AI300学院

机器学习之监督学习

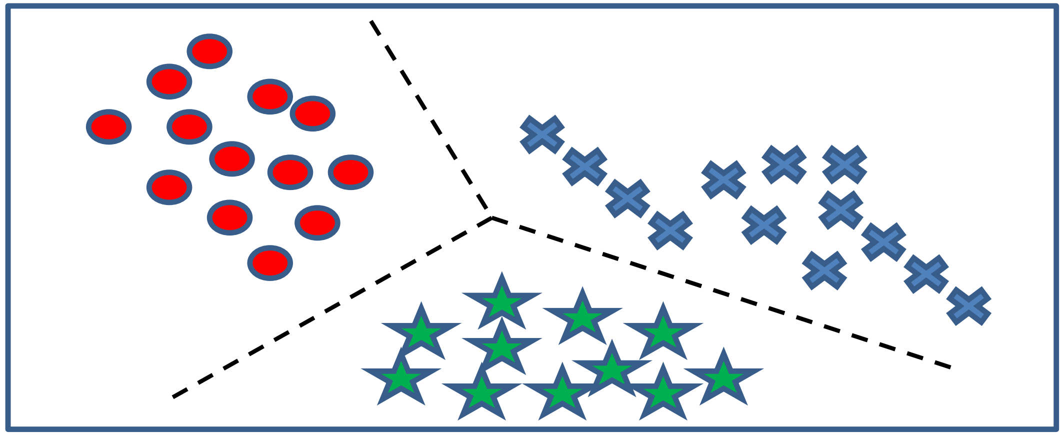
决策树、集成学习、随机森林

倪冰冰
上海交通大学

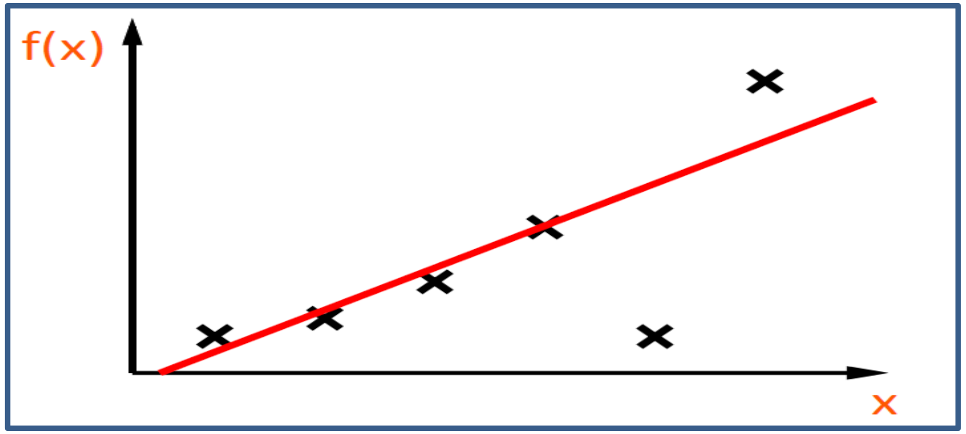
监督学习

- 给定一组数据，我们知道正确的输出结果应该是什么样子，并且知道在输入和输出之间有着一个特定的关系$f(x)$。

- 分类 vs 回归



分类(Classification)

