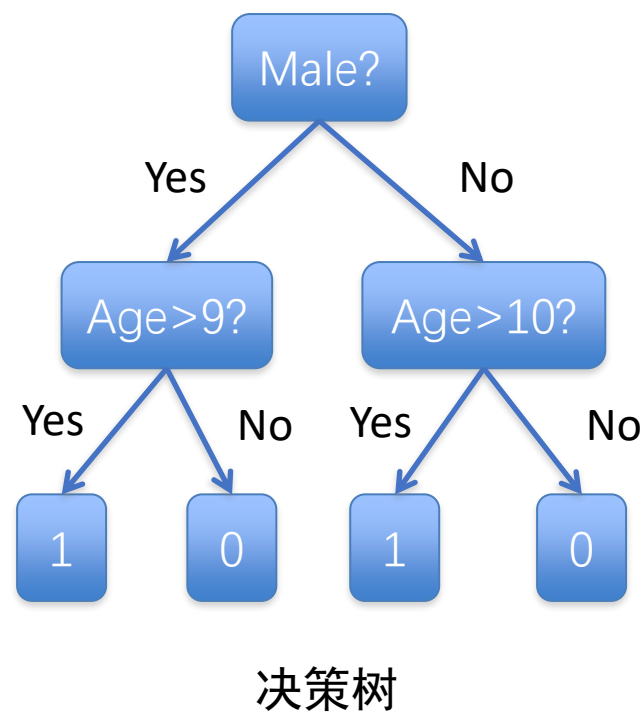


# 课程6： 集成学习

# 监督学习

- 目的：学习预测器 $h(x)$ 
  - 高精度（低误差）
  - 使用训练数据 $\{(x_1, y_1), \dots, (x_n, y_n)\}$

举例：判断身高是否大于145cm



Person	Age	Male?	Height > 145cm	
小张	14	0	1	✓
小刘	10	1	1	✓
小王	13	0	1	✓
小李	8	1	0	✓
小明	11	0	0	✗
小赵	9	1	1	✗
小钱	8	0	0	✓

$$x = \begin{bmatrix} \text{age} \\ 1_{[\text{gender}=\text{male}]} \end{bmatrix} \quad y = \begin{cases} 1 & \text{height} > 145 \\ 0 & \text{height} \leq 145 \end{cases}$$


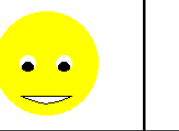
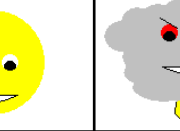


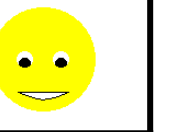




















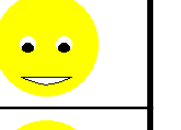






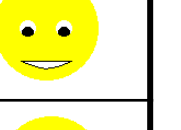



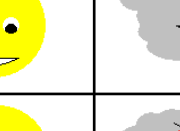


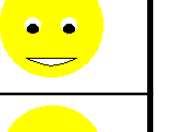





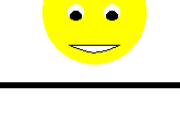
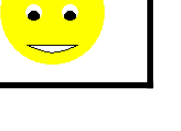

# 分类器

- 性能
  - 各个分类器均有优缺点，没有完美的分类器
  - 具有互补性
    - 一个分类器没有正确分类的例子可能会被其他分类器正确分类
- 如何改进这些分类器？
  - 利用互补的特性

# 将分类器集成

- 目的
  - 结合分类器以提高性能
- 分类器的集合体
  - 将不同分类器的分类结果结合起来，产生最终的输出
    - 非加权投票
    - 加权投票

举例：天气预测

Reality							
1							
2							
3							
4							
5							
Combine							

投票得到最终预测结果

# 主要内容

- 偏差/方差的权衡
- 使方差最小化的算法
  - Bagging
  - Random Forests
- 减少偏差的算法
  - Functional Gradient Descent
  - Boosting
  - Ensemble Selection

# 泛化误差

- 真实的分布:  $P(x,y)$ 
  - 一般未知
- 训练分类器:  $h(x) = y$ 
  - 使用训练数据  $S = \{(x_1, y_1), \dots, (x_n, y_n)\}$
  - 从 $P(x,y)$ 取样
- 泛化误差:
  - $l(h) = E_{(x,y) \sim P(x,y)}[f(h(x), y)]$
  - E.g.  $f(a,b) = (a-b)^2$



Person	Age	Male?	Height > 145cm
小冯	11	1	1
小陈	14	0	1
小张	14	0	1
小楚	12	0	1
小刘	10	1	1
小魏	9	1	0
张三	9	0	1
小王	13	0	1
李四	13	1	0
王五	12	1	1
Dave	8	1	0
Peter	9	1	0
Henry	13	1	0
小李	11	0	0
Rose	7	0	0
Iain	8	1	1
Paulo	12	1	0
Margaret	10	0	1
小明	9	1	1
Jill	13	0	0
Leon	10	1	0
Sarah	12	0	0
Gena	8	0	0
小钱	5	1	1

⋮

Person	Age	Male?	Height>145cm	
小张	14	0	1	✓
小刘	10	1	1	✓
小王	13	0	1	✓
小李	8	1	0	✓
小明	11	0	0	✗
小赵	9	1	1	✗
小钱	8	0	0	✓

$y$ 
 $h(x)$

**Generalization Error:**

$$l(h) = E_{(x,y) \sim P(x,y)} [f(h(x), y)]$$

## 偏差/方差的权衡

- 对  $h(x|S)$  有一个随机函数
  - 该随机函数取决于训练数据集S
- $L = E_S [E_{(x,y) \sim P(x,y)} [f(h(x), y)]]$ 
  - 预期的泛化误差

# 偏差/方差的权衡

- 损失函数:  $f(a, b) = (a - b)^2$
- 考虑一个数据点  $(x, y)$
- 符号表示:
  - $Z = h(x|S) - y$
  - $\hat{z} = E_S[Z]$
  - $Z - \hat{z} = h(x|S) - E_S[h(x|S)]$

