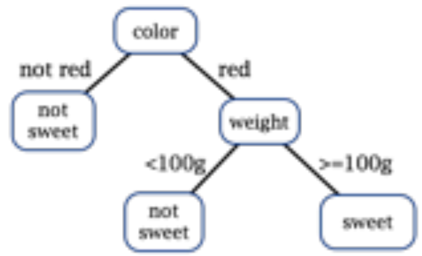
Lecture 7: Ensemble Methods

http://cs.nju.edu.cn/yuy/course_dm12.ashx

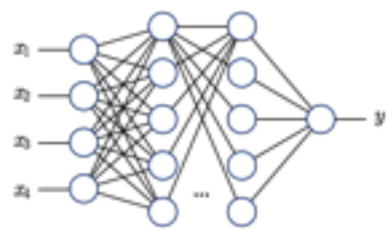
A summary of learning algorithms

decision tree



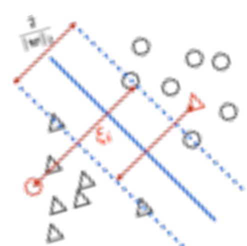
fast training & testing
moderate accuracy
comprehensible
nominal + numerical feature

neural networks



slow training & testing
high accuracy
not comprehensible
numerical feature

linear models



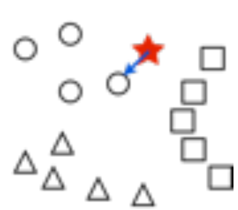
fast with linear kernel
high accuracy (with good kernel)
numerical feature