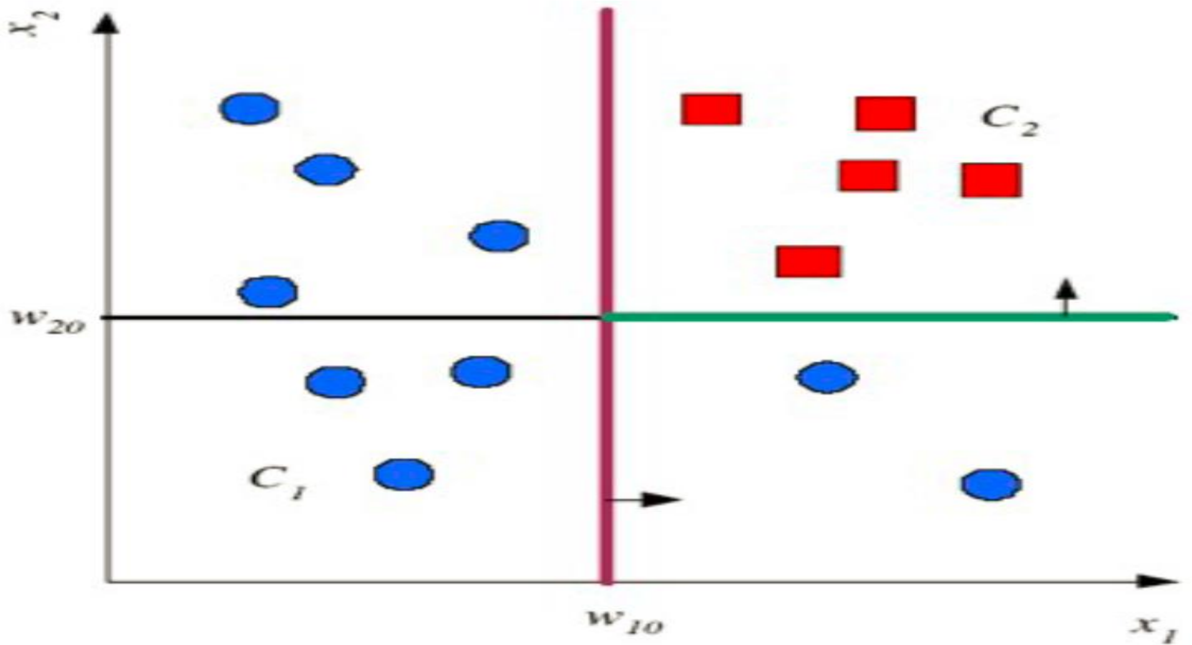


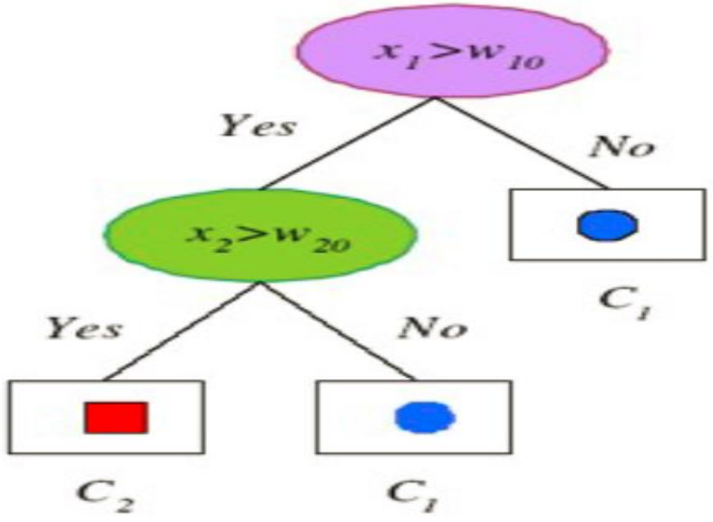
回归(Regression)

决策树Decision Tree

- 决策树 *Decision Tree*
  - 树状数据结构：节点（根，叶，中间），分叉（判别条件）
  - 每个中间节点包含branching条件判断., “Is $x_4 \geq 0.4$?”. “是”走左子树，“不是”走右子树
  - 每个数据样本从根节点，根据分叉条件，往下沉
  - 每个叶节点收集走到它的所有数据样本

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_D \end{pmatrix}$$

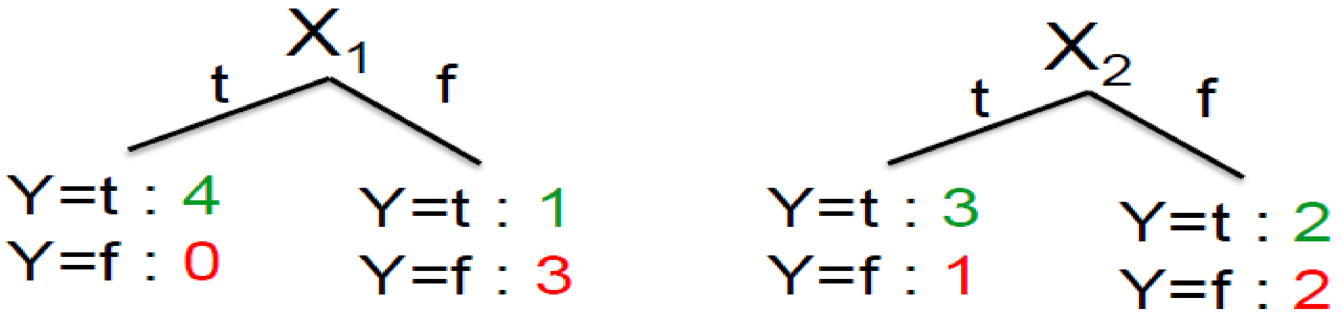




Decision Tree

- 所谓决策树的学习，即学习每个中间节点（包括根节点）的判别条件

Would we prefer to split on X_1 or X_2?

