa)
L, R	0	1.0
L, ¬R	0	1.0
¬L, R	0.2	0.8
¬L, ¬R	0.9	0.1



A Bayes network represents the joint probability distribution over a collection of random variables

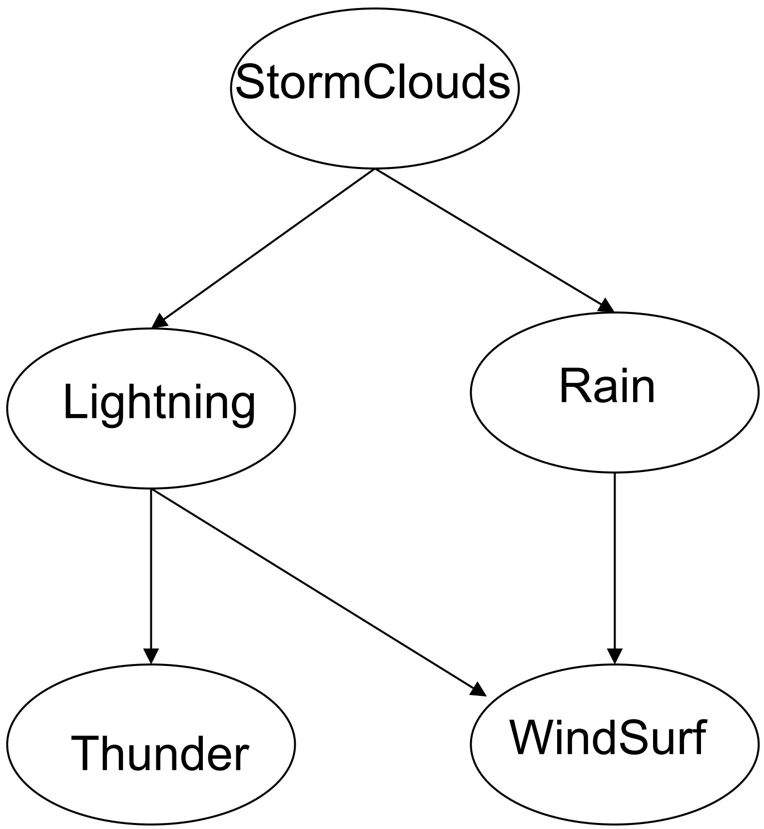
A Bayes network is a directed acyclic graph and a set of conditional probability distributions (CPD's)

- Each node denotes a random variable
- Edges denote dependencies
- For each node X_i its CPD defines $P(X_i \mid Pa(X_i))$
- The joint distribution over all variables is defined to be

$$P(X_1 \ldots X_n) = \prod_i P(X_i | Pa(X_i))$$

Pa(X) = immediate parents of X in the graph

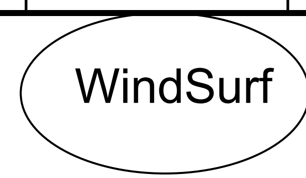
Bayesian Network



Nodes = random variables

A conditional probability distribution (CPD) is associated with each node N, defining P(N | Parents(N))