Bayes classifiers

$$P(y \mid x) = \frac{P(x \mid y)P(y)}{P(x)}$$

fast training
moderate accuracy (high for semi-naive)
nominal feature

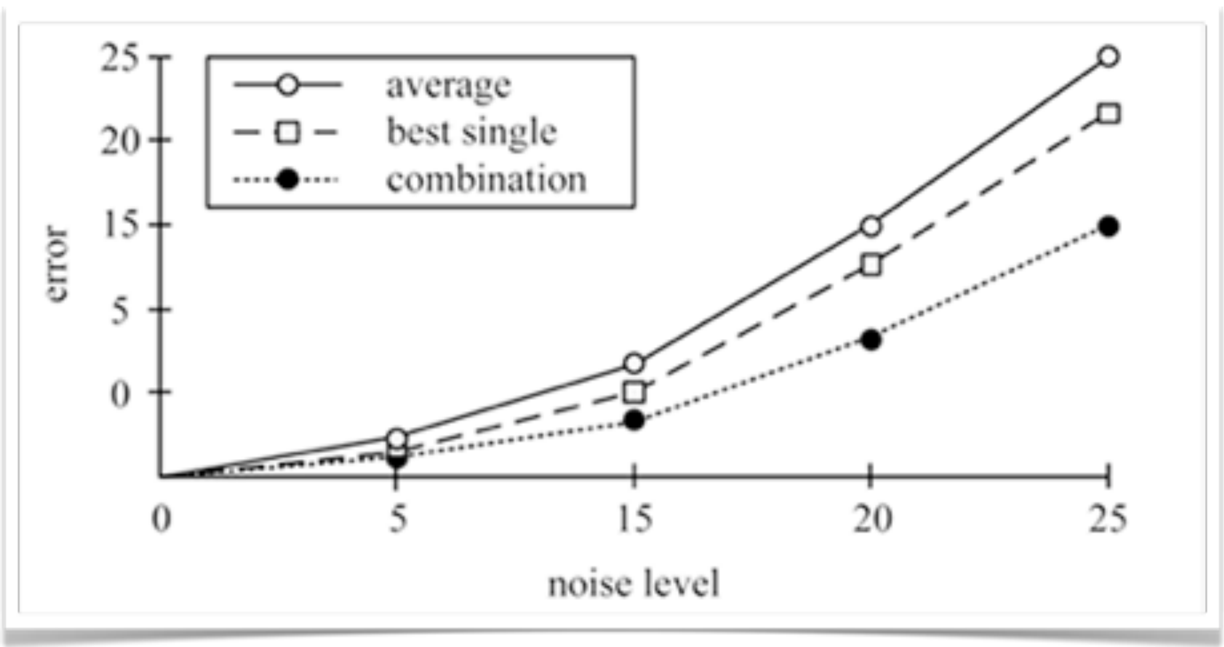
lazy classifiers



fast training + slow testing
moderate accuracy
numerical feature

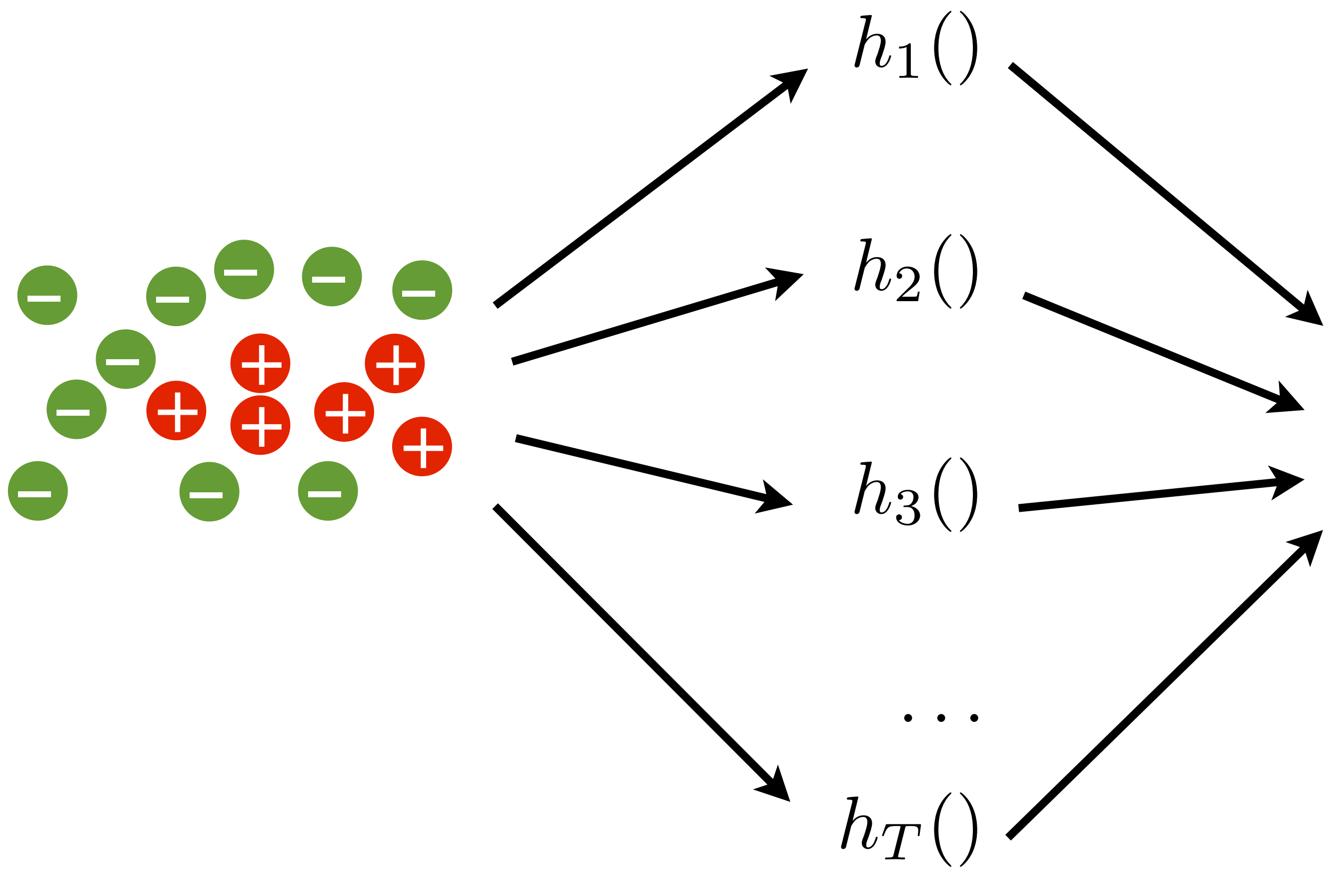
How can we improve one algorithm

Hansen and Salamon [PAMI'90] reported an observation that combination of multiple BP-NN is better than the best single BP-NN