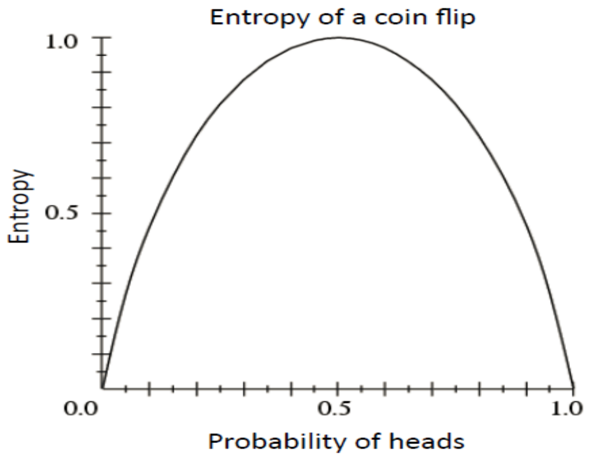


X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F
F	T	F
F	F	F

Y:标签　　　X_1,X_2:特征

Decision Tree

- Entropy熵 ***More uncertainty, more entropy!***



$$H(Y) = -\sum_{i=1}^{k} P(Y = y_i) \log_2 P(Y = y_i)$$

X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F

P(Y=t) = 5/6

P(Y=f) = 1/6

$H(Y) = - 5/6 \log_2 5/6 - 1/6 \log_2 1/6$

$= 0.65$

Decision Tree

- 条件熵–测量给定x条件下y的不确定性

$$H(Y \mid X) = -\sum_{j=1}^{v} P(X = x_j) \sum_{i=1}^{k} P(Y = y_i \mid X = x_j) \log_2 P(Y = y_i \mid X = x_j)$$

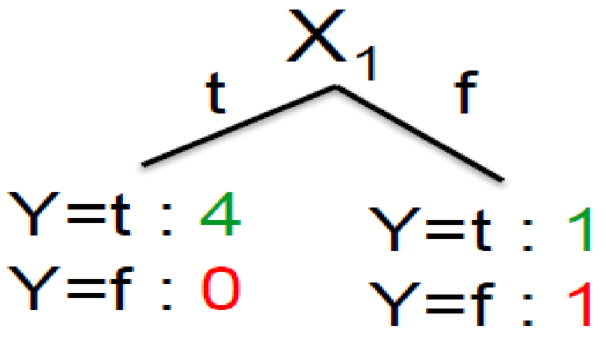
或者：

$$H(Y|X) = \sum_{v} \frac{|D_v|}{|D|} H(Y|D_v)$$

Example:

$P(X_1=t) = 4/6$

$P(X_1=f) = 2/6$



X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F

$$H(Y|X_1) = -\ 4/6\ (1 \log_2 1 + 0 \log_2 0)$$
$$-\ 2/6\ (1/2 \log_2 1/2 + 1/2 \log_2 1/2)$$
$$= 2/6$$

Decision Tree

- Information gain信息增益

$$IG(X) = H(Y) - H(Y \mid X)$$

In our running example:

$$IG(X_1) = H(Y) - H(Y|X_1)$$
$$= 0.65 - 0.33$$

$IG(X_1) > 0 \rightarrow$ we prefer the split!

X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F

- 策略：每次分叉，选择信息增益最大的分叉条件

Decision Tree

- 决策树训练算法（输入：一批训练数据包括特征和标签）

1. 建立一棵空的决策树（从根节点开始)
2. 迭代: 在当前节点，根据最佳的特征进行Split操作
   - 例如：可以考虑以下最大熵增的原则

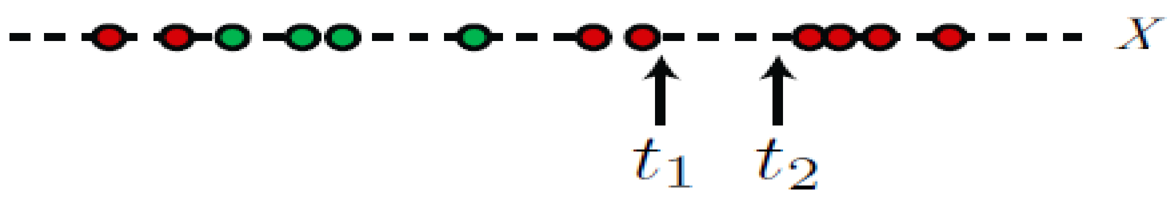
$$\arg\max_i IG(X_i) = \arg\max_i H(Y) - H(Y \mid X_i)$$