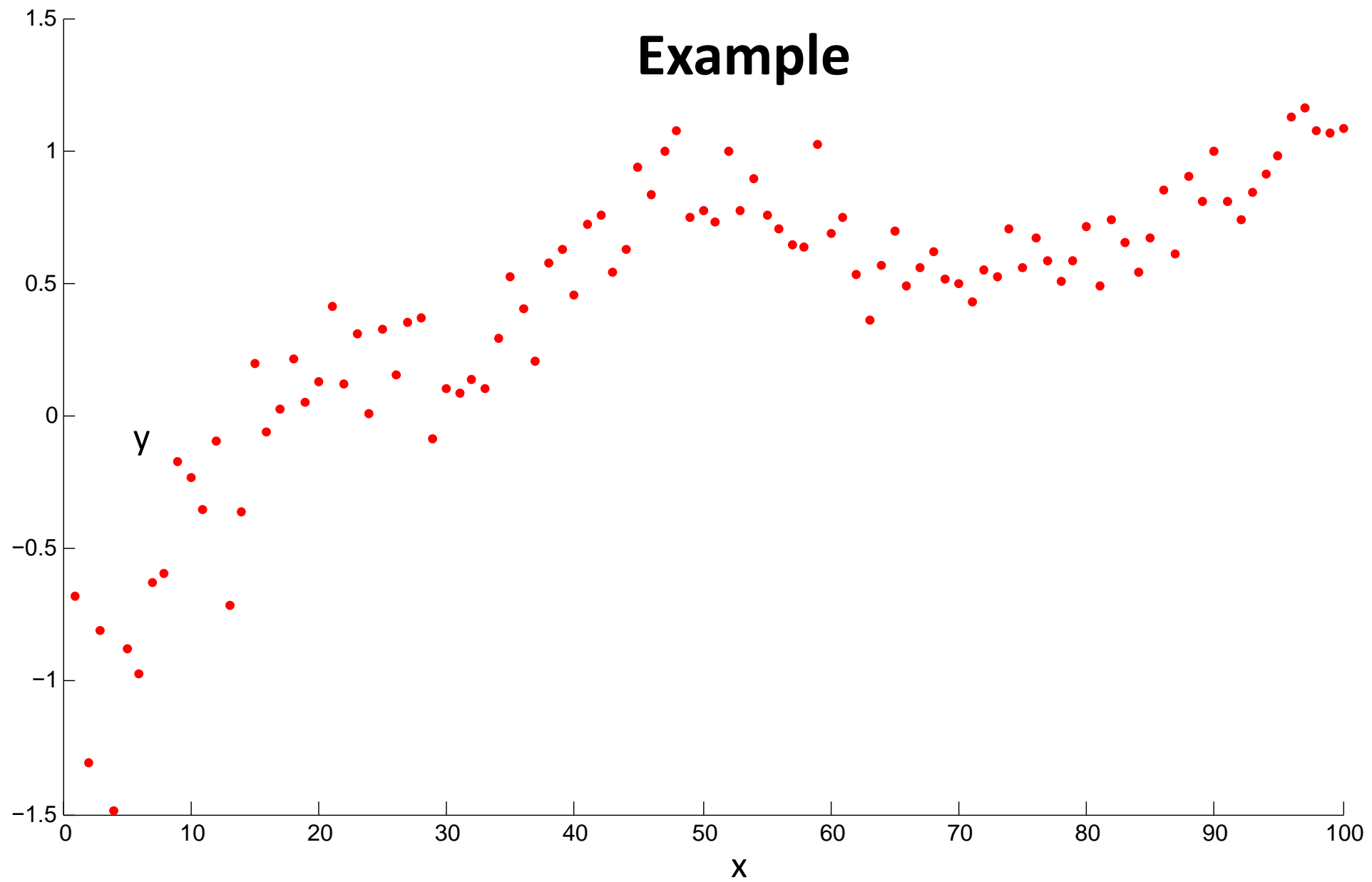
$$E_S[(Z - \hat{z})^2] = E_S[Z^2] - \hat{z}^2$$

