CS 461: Machine Learning Principles

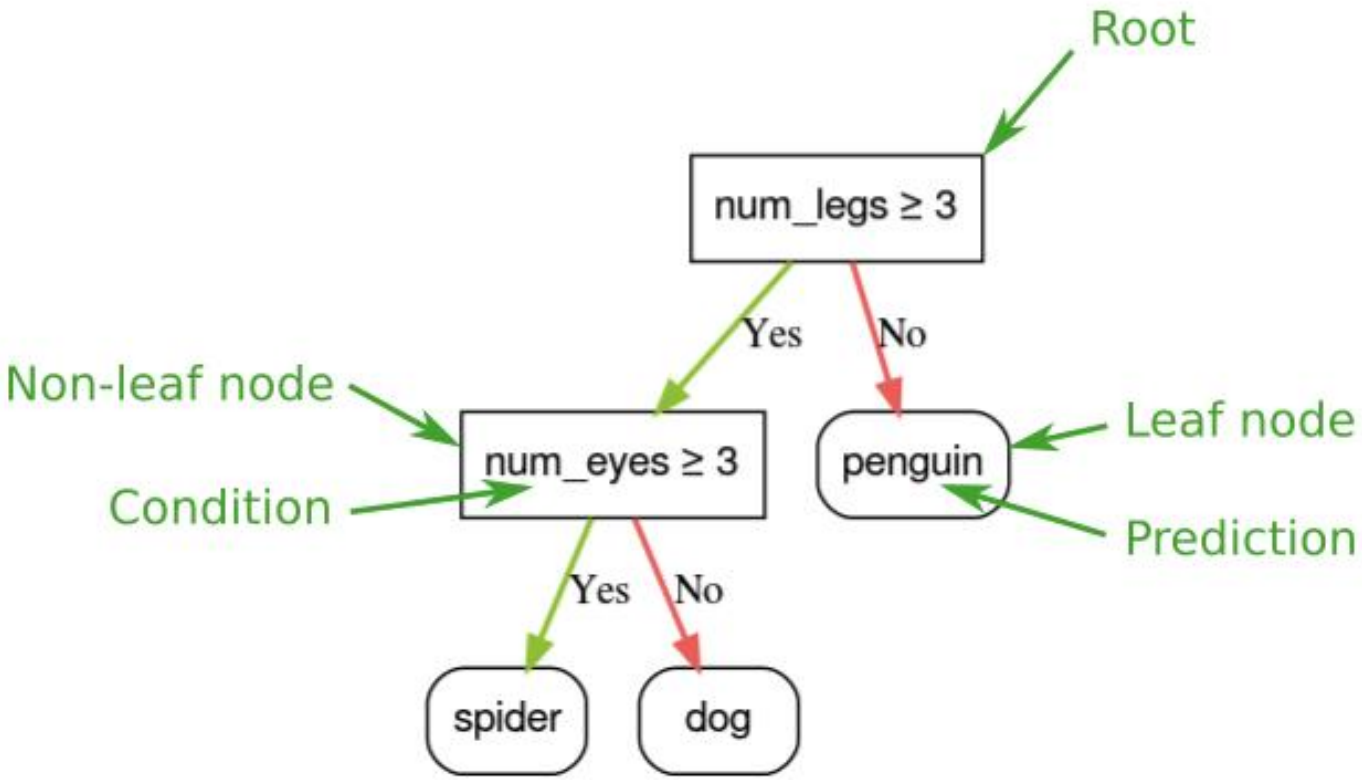
Class 14: Oct. 21

Boolean Decision Tree and Ensemble Learning

& Learning Theory

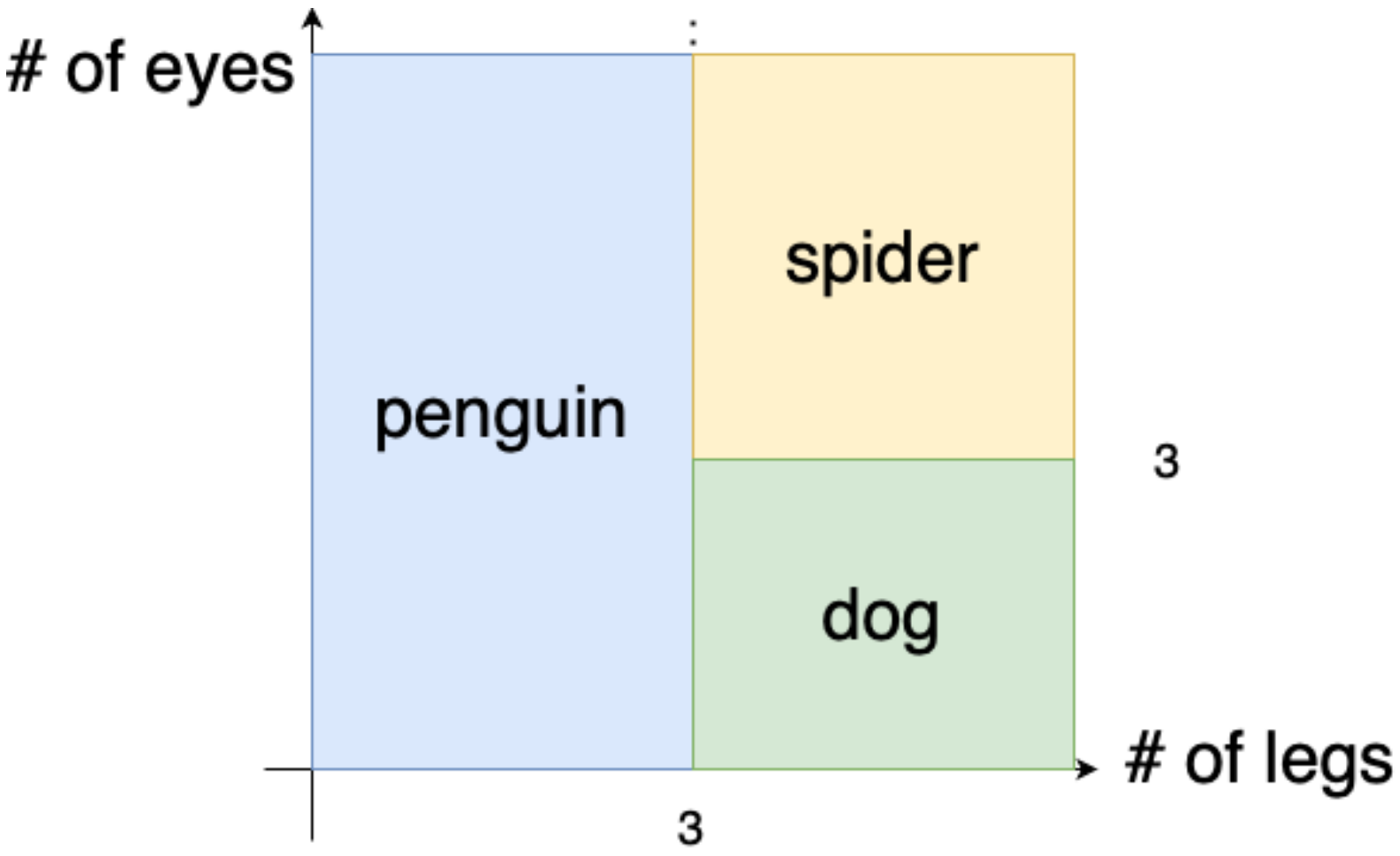
Instructor: Diana Kim

Decision Tree
: make a final classification decision through sequential questions.



https://developers.google.com/machine-learning/decision-forests/decision-trees