Ensemble learning

combination of multiple classifiers/regressors



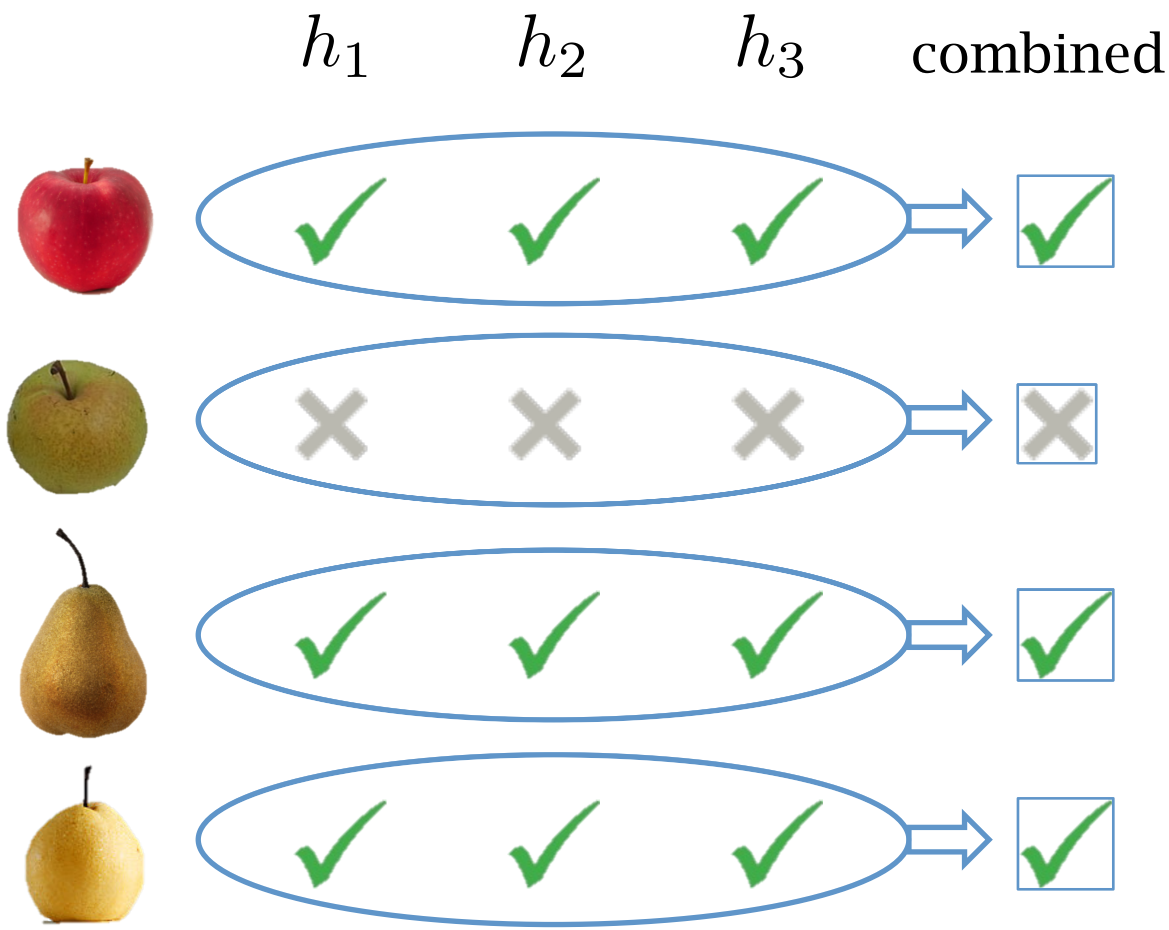
$$H() = \sum_{t=1}^{T} \alpha_t h_t()$$

base learner

combined learner

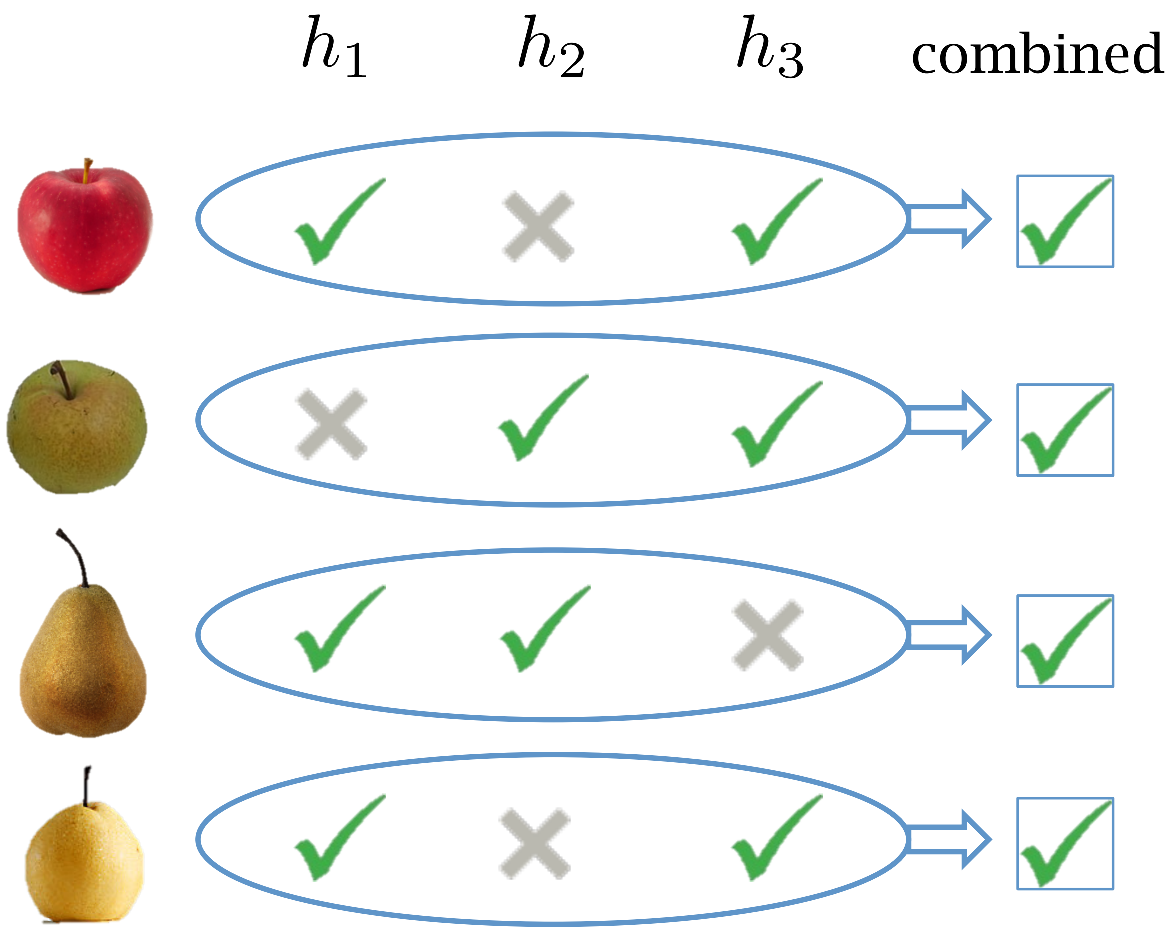
What base classifiers should be?

not useful to combine identical base learners



What base classifiers should be?