所有  $(x, y)$  的偏差/方差是  $P(x, y)$  的期望值  
也可以将其归为噪声

**Expected Error**

$$E_S[f(h(x|S), y)] = E_S[Z^2]$$
$$= E_S[(Z - \hat{z})^2] + \hat{z}^2$$

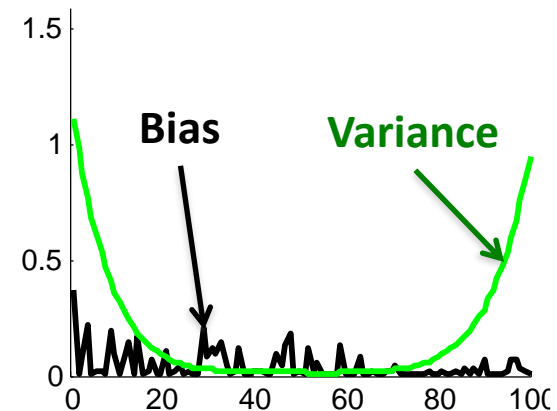
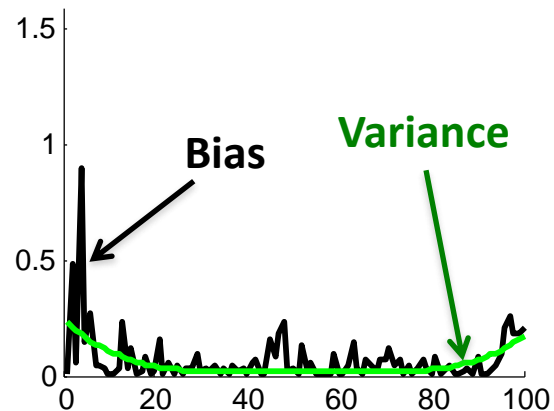
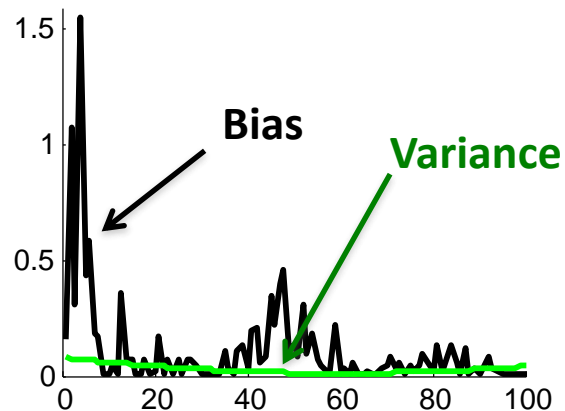
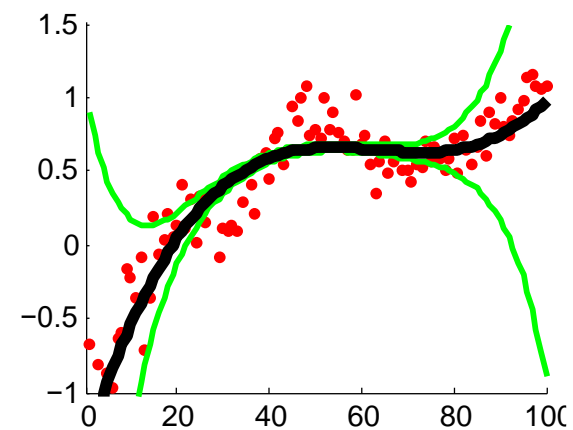
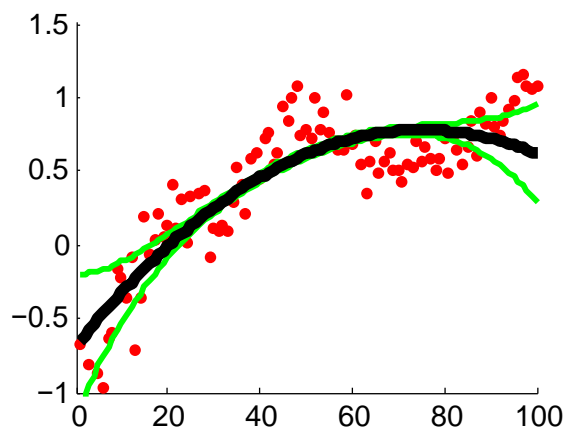
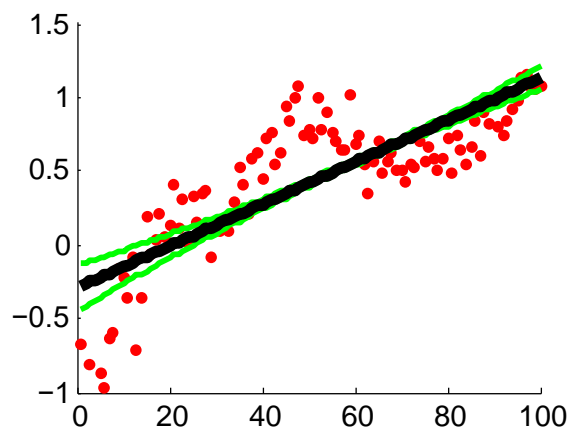
**Variance**      **Bias**



$$h(x|S)$$

$$h(x|S)$$

$$h(x|S)$$



$$E_S[f(h(x|S), y)] = E_S[(Z - \hat{z})^2] + \hat{z}^2$$

Expected Error

Variance

Bias

$$Z = h(x|S) - y$$

$$\hat{z} = E_S[Z]$$



# 主要内容

- 偏差/方差的权衡
- 使方差最小化的算法
  - Bagging
  - Random Forests
- 减少偏差的集成算法
  - Functional Gradient Descent
  - Boosting
  - Ensemble Selection

# Bagging

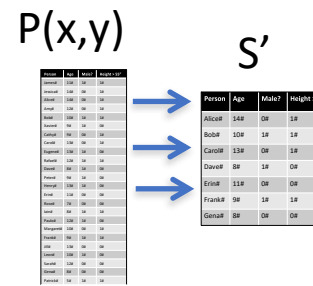
- 目的：减小方差

- 理想的条件：许多训练集  $S'$

- 使用每个  $S'$  来训练模型
- 平均预测结果

独立采样

方差线性减少，偏差不变



$$E_S[f(h(x|S), y)] = E_S[(Z - \hat{z})^2] + \hat{z}^2$$

Expected Error

Variance

Bias

$$Z = h(x|S) - y$$

$$\hat{z} = E_S[Z]$$

# Bagging

- 目的: 减小方差
- 在实际中: 用替换法重新取样  $S'$ 
  - 使用每个  $S'$  来训练模型
  - 平均预测结果

$S$                        $S'$

Person	Age	Male?	Height > 5'7"
Alison	188	0	0
Bob	120	1	1
Carol	128	0	0
David	98	1	0
Eve	118	0	0
Frank	98	1	1
Grace	98	0	0

→

Person	Age	Male?	Height > 5'7"
Alison	188	0	0
Bob	120	1	1
Carol	128	0	0
David	98	1	0
Eve	118	0	0
Frank	98	1	1
Grace	98	0	0