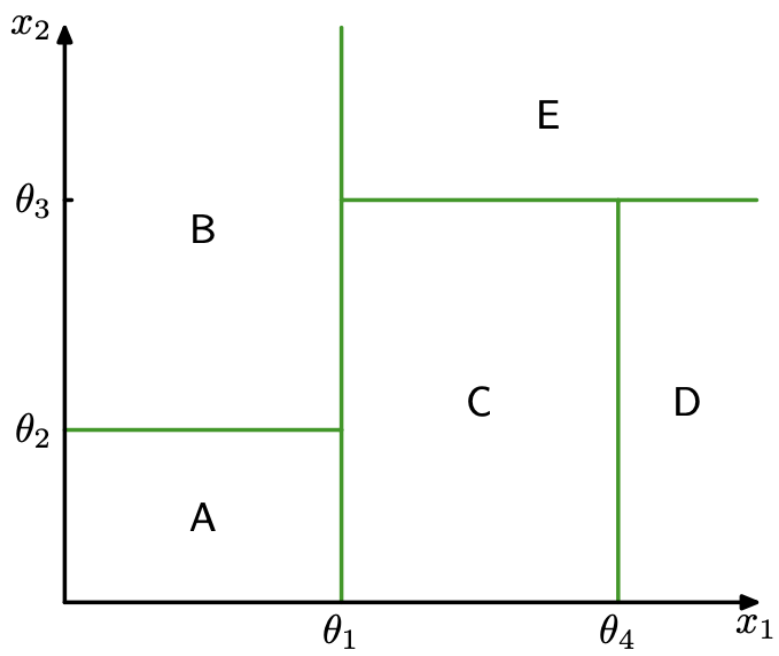
Decision tree partitions the data space using axis-aligned boundaries.

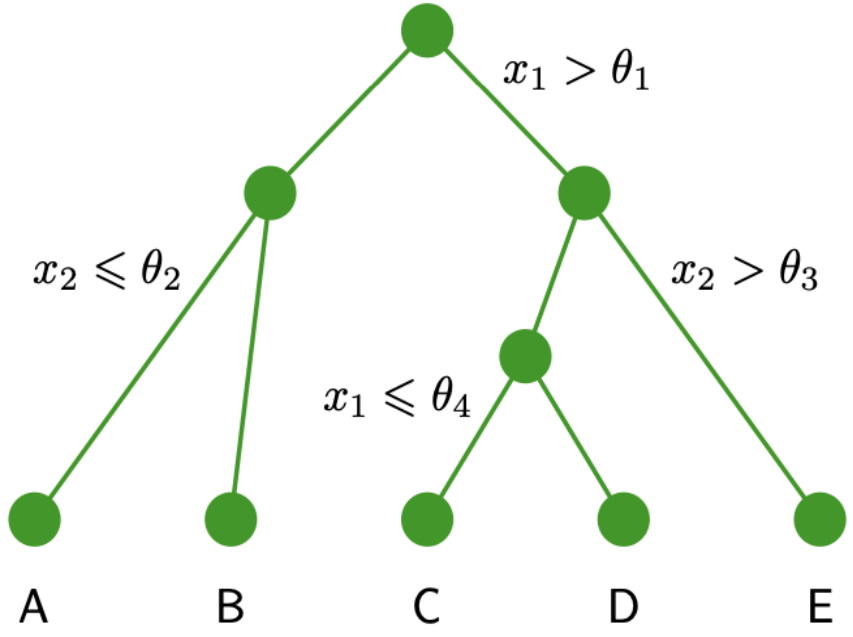


From Bishop 14.5 and 14.6

Decision Tree

:partitioning input space into the regions whose edges are aligned with axes.





Suppose there exist a logical flow in our mind
as deciding to wait or not in the restaurant. We want to learn the decision structure from data.