good to combine different base learners



Motivation theories

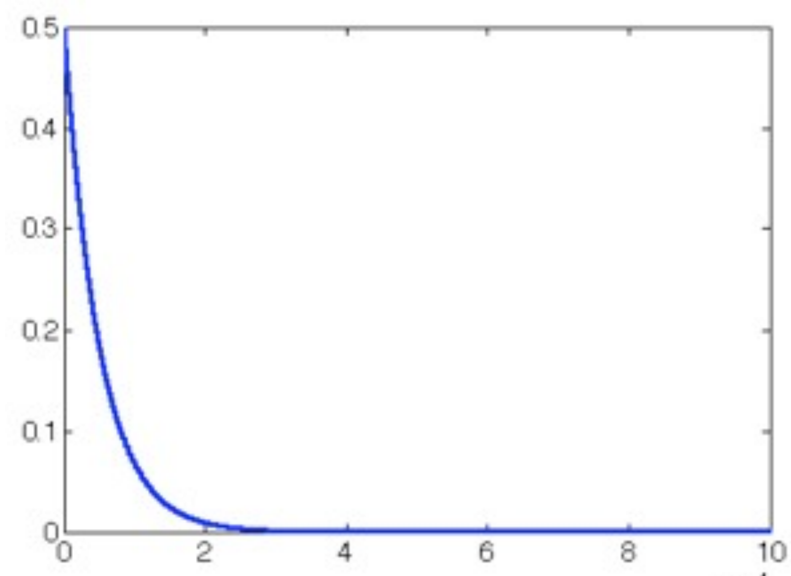
for binary classification, what if the classifiers give *independent* output and are little bit better than random guess?

each classifier has error 0.49
error of combining T classifiers:

$$\sum_{t=\lceil T/2 \rceil}^{T} \binom{T}{t} \cdot 0.49^{t} \cdot 0.51^{T-t}$$

$$\leq \frac{1}{2} e^{-2T(0.5-0.49)^2}$$

error



T

Motivation theories

for regression task:

mean error of base regressors

$$\frac{1}{T}\sum_t (h_t - f)^2$$

$$= \frac{1}{T}\sum_t (h_t - H + H - f)^2$$

$$= \frac{1}{T}\sum_t (h_t - H)^2 + \frac{1}{T}\sum_t (H - f)^2 - 2\frac{1}{T}\sum_t (h_t - H)(H - f)$$

$$= \frac{1}{T}\sum_t (h_t - H)^2 + (H - f)^2$$

error of combined regressor

mean difference to the combined regressor

error of ensemble =
mean error of base regressors
– mean difference base regressors to the ensemble

accurate and *diverse*

Ensemble methods

Parallel ensemble

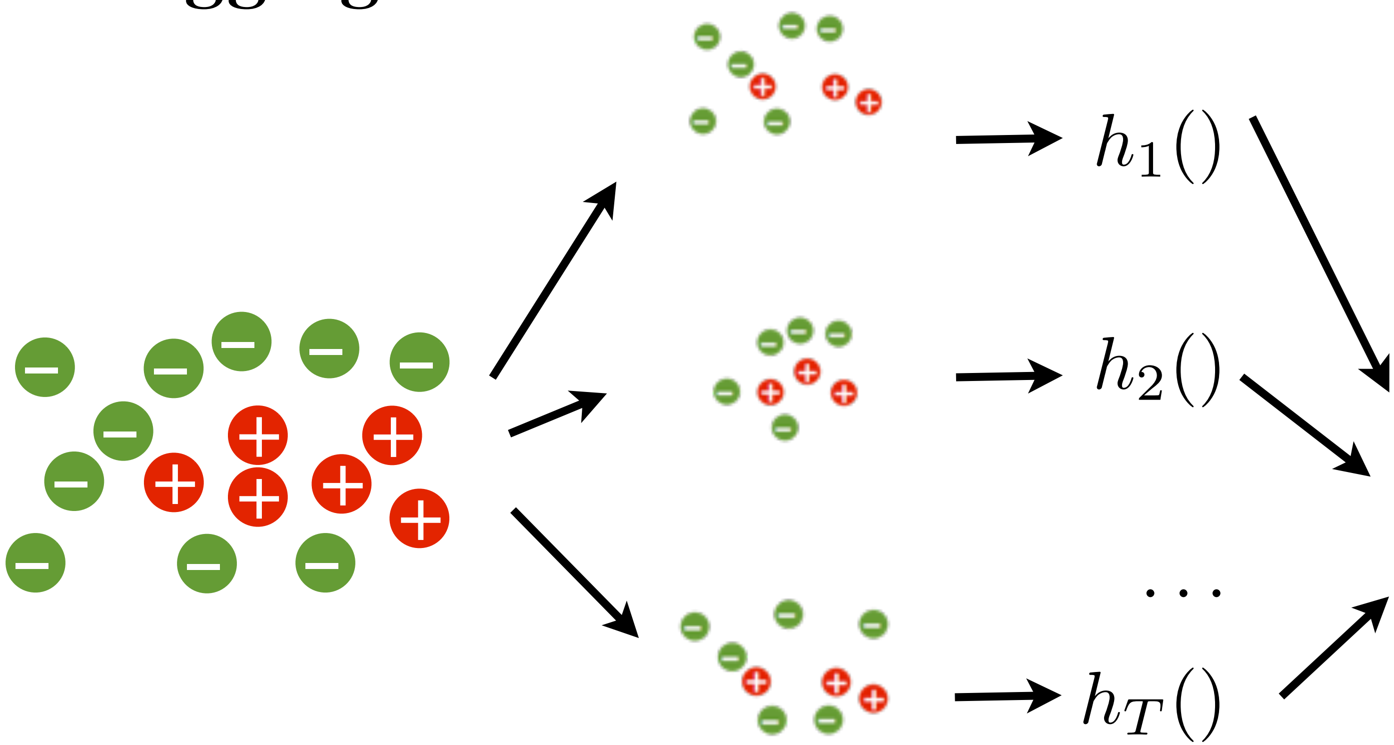
create diverse base learners by introducing randomness