from  $S$

方差以Sublinear方式  
减少(因为 $S'$ 是相关的)  
偏差通常会略有增加

$$E_S[f(h(x|S), y)] = E_S[(Z - \hat{z})^2] + \hat{z}^2$$

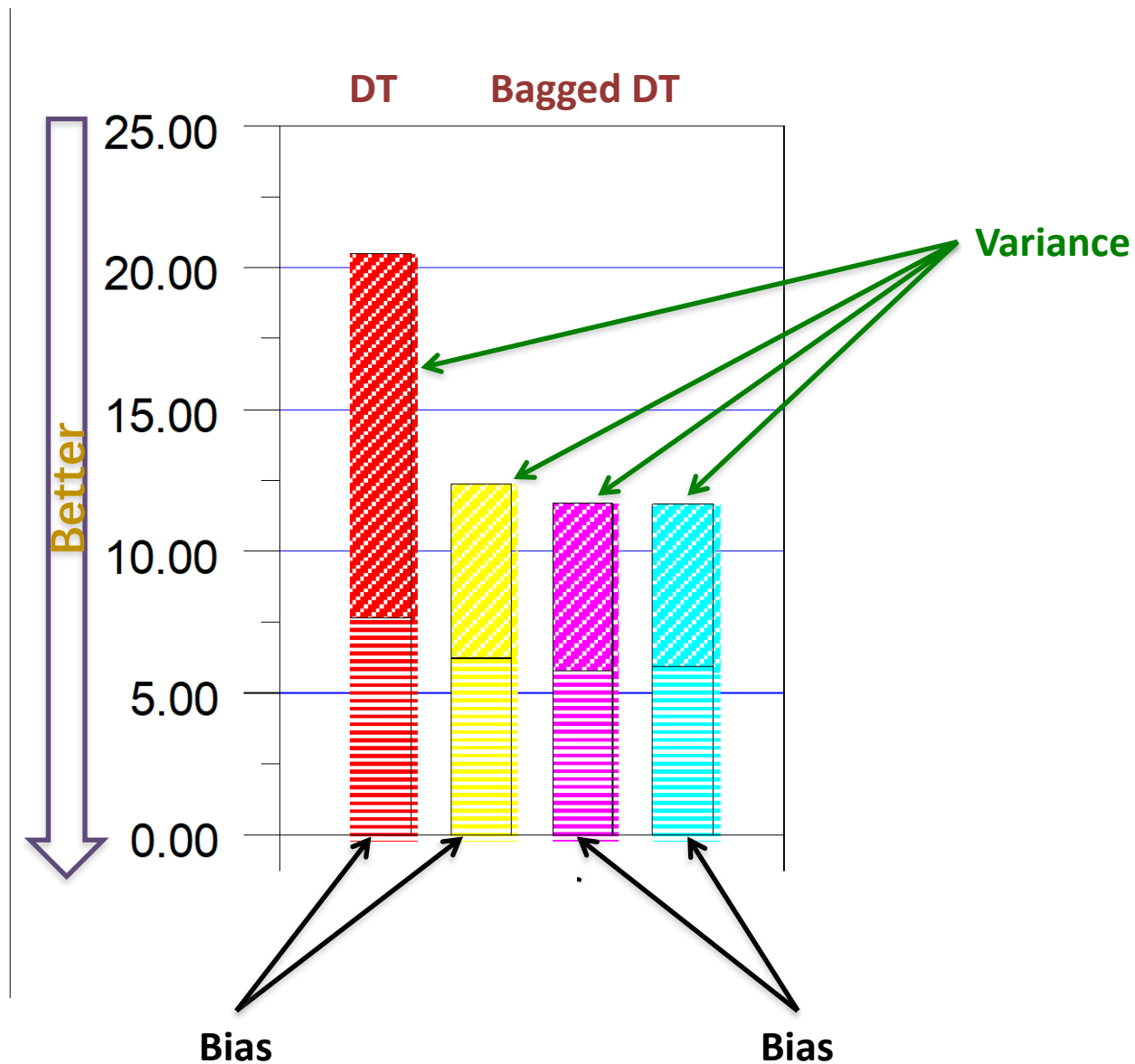
Expected Error

↑  
Variance

↑  
Bias

$$Z = h(x|S) - y$$

$$\hat{z} = E_S[Z]$$



**“An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants”**

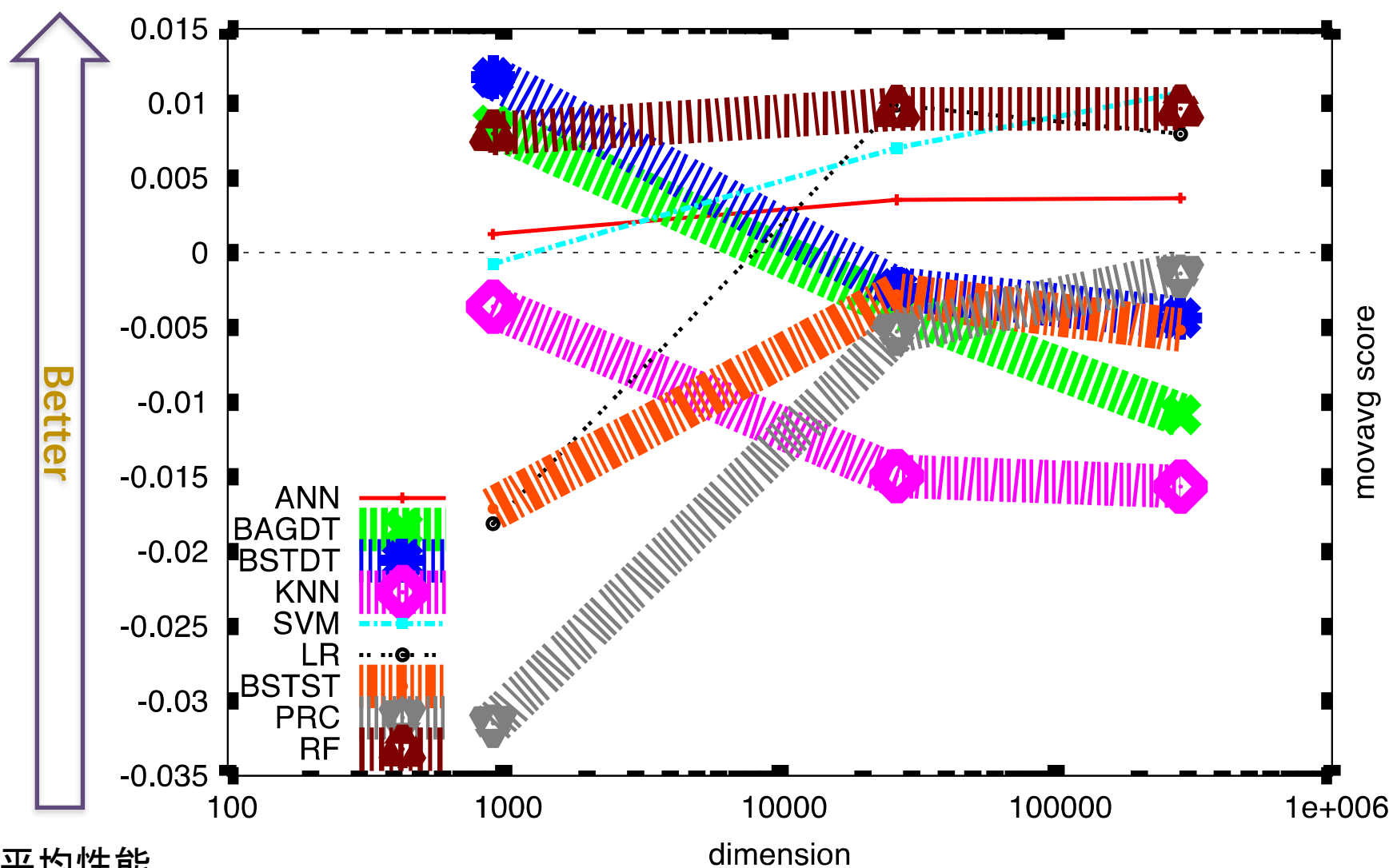
Eric Bauer & Ron Kohavi, Machine Learning 36, 105–139 (1999)