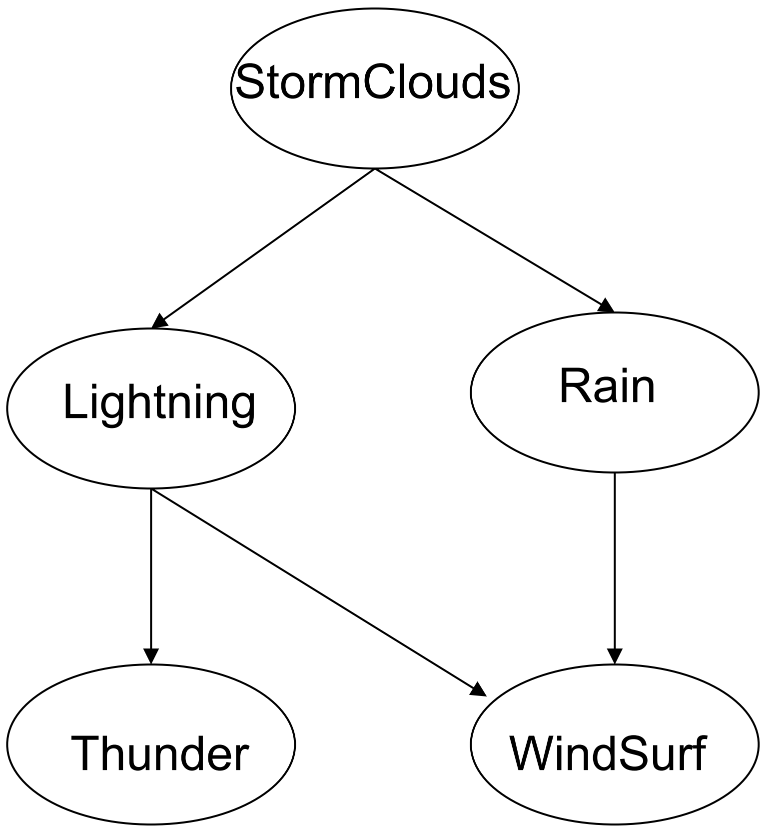
Parents	P(W|Pa)	P(¬W|Pa)
L, R	0	1.0
L, ¬R	0	1.0
¬L, R	0.2	0.8
¬L, ¬R	0.9	0.1



The joint distribution over all variables:

$$P(X_1 \ldots X_n) = \prod_i P(X_i | Pa(X_i))$$

Bayesian Network

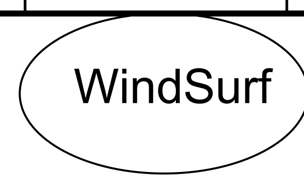


What can we say about conditional independencies in a Bayes Net?

One thing is this:

Each node is conditionally independent of its non-descendents, given only its immediate parents.

Parents	P(W|Pa)	P(¬W|Pa)
L, R	0	1.0
L, ¬R	0	1.0
¬L, R	0.2	0.8
¬L, ¬R	0.9	0.1



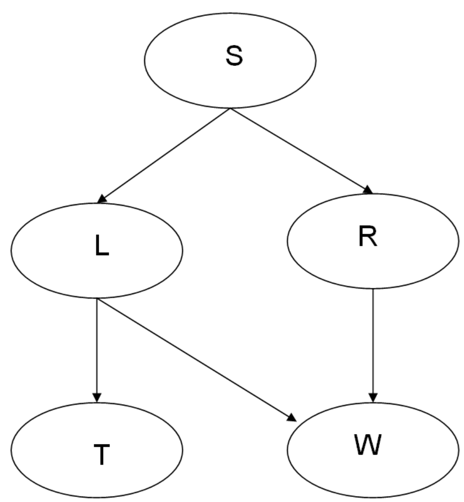
Some helpful terminology

Parents = Pa(X) = immediate parents

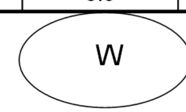
Antecedents = parents, parents of parents, ...

Children = immediate children

Descendents = children, children of children, ...

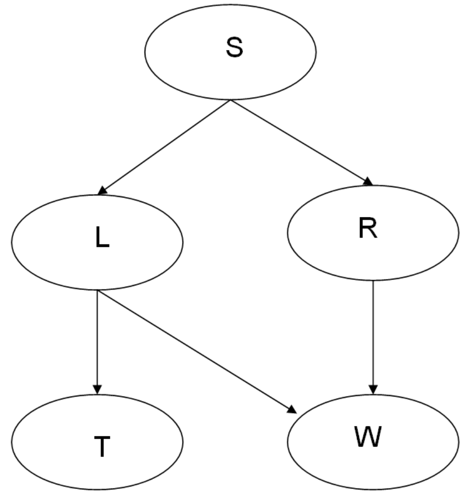


Parents	P(W|Pa)	P(¬W|Pa)
L, R	0	1.0
L, ¬R	0	1.0
¬L, R	0.2	0.8
¬L, ¬R	0.9	0.1

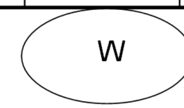


Bayesian Networks

- CPD for each node X_i describes $P(X_i \mid Pa(X_i))$



Parents	P(W|Pa)	P(¬W|Pa)
L, R	0	1.0
L, ¬R	0	1.0
¬L, R	0.2	0.8
¬L, ¬R	0.9	0.1



Chain rule of probability says that in general:

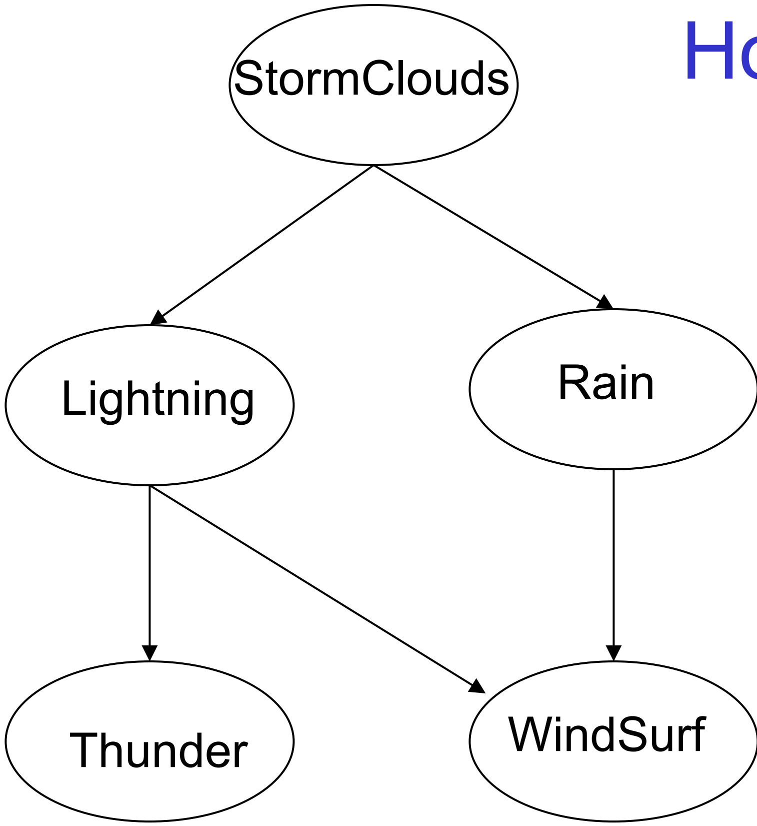
$$P(S,L,R,T,W) = P(S)P(L|S)P(R|S,L)P(T|S,L,R)P(W|S,L,R,T)$$

8 params

But in a Bayes net: $$P(X_1 \ldots X_n) = \prod_i P(X_i|Pa(X_i))$$

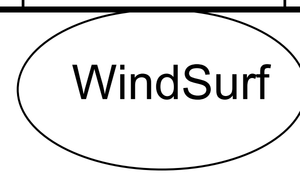
$$P(S,L,R,T,W) = P(S)P(L|S)P(R|S)P(T|L)P(W|L R)$$

2



How Many Parameters?

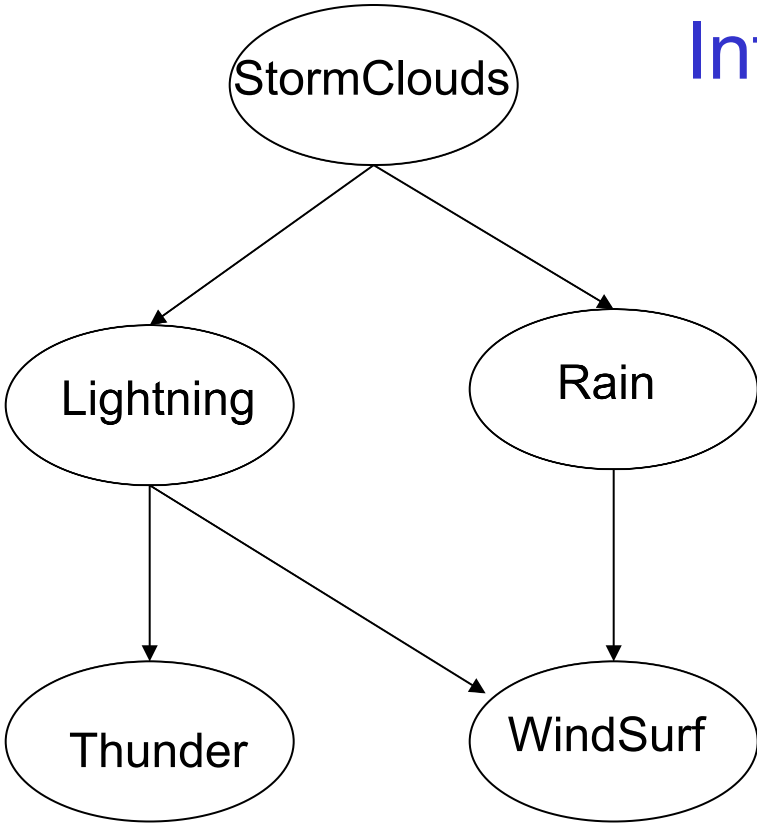
Parents	P(W|Pa)	P(¬W|Pa)
L, R	0	1.0
L, ¬R	0	1.0
¬L, R	0.2	0.8
¬L, ¬R	0.9	0.1