Sequential ensemble

create base learners by complementarity

Parallel ensemble methods

Bagging



randomly sample data

$$H(\boldsymbol{x}) =$$

$$\frac{1}{T}\sum_{t=1}^{T} h_t(\boldsymbol{x})$$ (regression)

$$\arg\max_y \sum_{t=1}^{T} I(h_t(\boldsymbol{x}) = y)$$

(classification)

Base classifiers should be sensitive to sampling

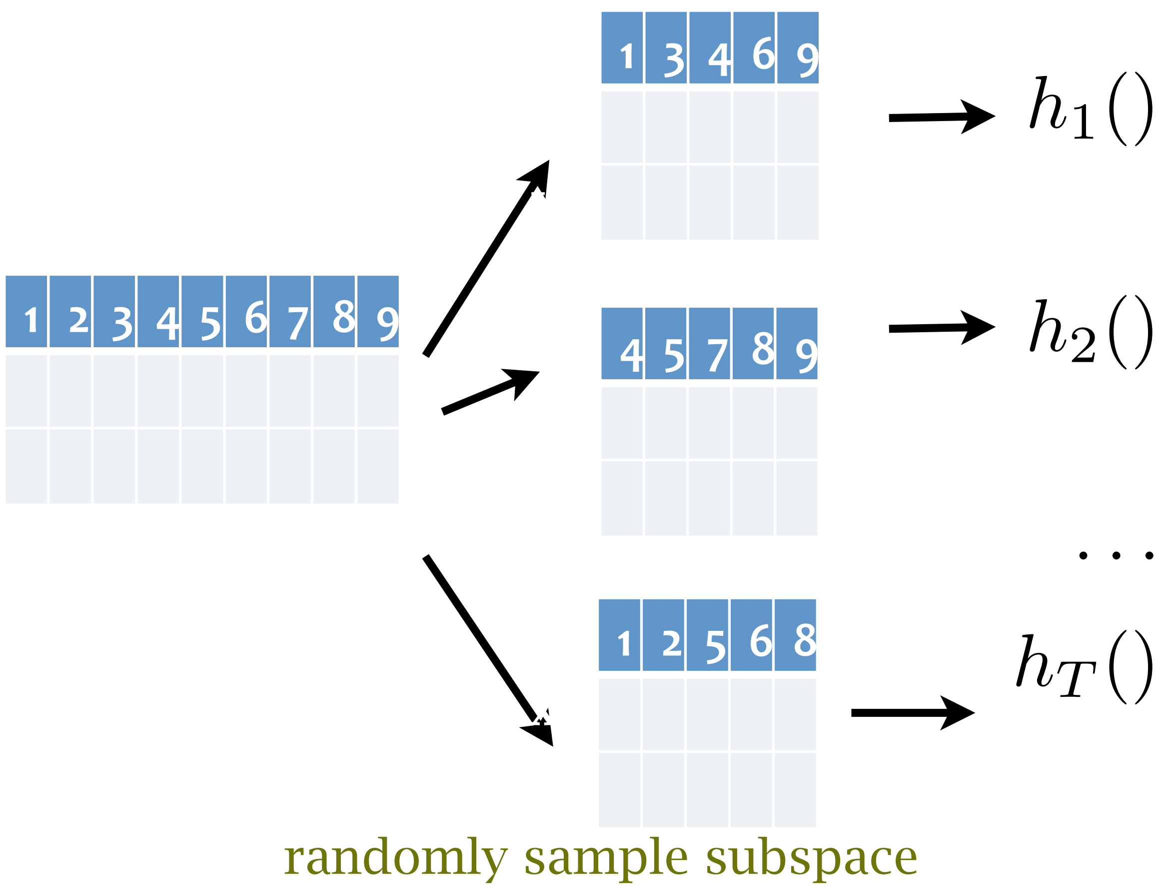
» decision tree, neural network are good

» NB, linear classifier are not

Good for handling large data set

Parallel ensemble methods

Random subspace



$H(\boldsymbol{x}) =$

$$\frac{1}{T}\sum_{t=1}^{T} h_t(\boldsymbol{x}) \quad \text{(regression)}$$

$$\arg\max_{y} \sum_{t=1}^{T} I(h_t(\boldsymbol{x}) = y)$$

(classification)