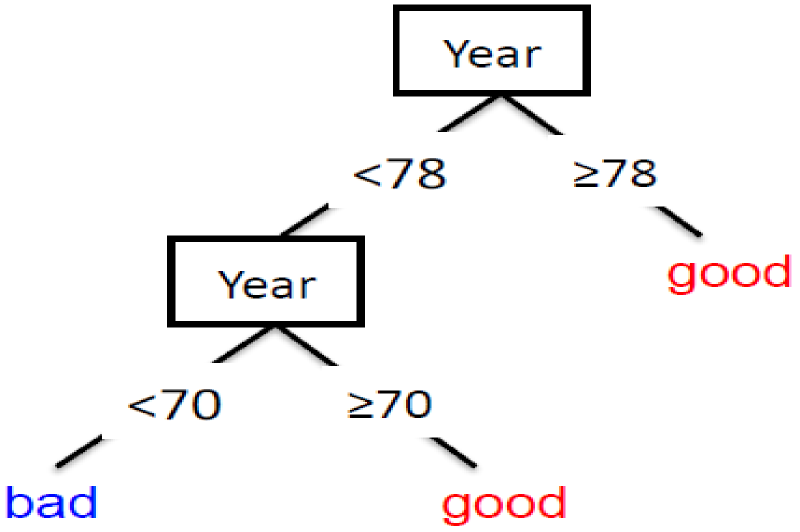
3. 直至收敛 (例如：最大深度达到, purity值达到)

- 测试新样本 $\boldsymbol{x}$
  - 根据所学习的split rule，将样本从根节点置入，往下遍历至叶节点
  - 一旦到达叶节点，将叶节点所属类别标签赋给样本 $\boldsymbol{x}$

Decision Tree

- 连续变量: 对于特征 x 根据阈值 t 进行split操作
  - 左子树 $x < t$
  - 右子树 $x \geq t$
  - 由于连续变量 t 可取值无限多
  - Only need check a finite number of t





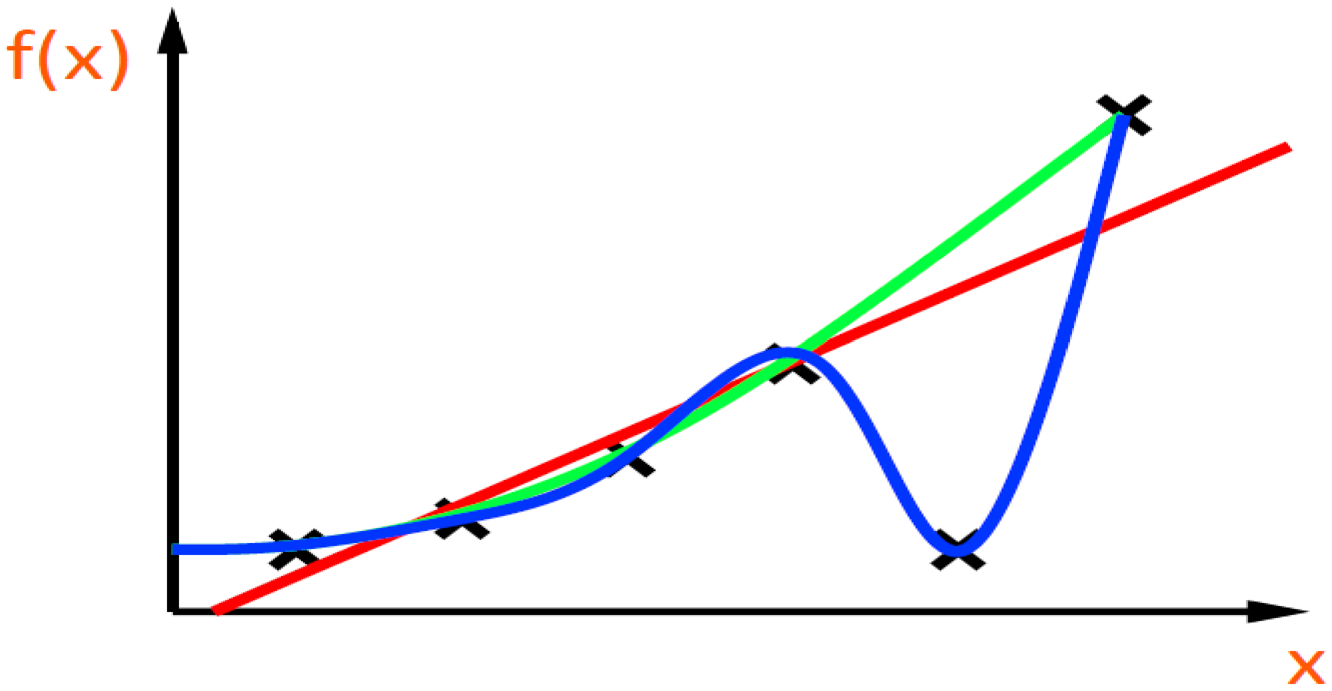
  - 可以将X 排序 $\{x_1, x_2, \dots, x_m\}$
  - 选择阈值 $x_i + (x_{i+1} - x_i)/2$

- 可以在一个特征上连续split

AI300学院

学习模型中的误差分析

- 那个模型比较好?