https://people.engr.tamu.edu/guni/csce421/files/AI_Russell_Norvig.pdf

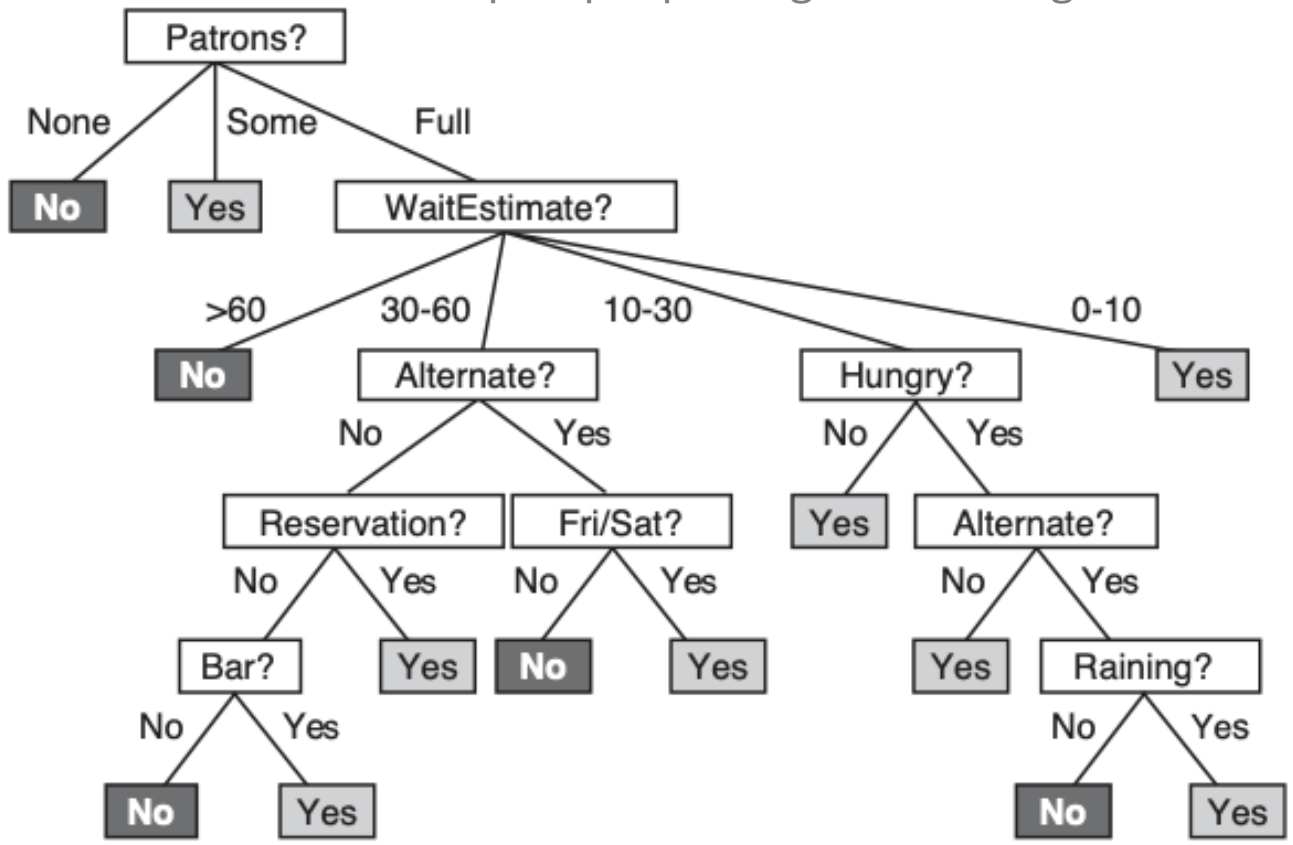


Figure 18.2 A decision tree for deciding whether to wait for a table.

Learning a decision tree <u>from data:</u>

It aims to build a structured rule to answer the question: wait or not in the restaurant?

https://people.engr.tamu.edu/guni/csce421/files/AI_Russell_Norvig.pdf

Example	Input Attributes										Goal
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
$\mathbf{x}_1$	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0–10*	$y_1 = Yes$
$\mathbf{x}_2$	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30–60*	$y_2 = No$
$\mathbf{x}_3$	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0–10*	$y_3 = Yes$
$\mathbf{x}_4$	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*Yes*	*No*	*Thai*	*10–30*	$y_4 = Yes$
$\mathbf{x}_5$	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	*>60*	$y_5 = No$
$\mathbf{x}_6$	*No*	*Yes*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Italian*	*0–10*	$y_6 = Yes$
$\mathbf{x}_7$	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0–10*	$y_7 = No$
$\mathbf{x}_8$	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0–10*	$y_8 = Yes$
$\mathbf{x}_9$	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	*>60*	$y_9 = No$
$\mathbf{x}_{10}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$$$*	*No*	*Yes*	*Italian*	*10–30*	$y_{10} = No$
$\mathbf{x}_{11}$	*No*	*No*	*No*	*No*	*None*	*$*	*No*	*No*	*Thai*	*0–10*	$y_{11} = No$
$\mathbf{x}_{12}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Burger*	*30–60*	$y_{12} = Yes$

Figure 18.3 Examples for the restaurant domain. human understandable semantic features!

How can we build a binary decision tree from the data?

Training a Decision Tree

Learning a Decision Tree:

Instead of searching a tree that minimizes a loss such as misclassification error,
the training algorithm starts with a single root node,
corresponding to the whole input space,
then it grows the tree by adding nodes one at a time.

- by selecting <u>one</u> optimal feature <u>at a time (local optimal)</u>
- by selecting <u>one</u> optimal threshold <u>at a time (local optimal)</u>

Learning a Decision Tree:

Greedy Divide and Conquer Strategy

: recursively breaking down a region into subregions by finding local optimal solutions one at a time.

Suppose in training, you got the training sample below at a node.
which feature would you like to select to predict Y between X_1 and X_2 ?

X1	X2	Y
T	T	+
F	T	+
T	T	+
F	T	+
T	F	-
F	T	-
T	F	-
F	T	-

How can we quantify the information that a feature provide about Y?

How can we quantify the information that a feature provide about Y?
"Mutual Information"