Data should be rich in features

Good for handling high dimensional data

Parallel ensemble methods

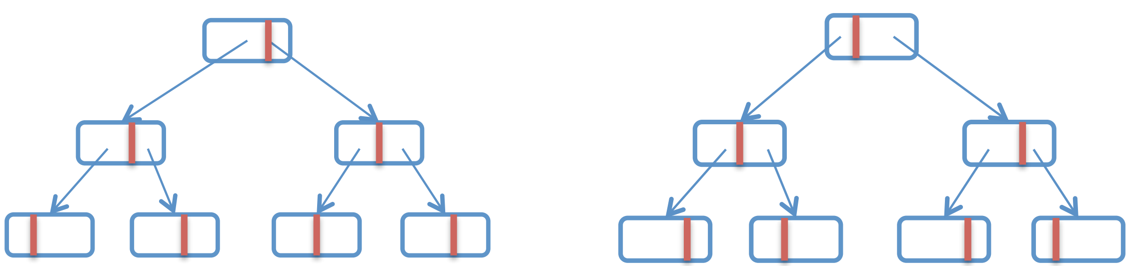
Random forest

Randomized decision tree

at each node

1. randomly select a subset of features
2. use C4.5 method to select a feature (and split point) from the subset to split the data

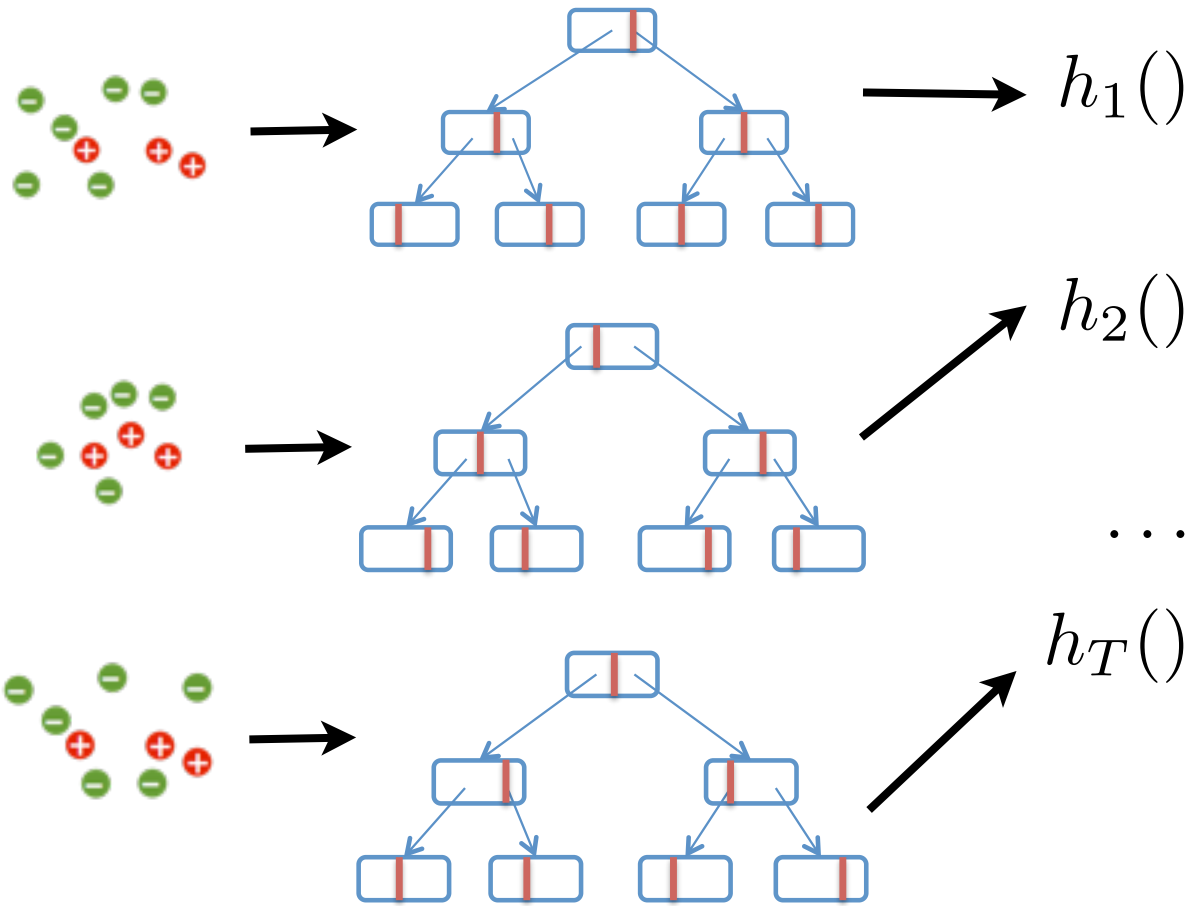
(other variants are available)