学习模型中的误差分析

- Variance-Bias-Noise 分析
  - 物理模型 $y = f(x) + \varepsilon$
  - ε 是0均值，σ方差的正态分布.
  - 给定训练数据D = $\{(x_i, y_i)\}$, 我们拟合hypothesis $\hat{f}$
  - 误差可以分解为：

$$\mathbf{E}\left[\left(y-\hat{f}(x)\right)^2\right] = \mathbf{Bias}\left[\hat{f}(x)\right]^2 + \mathbf{Var}\left[\hat{f}(x)\right] + \sigma^2$$

$$\mathbf{Bias}\left[\hat{f}(x)\right] = \mathbf{E}\left[\hat{f}(x) - f(x)\right]$$ *Structural Error*

$$\mathbf{Var}\left[\hat{f}(x)\right] = \mathbf{E}[\hat{f}(x)^2] - \mathbf{E}[\hat{f}(x)]^2$$ *Data Sample Error*

Bias-Variance 分解

- 证明:

$$
\begin{aligned}
\mathbf{E}[(y-\hat{f})^2] &= \mathbf{E}[y^2+\hat{f}^2-2y\hat{f}] \\
&= \mathbf{E}[y^2]+\mathbf{E}[\hat{f}^2]-\mathbf{E}[2y\hat{f}] \\
&= \mathbf{Var}[y]+\mathbf{E}[y]^2+\mathbf{Var}[\hat{f}]+\mathbf{E}[\hat{f}]^2-2f\mathbf{E}[\hat{f}] \\
&= \mathbf{Var}[y]+\mathbf{Var}[\hat{f}]+(f-\mathbf{E}[\hat{f}])^2 \\
&= \mathbf{Var}[y]+\mathbf{Var}[\hat{f}]+\mathbf{E}[f-\hat{f}]^2 \\
&= \sigma^2+\mathbf{Var}[\hat{f}]+\mathbf{Bias}[\hat{f}]^2
\end{aligned}
$$

$$\boxed{\mathbf{Var}[X]=\mathbf{E}[X^2]-\mathbf{E}[X]^2}$$

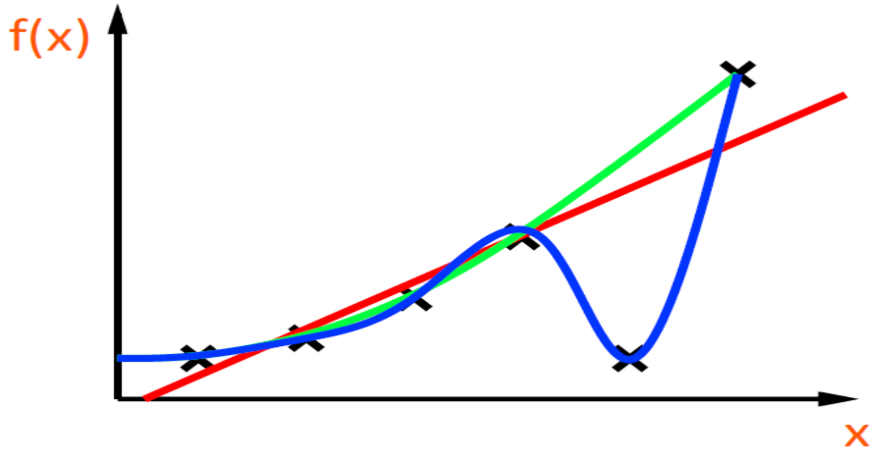
$$\boxed{\mathbf{E}[y]=\mathbf{E}[f+\epsilon]=\mathbf{E}[f]=f}$$

$$\boxed{\mathbf{Var}[y]=\mathbf{E}[(y-\mathbf{E}[y])^2]=\mathbf{E}[(y-f)^2]=\mathbf{E}[(f+\epsilon-f)^2]=\mathbf{E}[\epsilon^2]=\mathbf{Var}[\epsilon]+\mathbf{E}[\epsilon]^2=\sigma^2}$$

AI300学院

Bias-Variance 分解

- 对于 **classification**, 我们可以将误差分解为: bias 和 variance
  - Bias: 若模型过于简单，所提模型**不能**拟合数据，容易产生“under-fitting”
  - **Fix**: 使用更加复杂的模型
  - 但是variance会升高!
  - Variance: 若模型过于复杂，正确拟合数据变得困难，容易产生“over-fitting”
  - **Fix**: 使用更加简单的模型
  - 但是bias会升高!