# Random Forests

- **目的：减小方差**
  - 重新取样训练数据的渐近线
- **Random Forests: 样本数据和特征**
  - 样品 $S'$
  - 训练 DT (decision tree)
    - 在每个节点上, 采样特征 (sqrt)
  - 平均预测结果

进一步消除关联性





多个数据集的平均性能  
其中随机森林的表现最好

“An Empirical Evaluation of Supervised Learning in High Dimensions”

Caruana, Karampatziakis & Yessenalina, ICML 2008

# Structured Random Forests

- DTs通常在二分类任务上训练（标签为一元的 0/1）
- 如果是结构化的标签呢？
  - 必须定义结构化标签的信息增益
- 边缘检测：
  - E. g. 结构化标签是一个16x16的图像
  - 将结构化标签映射到另一个空间

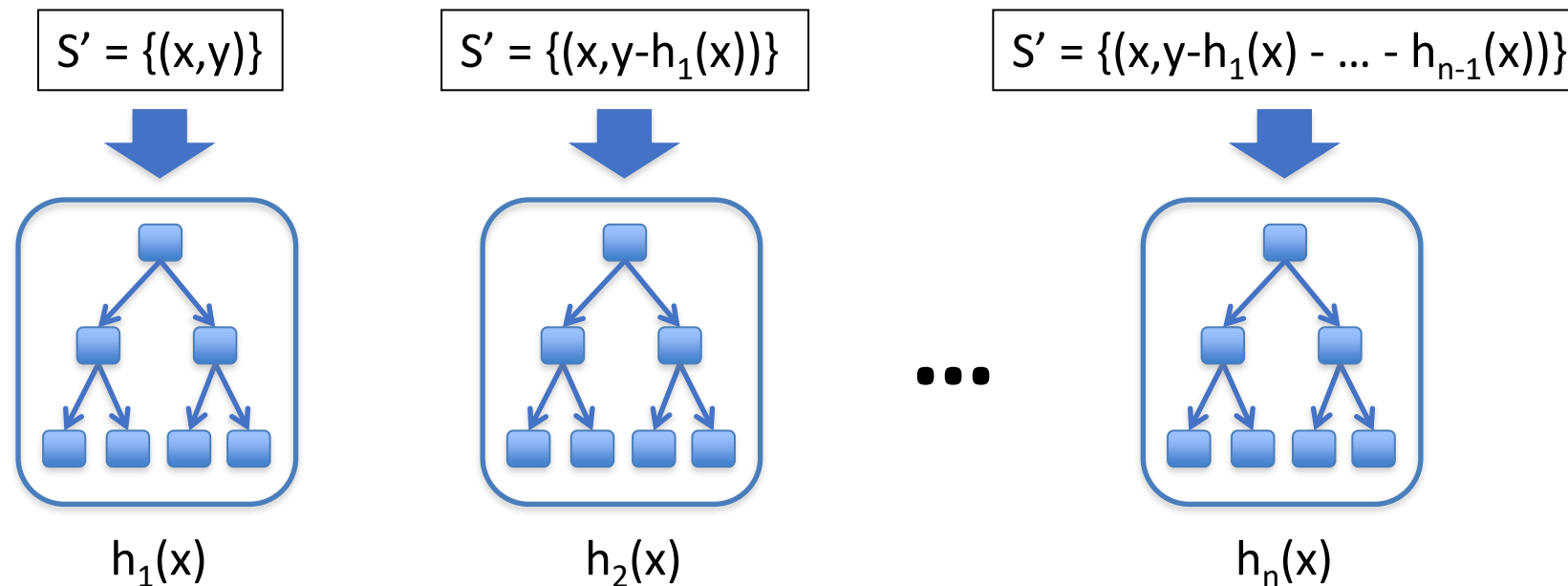
# 主要内容

- 偏差/方差的权衡
- 使方差最小化的算法
  - Bagging
  - Random Forests
- 减少偏差的算法
  - Functional Gradient Descent
  - Boosting
  - Ensemble Selection



# Functional Gradient Descent

$$h(x) = h_1(x) + h_2(x) + \dots + h_n(x)$$



# Coordinate Gradient Descent

- 学习  $w$ , 使  $h(x) = w^T x$
- Coordinate descent
  - 初始化  $w = 0$
  - 选择具有最高增益的维度
    - 设置  $w$  的分量
  - 重复进行上述步骤

# Coordinate Gradient Descent

- 学习  $w$ , 使  $h(x) = w^T x$
- Coordinate descent
  - 初始化  $w = 0$
  - 选择具有最高增益的维度
    - 设置  $w$  的分量
  - 重复进行上述步骤

$$w = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

# Coordinate Gradient Descent

- 学习  $w$ , 使  $h(x) = w^T x$
- Coordinate descent
  - 初始化  $w = 0$
  - 选择具有最高增益的维度
    - 设置  $w$  的分量
  - 重复进行上述步骤

$$w = \begin{pmatrix} 0 \\ 0 \\ 0 \\ +3 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