every run produce a different tree

Parallel ensemble methods

Random forest



randomly sample data

randomized decision tree

$$H(\boldsymbol{x}) =$$

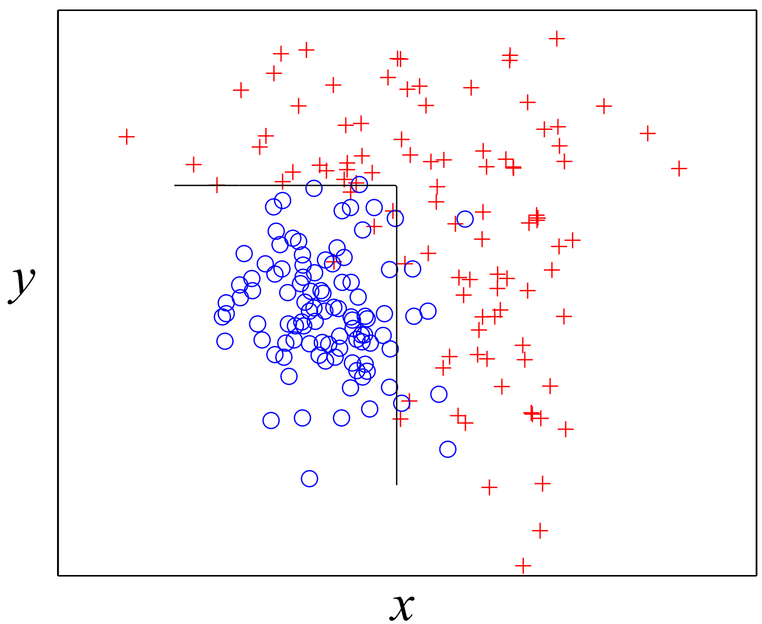
$$\frac{1}{T}\sum_{t=1}^{T} h_t(\boldsymbol{x}) \text{ (regression)}$$

$$\arg\max_y \sum_{t=1}^{T} I(h_t(\boldsymbol{x}) = y)$$

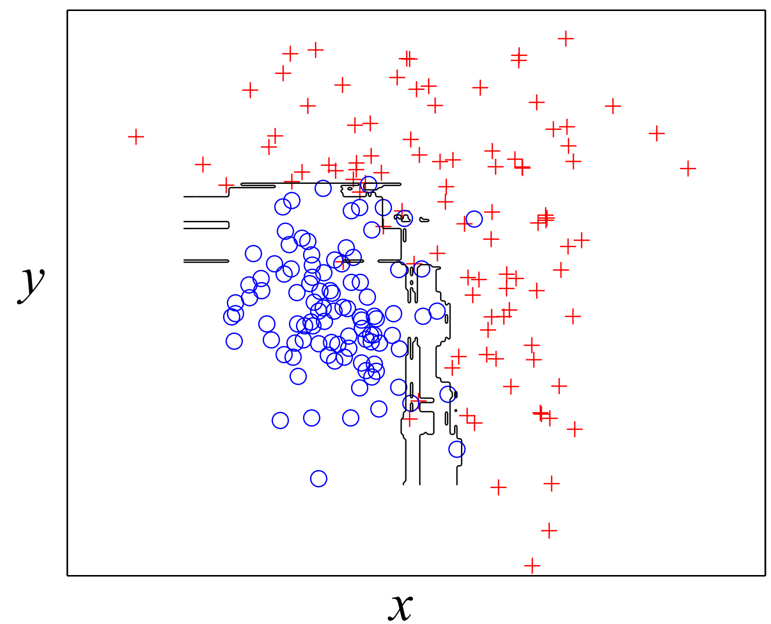
(classification)

Parallel ensemble methods

Random forest



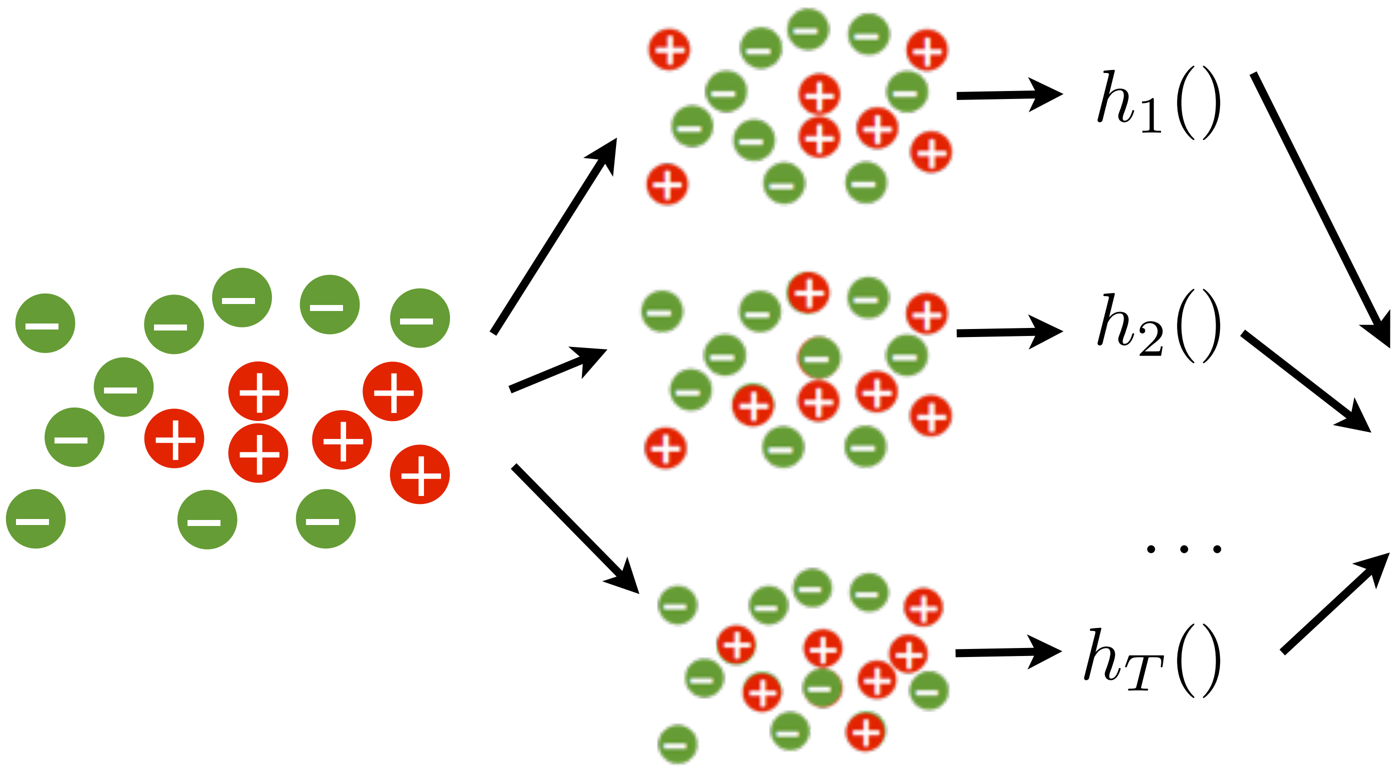
decision boundary of single decision tree