Ensemble Learning集成学习

- 为什么需要集成学习
  - 单个的分类算法能力不足
  - 训练数据量不够

- 集成学习原则
  - 一堆弱分类器
  - 数据复用

Ensemble Learning集成学习

- Bootstrapping
  - 从大小为n的数据库D中根据它的数据分布，随机采样n’数据样本，并且对每个采样得到的样本做一定的数据偏移
  - 可重复采样

A sample of a single classifier on an imaginary set of data.	
	(Original) Training Set
Training-set-1:	1, 2, 3, 4, 5, 6, 7, 8

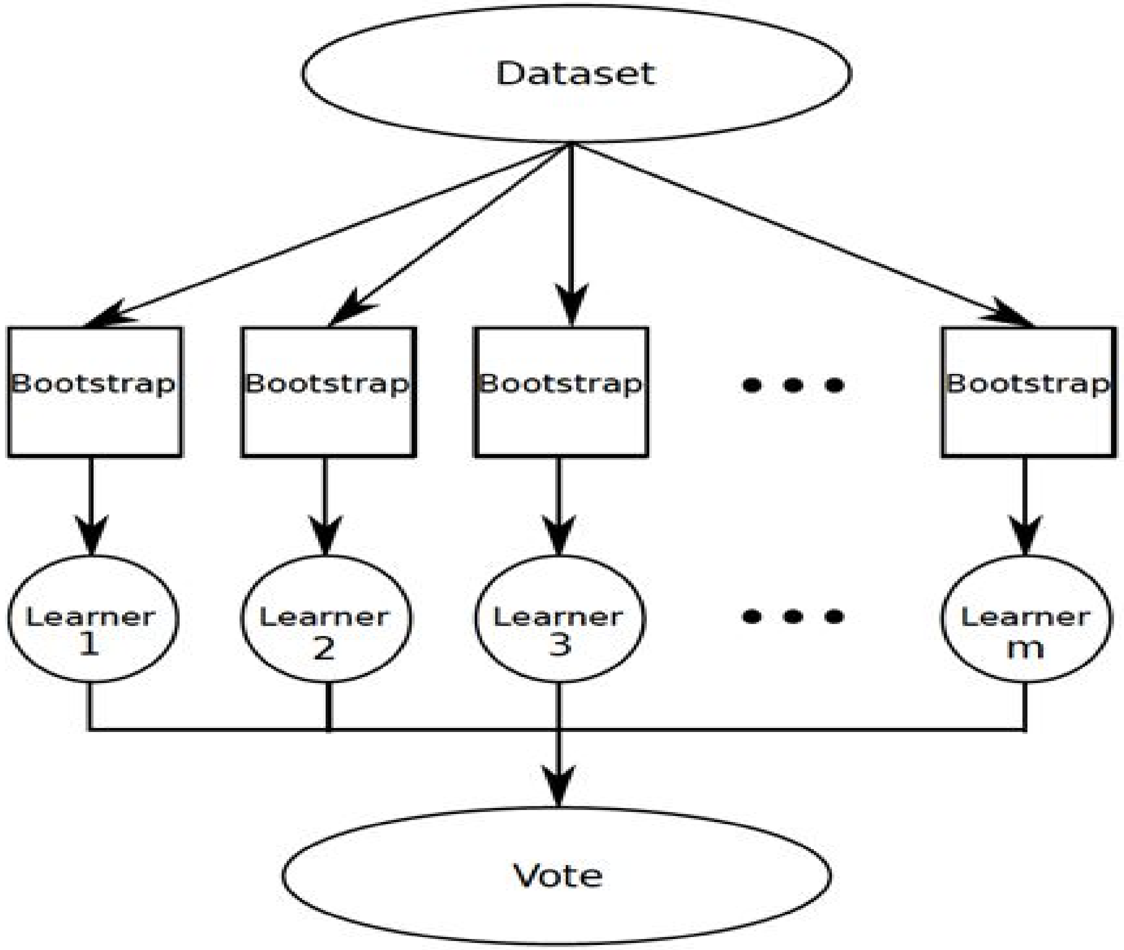
A sample of Bagging on the same data.	
	(Resampled) Training Set
Training-set-1:	2, 7, 8, 3, 7, 6, 3, 1
Training-set-2:	7, 8, 5, 6, 4, 2, 7, 1
Training-set-3:	3, 6, 2, 7, 5, 6, 2, 2
Training-set-4:	4, 5, 1, 4, 6, 4, 3, 8

- Bagging (known as Bootstrap Aggregation)
  - 重复多次:
    - 通过bootstrap方式 D 中采样得到集合D_k
    - 使用 D_K 训练得到一个分类器