# Coordinate Gradient Descent

- 学习  $w$ , 使  $h(x) = w^T x$
- Coordinate descent
  - 初始化  $w = 0$
  - 选择具有最高增益的维度
    - 设置  $w$  的分量
  - 重复进行上述步骤

$$w = \begin{pmatrix} 0 \\ 0 \\ 0 \\ +3 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1.5 \\ 0 \\ 0 \end{pmatrix}$$

# Coordinate Gradient Descent

- 学习  $w$ , 使  $h(x) = w^T x$
- Coordinate descent
  - 初始化  $w = 0$
  - 选择具有最高增益的维度
    - 设置  $w$  的分量
  - 重复进行上述步骤

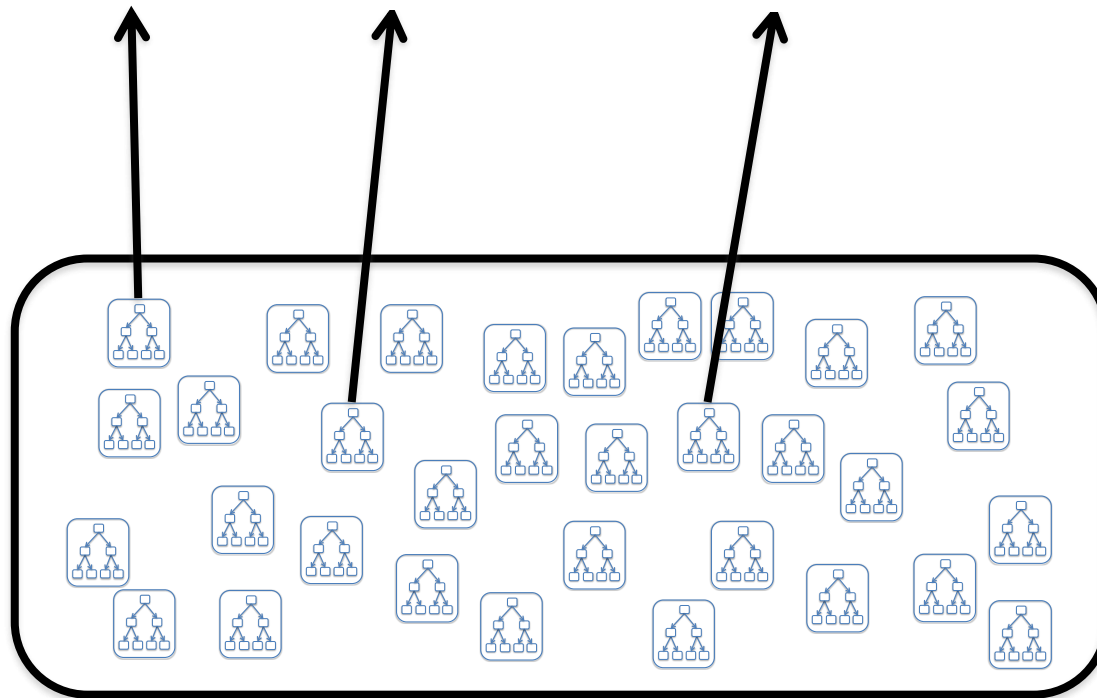
$$w = \begin{pmatrix} +2.1 \\ 0 \\ 0 \\ +3 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1.5 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

# Coordinate Gradient Descent

- 学习  $w$ , 使  $h(x) = w^T x$
- Coordinate descent
  - 初始化  $w = 0$
  - 选择具有最高增益的维度
    - 设置  $w$  的分量
  - 重复进行上述步骤

$$w = \begin{pmatrix} +2.1 \\ 0 \\ +3 \\ 0 \\ 0 \\ -0.9 \\ 0 \\ -1.5 \\ 0 \\ 0 \end{pmatrix}$$

# Functional Gradient Descent



$$h(x) = h_1(x) + h_2(x) + \dots + h_n(x)$$

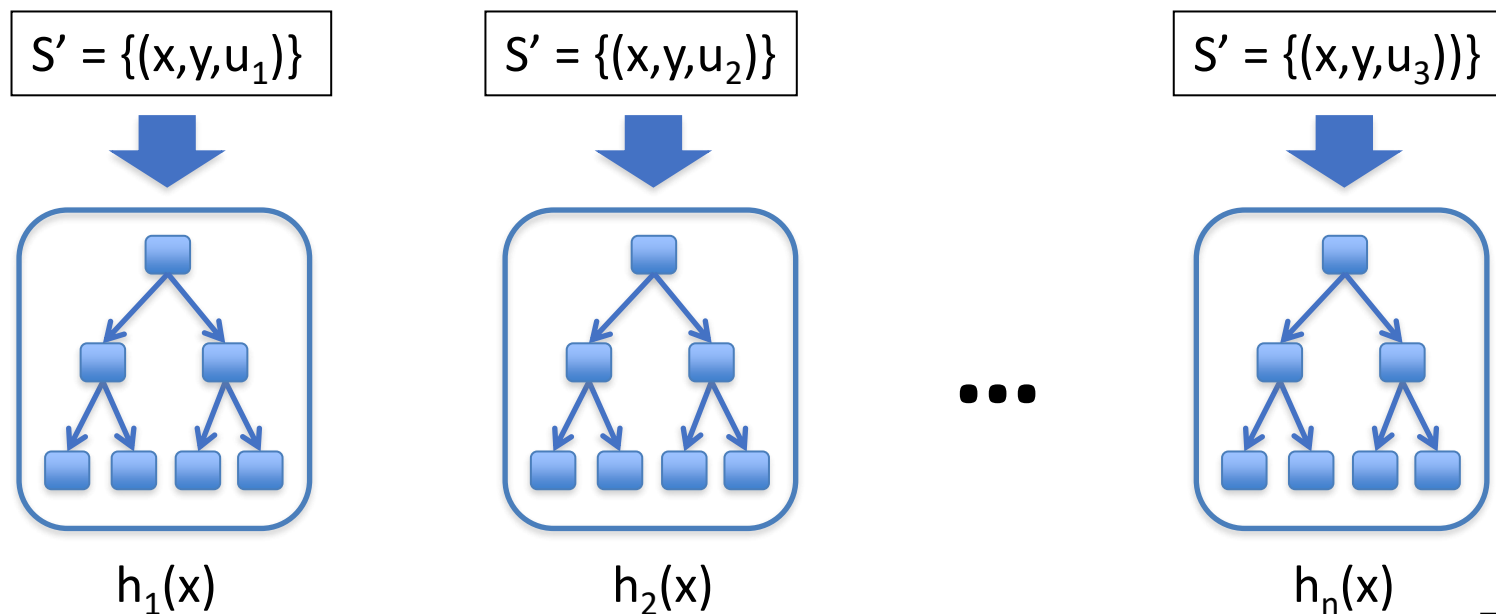
函数空间中的坐标下降法  
权重限制为0, 1, 2, ...