$$I(X;Y) = \sum_x \sum_y P(x,y) \log \frac{P(x,y)}{P(x)P(y)}$$

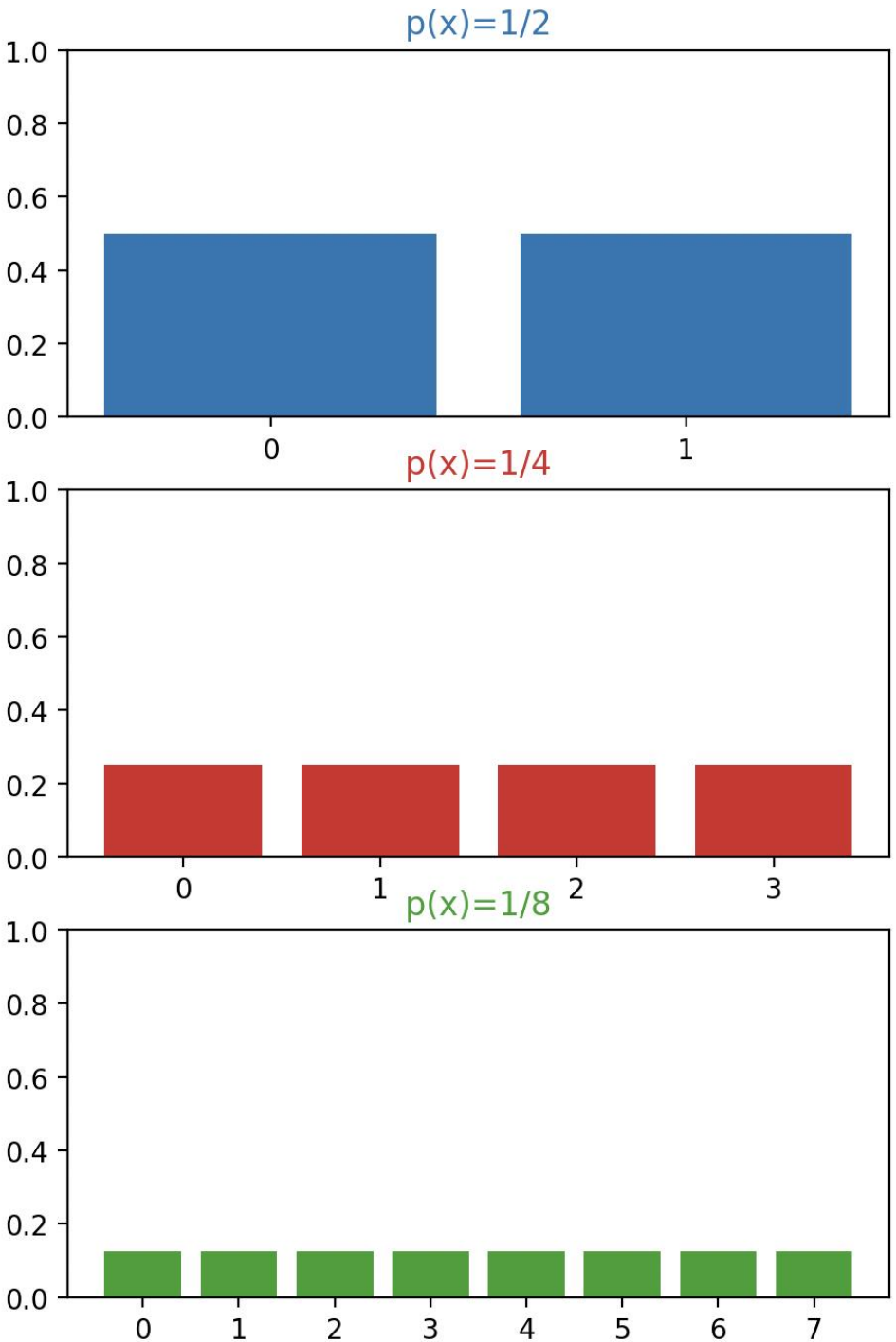
How can we quantify the information that a feature provide about Y?

"Mutual Information"

$$I(X;Y) = \sum_x \sum_y P(x,y) \log \frac{P(x,y)}{P(x)P(y)}$$

P(x,y)	X = 0	X = 1
Y=0	1/4	1/2
Y=1	1/8	1/8

Some Useful Information Theory Concepts for ML



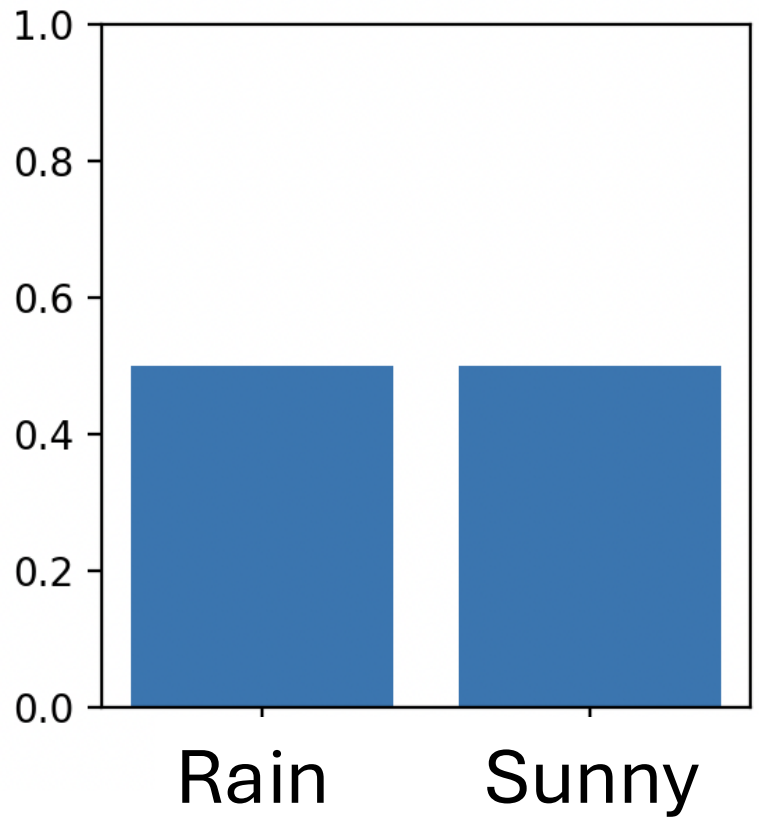
How many bits are required to represent the information?

The Entropy H(X) of Discrete Random Variable X ~ $P(X)$

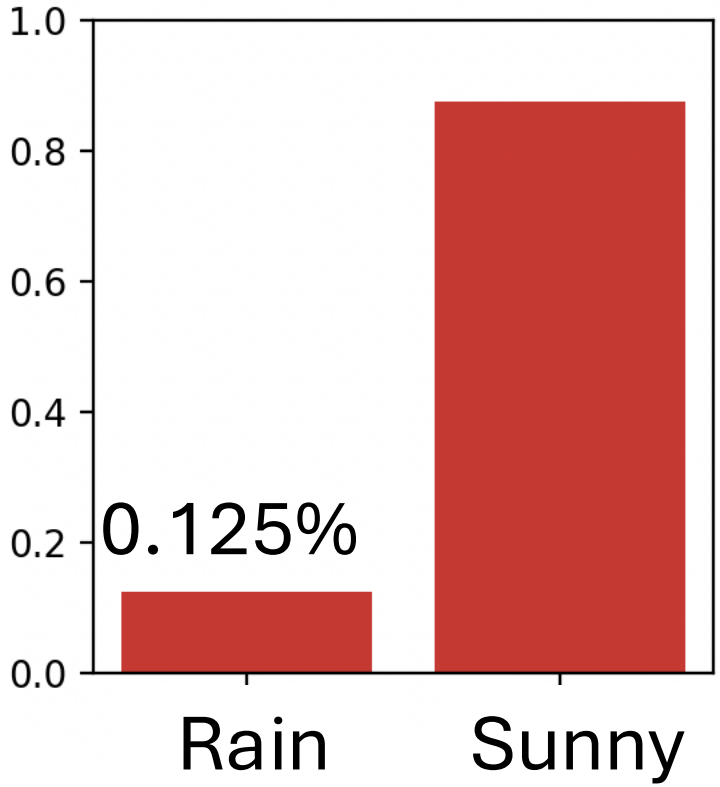
$$H(X) = E[\log \frac{1}{P(X)}] = \sum_x p(x) \log \frac{1}{p(x)}$$

Entropy measures the uncertainty of a random variable.