To define joint distribution in general?

To define joint distribution for this Bayes Net?

Inference in Bayes Nets

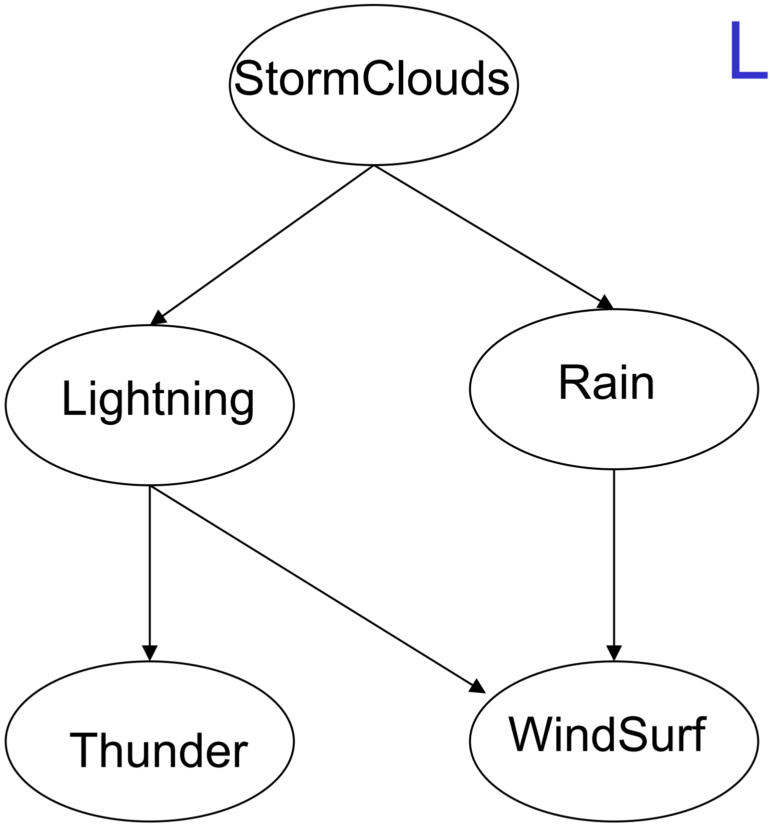


Parents	P(W|Pa)	P(¬W|Pa)
L, R	0	1.0
L, ¬R	0	1.0
¬L, R	0.2	0.8
¬L, ¬R	0.9	0.1

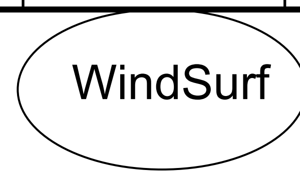
WindSurf

P(S=1, L=0, R=1, T=0, W=1) =

Learning a Bayes Net



Parents	P(W|Pa)	P(¬W|Pa)
L, R	0	1.0
L, ¬R	0	1.0
¬L, R	0.2	0.8
¬L, ¬R	0.9	0.1



Consider learning when graph structure is given, and data = { <s,l,r,t,w> }

What is the MLE solution? MAP?