**“Function Space”  
(All possible DTs)**



# Boosting (AdaBoost)

$$h(x) = a_1 h_1(x) + a_2 h_2(x) + \dots + a_n h_n(x)$$



$u$  - 对数据点进行加权  
 $a$  - 线性组合的权重

当验证性能趋于平稳时  
停止  
(后面将进行讨论)

Given:  $(x_1, y_1), \dots, (x_m, y_m)$  where  $x_i \in \mathcal{X}$ ,  $y_i \in \{-1, +1\}$ .

Initialize  $D_1(i) = 1/m$  for  $i = 1, \dots, m$ .

← 权重初始化

For  $t = 1, \dots, T$ :

- Train weak learner using distribution  $D_t$ .
- Get weak hypothesis  $h_t : \mathcal{X} \rightarrow \{-1, +1\}$ .
- Aim: select  $h_t$  with low weighted error:

← 训练模型

$$\epsilon_t = \Pr_{i \sim D_t} [h_t(x_i) \neq y_i].$$

← 模型的预测误差

- Choose  $\alpha_t = \frac{1}{2} \ln \left( \frac{1 - \epsilon_t}{\epsilon_t} \right)$ .

← 模型的系数

- Update, for  $i = 1, \dots, m$ :

$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

← 更新前权重分布

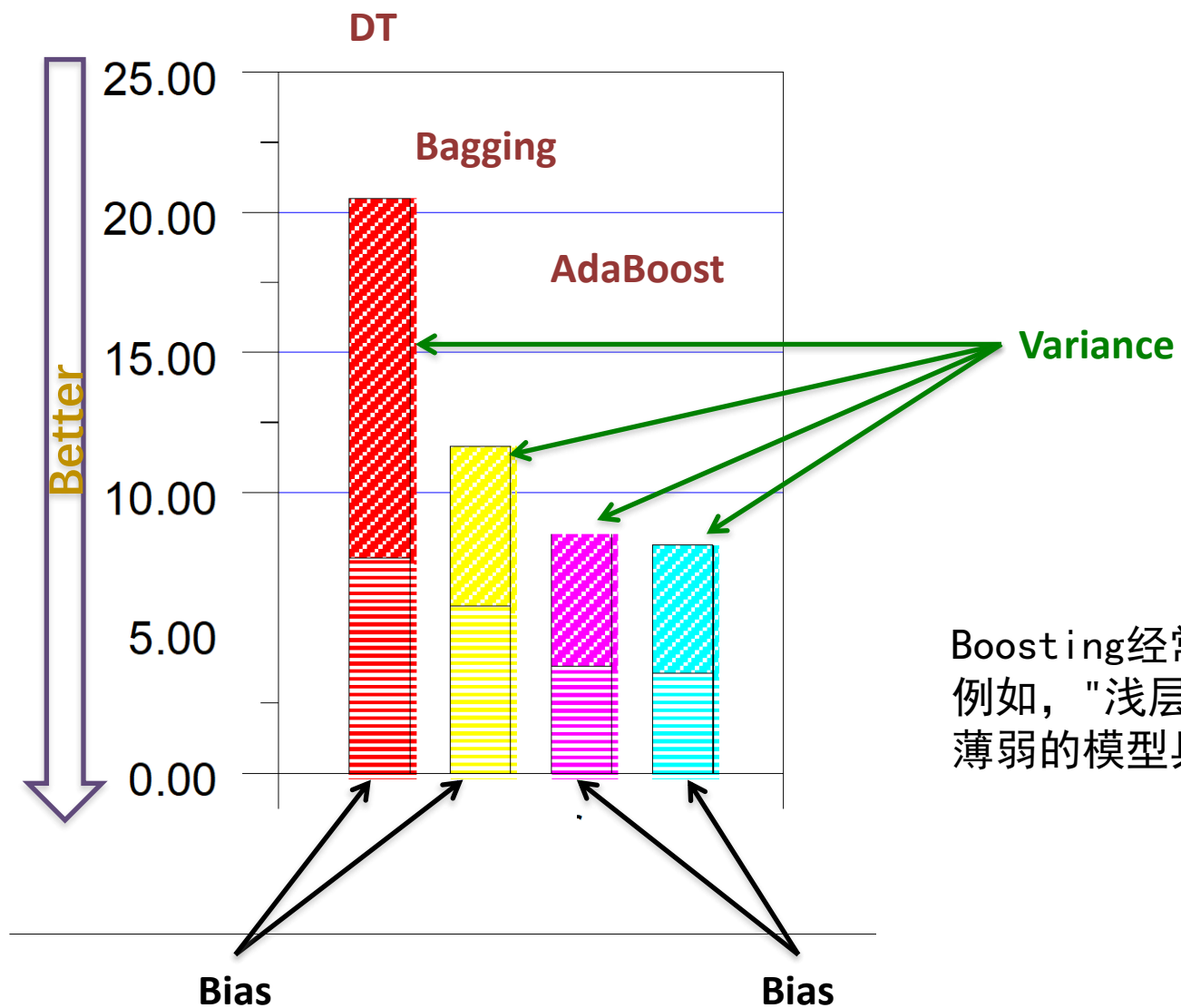
where  $Z_t$  is a normalization factor (chosen so that  $D_{t+1}$  will be a distribution).