In which case are we more uncertain about tomorrow's weather?



vs.



The Conditional Entropy H(Y|X)
of Discrete Random Variable X and Y $\sim P(X,Y) \;=\; P(Y)P(Y|X)$

$$
\begin{aligned}
H(Y|X) &= \sum_x p(x) H(Y|x=x) \\
&= \sum_x p(x) \sum_y p(y|x) \log \frac{1}{p(y|x)} \\
&= \sum_x \sum_y p(x,y) \log \frac{1}{p(y|x)}
\end{aligned}
$$

$$Q\colon\ H(X) - H(X|Y)?$$

Back to the original question,

How can we quantify the information that a feature provide about Y?

"Mutual Information"

$$I(X;Y) = \sum_x \sum_y P(x,y) \log \frac{P(x,y)}{P(x)P(y)}$$

$$\boxed{= H(Y) - H(Y|X) =} \sum_x \sum_y P(x,y) \log \frac{P(x,y)}{P(x)P(y)}$$

The Properties of $I(X;Y)$

- A special case of KL divergence (**K**ullback-**L**eibler)

$$D(p||q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$$

$$= \sum_x p(x) \log \frac{1}{q(x)} - \sum_x p(x) \log \frac{1}{p(x)}$$

- <u>This is a kind</u> of distance measure between the density $p(x)$ and $q(x)$

$$I(X;Y) = \sum_x \sum_y p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$$

$$= D(p(x,y)||p(x)p(y))$$

- The distance measure between $p(x,y)$ and $\mathrm{p}(x)p(y)$

The Properties of $I(X;Y)$

$$D(p||q) = \sum_x p(x) \log \frac{p(x)}{q(x)} \geq 0$$

$$I(X;Y) \geq 0$$

Selecting a Feature to Split:
which one would you like to select to predict Y between X1 and X2?

X1	X2	Y
T	T	+
F	T	+
T	T	+
F	T	+
T	F	-
F	T	-
T	F	-
F	T	-

$$
\begin{aligned}
I(X;Y) &= H(Y) - H(Y|X_2) \\
&= H(Y) - \{p(x_2 = T)H(Y|X2 = T) + p(x_2 = F)H(Y|X_2 = F)\} \\
&= 1 - \{3/4 * (2/3 * \log 3/2 + 1/3 * \log 3) + 1/4 * 0)\}
\end{aligned}
$$

Selecting a Feature to Split:
which feature would you like to select to predict Y?

$$i = \arg\max_i I(X_i; Y)$$

$$i = \arg\max_i I$$

$$i = \arg\min_i H(Y|X_i) = \sum_{x_i} p(x_i) H(Y|x_i)$$

$$i = \arg\min_i \text{Gini-Index}(X_i) = \sum_{x_i} p(x_i) \cdot \text{Gini-Index}(x_i) \quad \text{where Gini-Index}(x_i) = 1 - \sum_y p(y|x_i)^2$$

CART (**Classification** and **R**egression Tree Algorithm): use gini –index
ID3, C4.5 (Iterative Dichotomiser 3): use mutual information

Decision Tree Algorithm

1. Choose an optimal feature to split.

2. Form a split criterion.

Based on Gini-Index / Cross Entropy

3. Decide when to declare a node terminal or continue splitting it. (based on <u>a preset stopping criterion</u>)

4. The assignment of each terminal node to a class

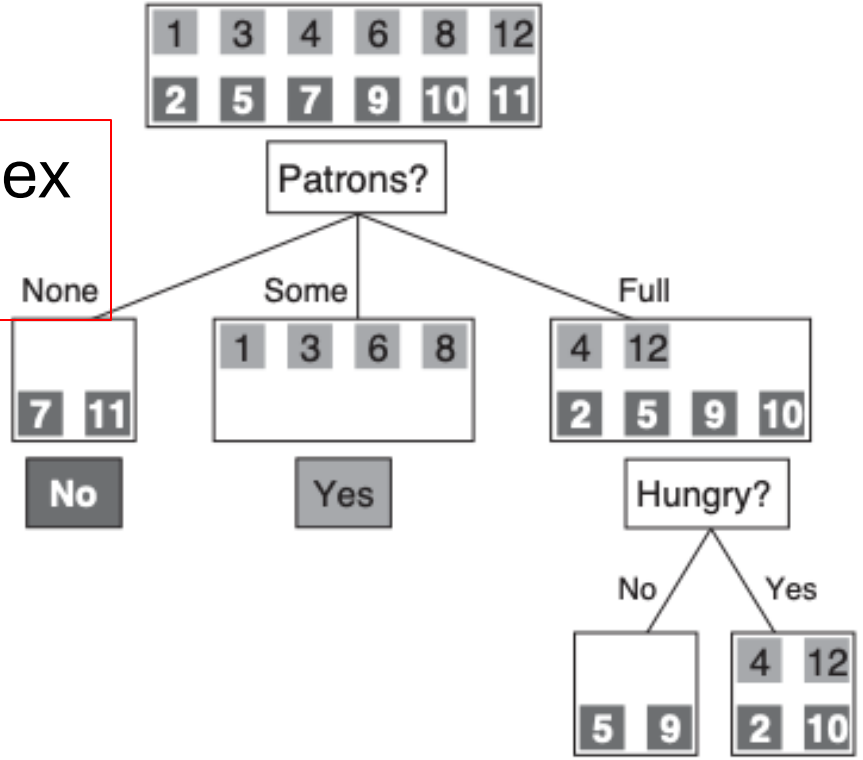


Figure 18.4
https://people.engr.tamu.edu/guni/csce421/files/AI_Russell_Norvig.pdf

Tree Algorithm Pro and Cons

1. They are readily interpretable by humans. +
2. Robust to the outliers. +
3. They are fast to fit, and scale well to larger data set. +
4. However, when the class structure depends on combinations of variables, the tree decision boundary (perpendicular to the coordinate axes) poorly uncover the structure. -