A sample of Boosting on the same data.	
	(Resampled) Training Set
Training-set-1:	2, 7, 8, 3, 7, 6, 3, 1
Training-set-2:	1, 4, 5, 4, 1, 5, 6, 4
Training-set-3:	7, 1, 5, 8, 1, 8, 1, 4
Training-set-4:	1, 1, 6, 1, 1, 3, 1, 5

Bagging

- Bagging
  - 投票决定最后的识别结果



AI300学院

Bagging

- Bagging
  - Bias [跟之前一样]
  - Variance [最小化]
  - bagging 能减少 variance，且能使 bias 不被改变

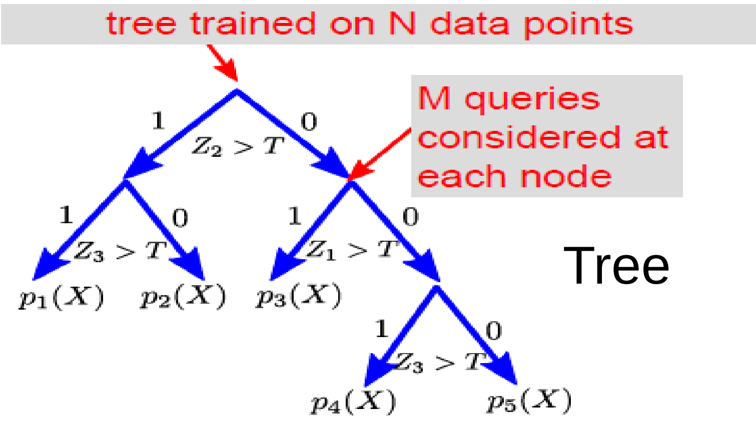
$$\mathbf{E}\left[\left(y-\hat{f}(x)\right)^2\right] = \mathbf{Bias}\left[\hat{f}(x)\right]^2 + \mathbf{Var}\left[\hat{f}(x)\right] + \sigma^2$$

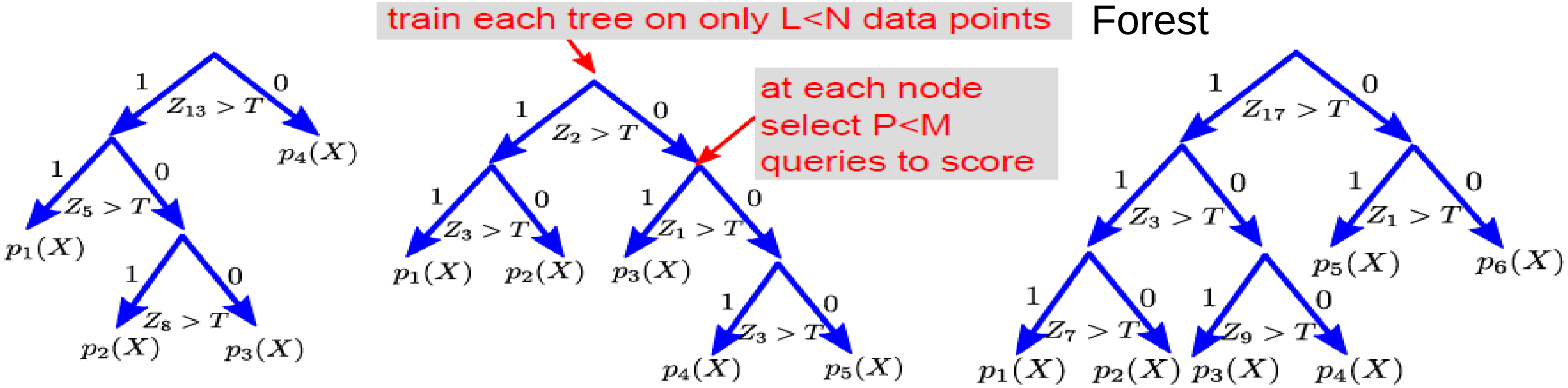
$$Var\left[\frac{1}{n}\sum_i (h^i)\right] = 1/n\ (Var[h])$$

每个boostrap 样本集 D_k 是独立的

Random Forest(随机森林)

- 从决策树到随机森林





Random Forest(随机森林)