Output the final hypothesis:

$$H(x) = \text{sign} \left( \sum_{t=1}^T \alpha_t h_t(x) \right).$$

← 最后再进行平均

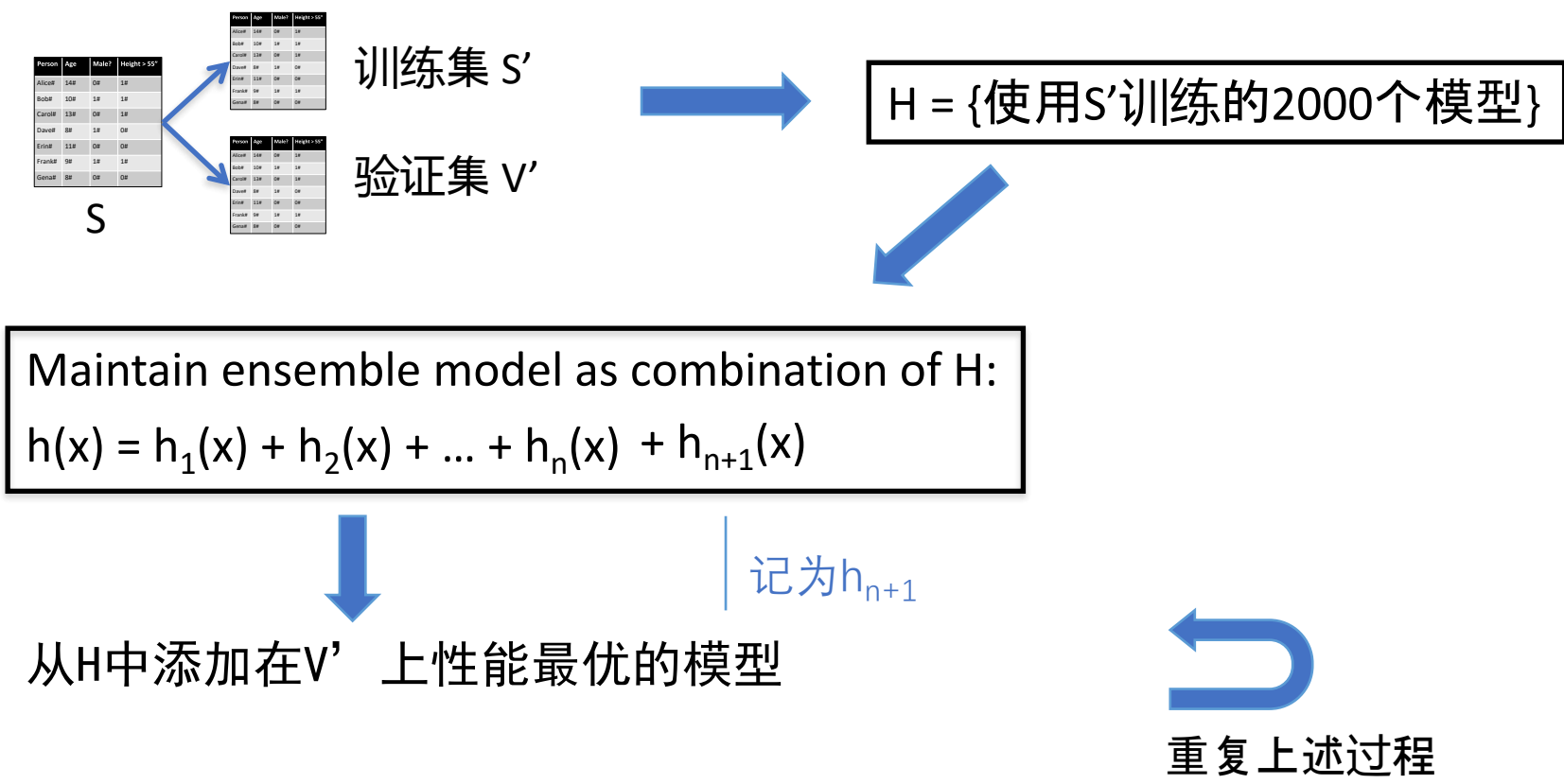
训练误差呈指数级快速下降



Boosting经常使用薄弱的模型  
例如, "浅层" 决策树  
薄弱的模型具有较低的方差

“An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants”  
Eric Bauer & Ron Kohavi, Machine Learning 36, 105–139 (1999)

# Ensemble Selection



方法	减小偏差	减小方差	评价
Bagging	复杂模型类 (deep DTs)	Bootstrap aggregation (重新取样训练数据)	对简单的模型不起作用。
Random Forests	复杂模型类 (Deep DTs)	Bootstrap aggregation + bootstrapping features	只适用于决策树。
Gradient Boosting (AdaBoost)	优化训练性能	简单的模型类 (Shallow DTs)	决定在运行时添加哪个 模型
Ensemble Selection	优化验证性能	优化验证性能	在训练集上学习的预先 指定的模型

- State-of-the-art prediction performance
  - Won Netflix Challenge
  - Won numerous KDD Cups
  - Industry standard

## 参考文献:

1. E. Bauer, R. Kohavi. An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants[J]. Machine Learning, 1999(36):105-139.
2. T. D. Dietterich, F. Roli. Ensemble Methods in Machine Learning[J]. Multiple Classifier Systems, 2000:1-15
3. R. Caruana, A. Niculescu-Mizil, G. Crew and et al. Ensemble selection from libraries of models[C]. Proceedings of the International Conference on Machine Learning (ICML), 2014.
4. R. Caruana, A. Munson, A. Niculescu-Mizil. Getting the Most Out of Ensemble Selection[C]. Proceedings of the 6th IEEE international Conference on Data Mining(ICDM), 2006:828-833.
5. P. Dollar, C. L. Zitnick. Structured Forests for Fast Edge Detection[C]. Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2013:1841-1848.

相关论文会放到课程网页中，如有需要请自行下载。