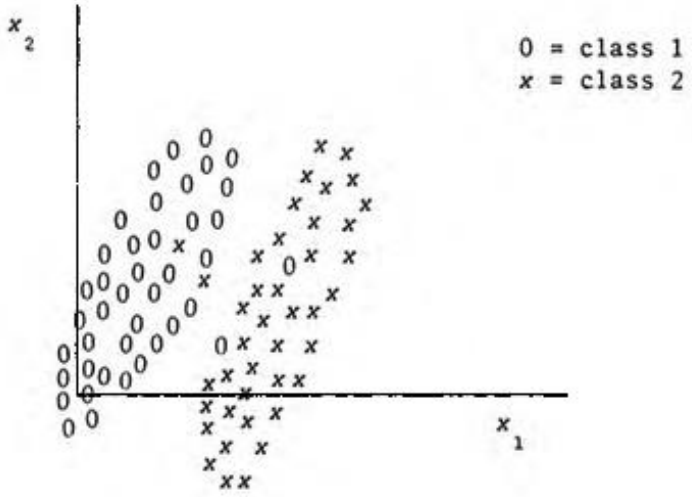


FIGURE 2.10

From the book "classification and regression trees" by Breiman, Leo,

Growing right sized trees is a primary issue.
<u>It is hard to set the right stopping criterion</u>.

- Stopping Rule: $max_{X_i} I(Y; X_i) < \eta$

- Too low η: too much splitting and tree is too large.
- Large η (early stopping):
  it is found empirically often that a small reduction for now but after <u>several more splits a substantial impurity reduction is found.</u>
  <u>(for greedy algorithm)</u>

$$max_{X_i} I(Y; X_i) < \eta$$

The best strategy is to set a low stopping criterion η,
so (1) <u>grow a very large tree</u> and (2) <u>prune</u> the tree later.

Pruning Trees

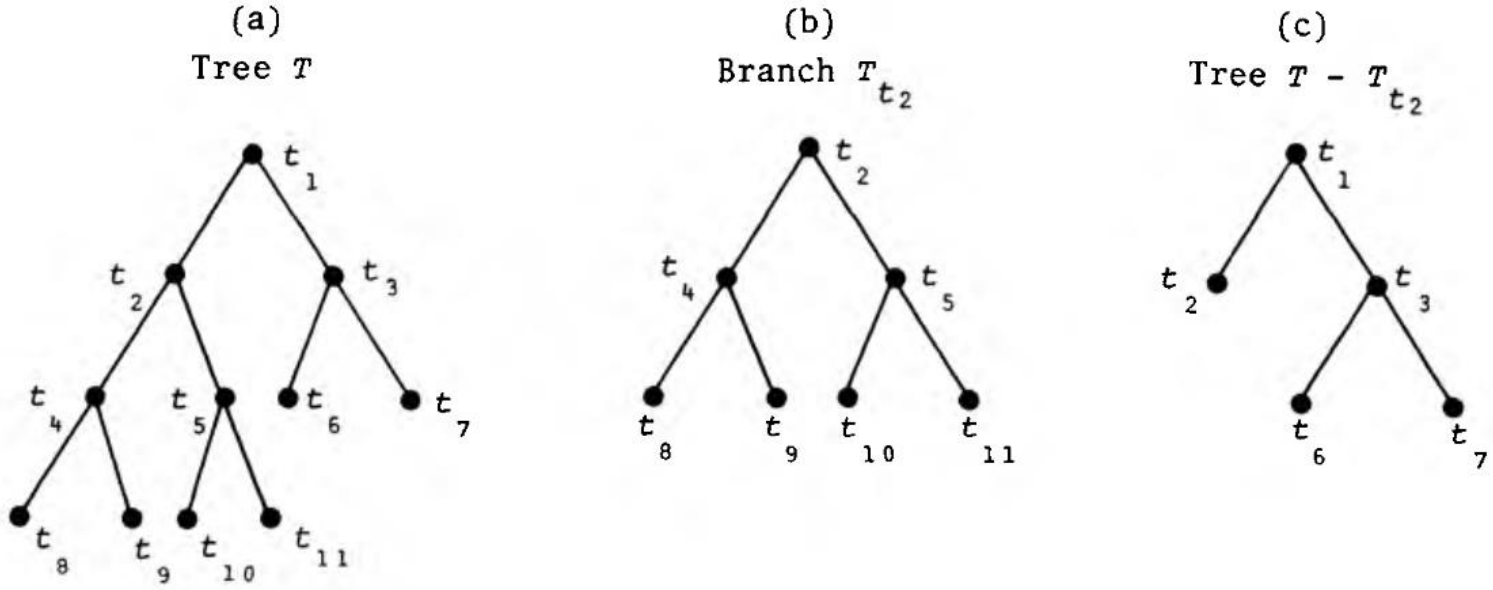


FIGURE 3.1

From the book "classification and regression trees" by Breiman, Leo,

Definition 1: branch]

A branch T_t of T with root node t consists of the node t and all descendants of t in T.

Definition 2: pruning]

Pruning T_t from a tree T consists of deleting all descendants of t except for the root node t.

Pruning Trees

(1) T , T_1 , $T_2 \ldots$, $root\ node$

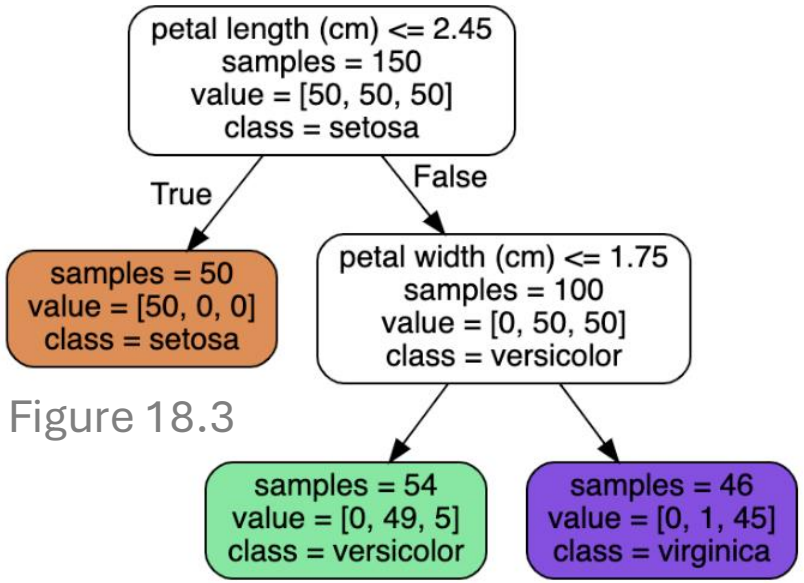
(A subtree whose # of terminal node reduced by 1)