- 随机森林训练

Random Forest (RF)

random forest (RF) = bagging + fully-grown C&RT decision tree

function RandomForest($\mathcal{D}$)
For $t = 1, 2, \ldots, T$
1. request size-N' data $\tilde{\mathcal{D}}_t$ by bootstrapping with $\mathcal{D}$
2. obtain tree g_t by DTree($\tilde{\mathcal{D}}_t$)

return $G = \text{Uniform}(\{g_t\})$

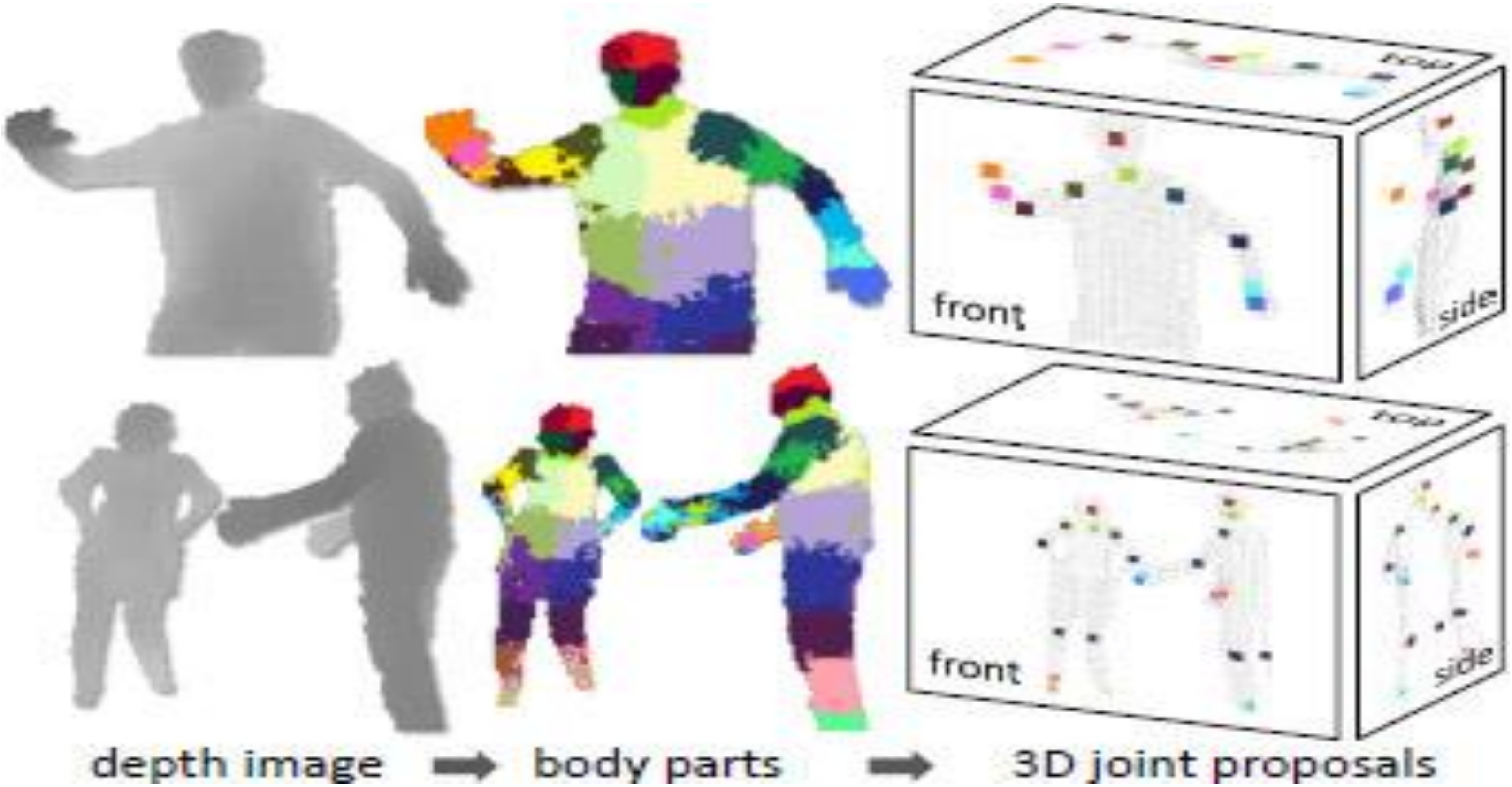
function DTree($\mathcal{D}$)
if termination return base g_t
else
1. learn $b(\mathbf{x})$ and split $\mathcal{D}$ to $\mathcal{D}_c$ by $b(\mathbf{x})$
2. build $G_c \leftarrow \text{DTree}(\mathcal{D}_c)$
3. return $G(\mathbf{x}) = \sum_{c=1}^{C} [\![b(\mathbf{x}) = c]\!] \, G_c(\mathbf{x})$

- highly **parallel/efficient** to learn
- **inherit pros** of C&RT
- **eliminate cons** of fully-grown tree

Slide Credit:
author of RF

Random Forest(随机森林)

- 最著名的应用: Kinect body pose estimation



Thank You

AI300学院