(2) Pick a subtree that minimizes the pruning criterion.

$$\text{Pruning Criterion} = \sum_{i=1}^{\tilde{T}} Q(i) + \alpha|\tilde{T}|$$

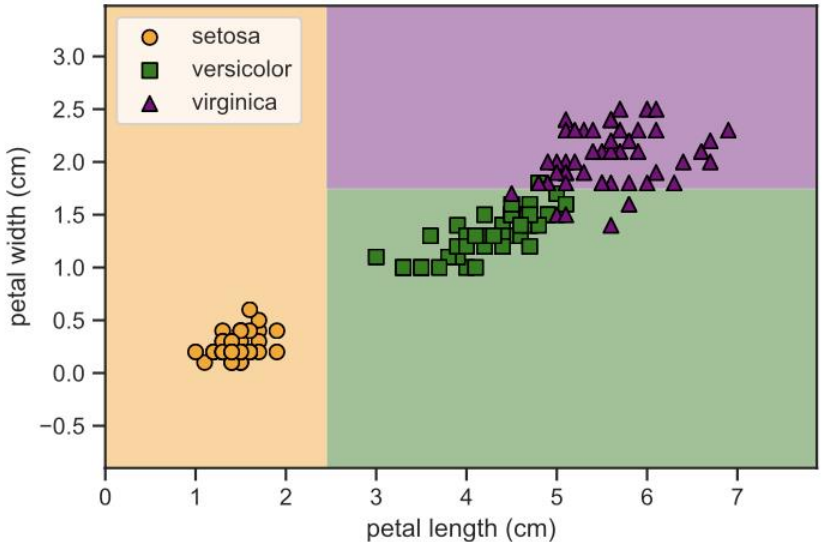
- $Q(i)$ is a performance measurement such as impurity, entropy, etc.
- $\tilde{T}$: $of\ terminal\ node\ of\ a\ subtree$
- α: $control\ model\ complexity\ and\ bias$.

Decision Tree naturally has high variance.
How can we handle the high variance issue in decision tree?

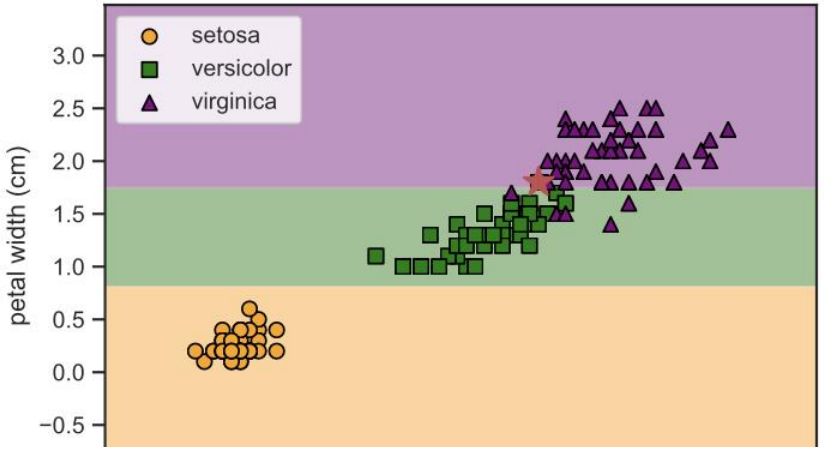


From bishop Figure 18.3

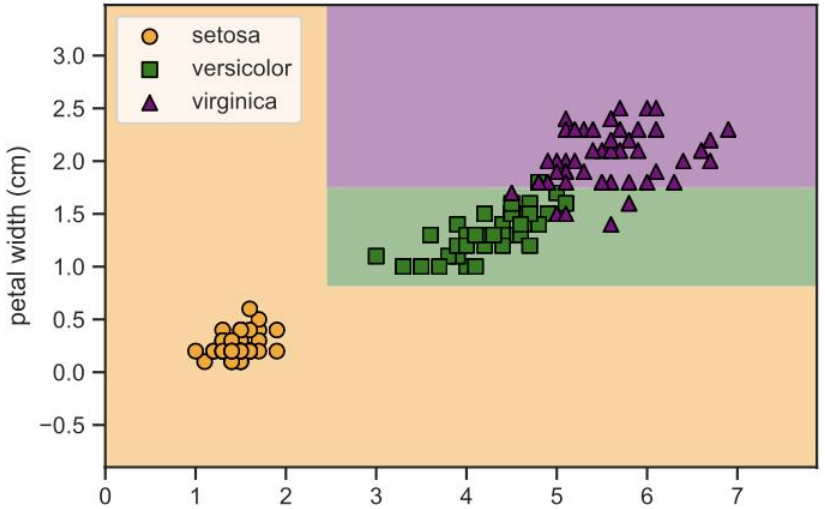
[a] decision tree



[b] decision surface by the tree [a]



[c] one missing data results
in very different decision surface.



[d] Ensemble Method

Random Forest
(Ensemble Methods: Grow Many Trees and do Majority Vote!)

Ensemble Learning

$$f(y|x) = \frac{1}{M}\sum_{m=1}^{M} f_m(y|x)$$

class* = majority vote $(f_1(x), f_2(x), f_3(x), \ldots, f_m(x))$

Q: how ensemble learning $f(y|x)$ are different with $f_m(y|x)$ in terms of bias and variance?

Random Forest is Ensemble Learning.
It introduces the two source of randomness.

(1)Bagging: each tree is grown using a bootstrap sample of training data. (with replacement)
(2)Random Vector : at each node, best split is chosen from a random sample of K features instead of taking all the features as a candidate.

The bias is scarified in random forest a bit to leverage the advantage of diverse models and reduce variance more effectively.

Learning Theory

Theoretical bounds to compute,

(1) how many data samples are required
to achieve |train error – true error| $< \epsilon$ with probability at least (1-δ)?

(2) how large should hypothesis space be to ensure that
|train error – true error| $< \epsilon$ with probability at least (1-δ)?

Empirical vs. Population Classification Error

$$R(w) = \int \frac{1}{2}|y - f(x;w)|f(x,y)dxdy$$, where $y \in \{-1,1\}$

$$R_{emp}(w) = \frac{1}{2N}\sum_{n=1}^{N}|y_n - f(x_n, y)|$$

[Haussler's 88 Bound]

- P[a hypothesis true error > ϵ but consistent with N data samples] $\leq (1-\epsilon)^N$
  * a hypothesis true error > ϵ implies that error (h) > ϵ
    where error (h) is the probability that h misclassify a new sample.

- P[any learned hypothesis true error > ϵ but consistent with N data samples]
  $\leq |H|\,(1-\epsilon)^N$ a finite |H|

- P[any consistent hypothesis true error > ϵ]
  $\leq |H|\,(1-\epsilon)^N \leq |H| e^{-N\epsilon} \leq \delta$
- With prob at least 1- δ, any consistent hypothesis's true error $\leq \epsilon$
  when
  $$N \geq \frac{\ln|H| + \ln\frac{1}{\delta}}{\epsilon}$$

But in reality, training error may not be consistent with training data.
We are interested in P[|train error – true error| $< \epsilon$]

[Hoeffding's Bound]

- P[|train error (h) – true error (h) | $\geq \epsilon$] $\leq 2e^{-2N\epsilon^2}$
- P[any learned hypothesis |train error (h) – true error (h) | $\geq \epsilon$]
  $\leq 2|H|e^{-2N\epsilon^2}$

- true error $(h) \leq$ train error (h) + $\sqrt{\dfrac{\ln|H| + \ln\frac{2}{\delta}}{2m}}$, with prob at least $1-\delta$.

- With prob at least 1- δ, |train error (h) – true error (h)| $< \epsilon$ when

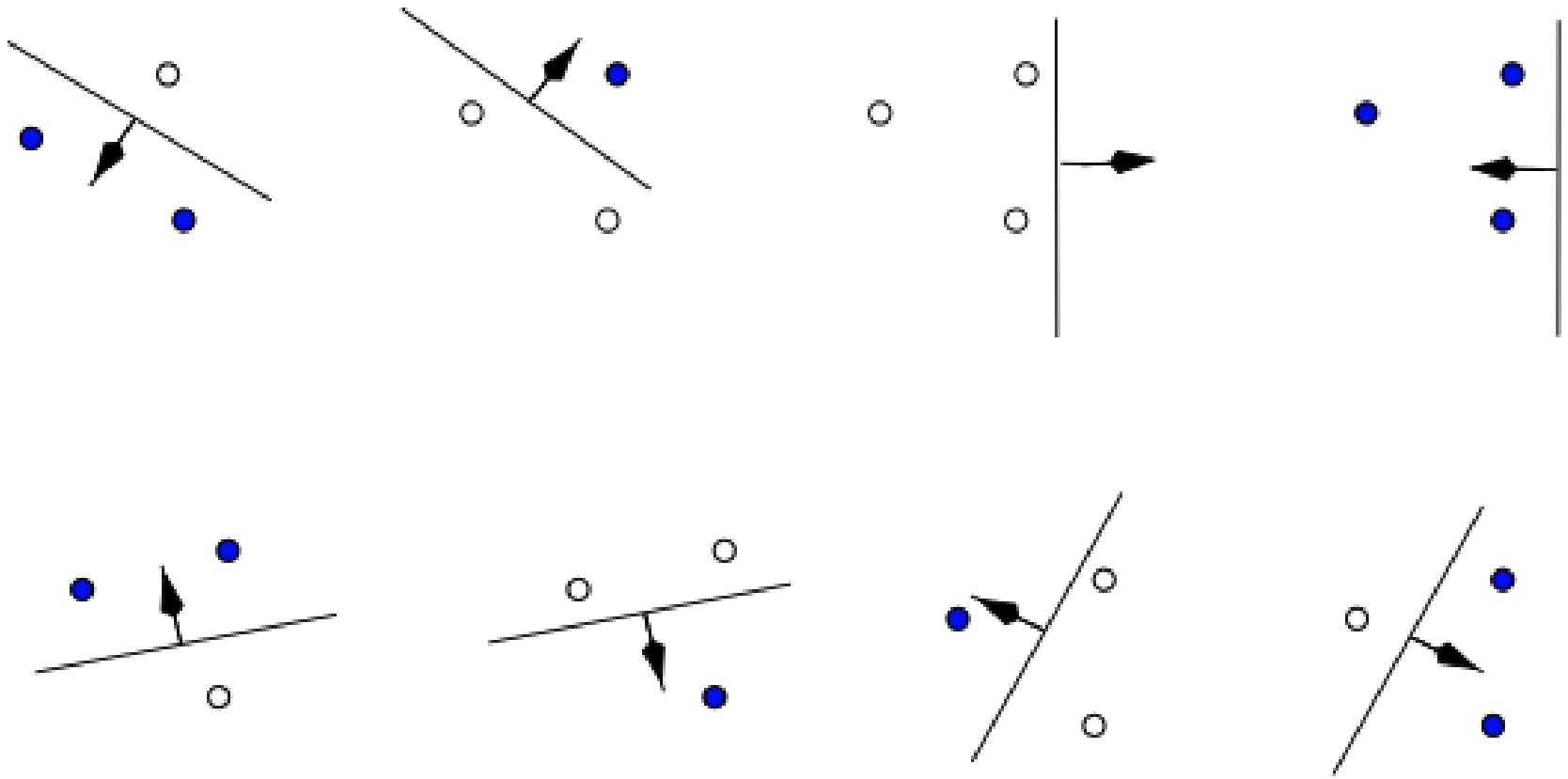
$$N \geq \frac{1}{2\epsilon^2}(\ln|H| + \ln\frac{2}{\delta})$$

In practice, hypothesis space dimension can be infinite and continuous.

[VC (Vapnik–Chervonenkis) dimension Bound]

- train error $(h) \leq$ train error (h) + $\sqrt{\frac{VC(H)(\ln(2N/VC(H)) + 1) + \ln 4/\delta}{N}}$

VC dimension for the set of functions $f(x; w)$
is the maximum number N of training points $(x_1, x_2, \ldots, x_N)$ that can be separated into two classes in all 2^N possible labeling configurations by $f(x; w)$.



- VC dimensions of the set of oriented hyperplanes in $R^N is\ N + 1$