decision boundary of random forest

Parallel ensemble methods

Output flipping



randomly modify the label

$H(\boldsymbol{x}) =$

$\frac{1}{T}\sum_{t=1}^{T} h_t(\boldsymbol{x})$ (regression)

$\arg\max_{y} \sum_{t=1}^{T} I(h_t(\boldsymbol{x}) = y)$

(classification)

May drastically reduce the accuracy of base learners

Parallel ensemble methods

Diversity generating categories:

- Data Sample Manipulation
  - bootstrap sampling/Bagging
- Input Feature Manipulation
  - random subspace
- Learning Parameter Manipulation
  - random initialization
  - Random Forests
- Output Representation Manipulation
  - flipping output/output smearing

combine two or more categories

Sequential ensemble methods

fit an additive model, sequentially

$$H() = \sum_{t=1}^{T} \alpha_t h_t()$$

1. every h_t is a weak learner (better than random)

2. every is to complement its predecessors

Sequential ensemble methods

example: least square regression

$$\min \frac{1}{m}\sum_{i=1}^{m}(H(\boldsymbol{x}_i) - y_i)^2$$

1. fit the first base regressor

$$\min \frac{1}{m}\sum_{i=1}^{m}(h_1(\boldsymbol{x}_i) - y_i)^2$$

then how to train the second base regressor ?

$$\min \frac{1}{m}\sum_{i=1}^{m}(h_1(\boldsymbol{x}_i)+h_2(\boldsymbol{x}_i) - y_i)^2$$

gradient descent *in function space*

Sequential ensemble methods

$$\min \frac{1}{m}\sum_{i=1}^{m}(h_1(\boldsymbol{x}_i)+h_2(\boldsymbol{x}_i)-y_i)^2$$

gradient descent *in function space*

$$h_{\text{new}} \leftarrow -\frac{\partial (H-f)^2}{\partial H} = -2(H-f)$$

this function is not directly operable

operate through data

$$\forall \boldsymbol{x}_i : \hat{y}_i = -2(H(\boldsymbol{x}_i)-y_i)$$

fit h_2 point-wisely

$$h_{\text{new}} = \arg\min_h \frac{1}{m}\sum_{i=1}^{m}(h(\boldsymbol{x}_i)-\hat{y}_i)^2$$

Sequential ensemble methods

Gradient boosting (for least square regression)

1. $h_0 = 0, H_0 = h_0$
2. For $t = 1$ to T
3. let $\forall \boldsymbol{x}_i : y_i = -2(H_{t-1}(\boldsymbol{x}_i) - y_i)$
4. solve $h_t = \arg\min_h \frac{1}{m} \sum_{i=1}^{m} (h(\boldsymbol{x}_i) - y_i)^2$

   (by some least square regression algorithm)
5. $H_t = H_{t-1} + \eta h_t$ (usually set $\eta = 0.01$)
6. next for

Output $H_T = \sum_{t=1}^{T} h_t$

Sequential ensemble methods

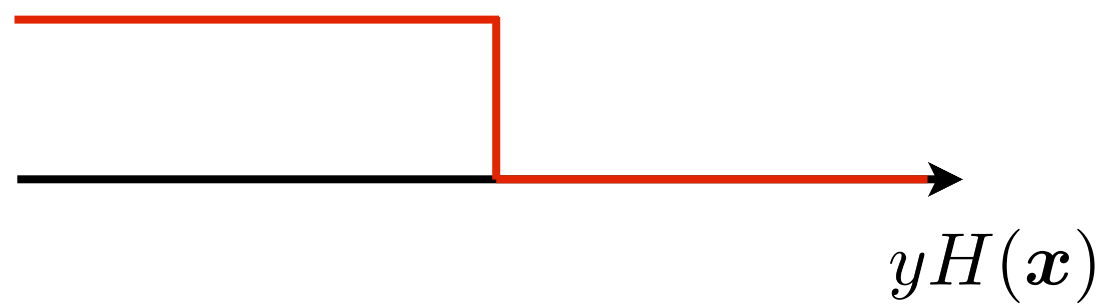
Gradient boosting (for classification)

Sequential ensemble methods

Gradient boosting (for classification)

0-1 loss

$\min I(yH(\boldsymbol{x}) \leq 0)$



Sequential ensemble methods

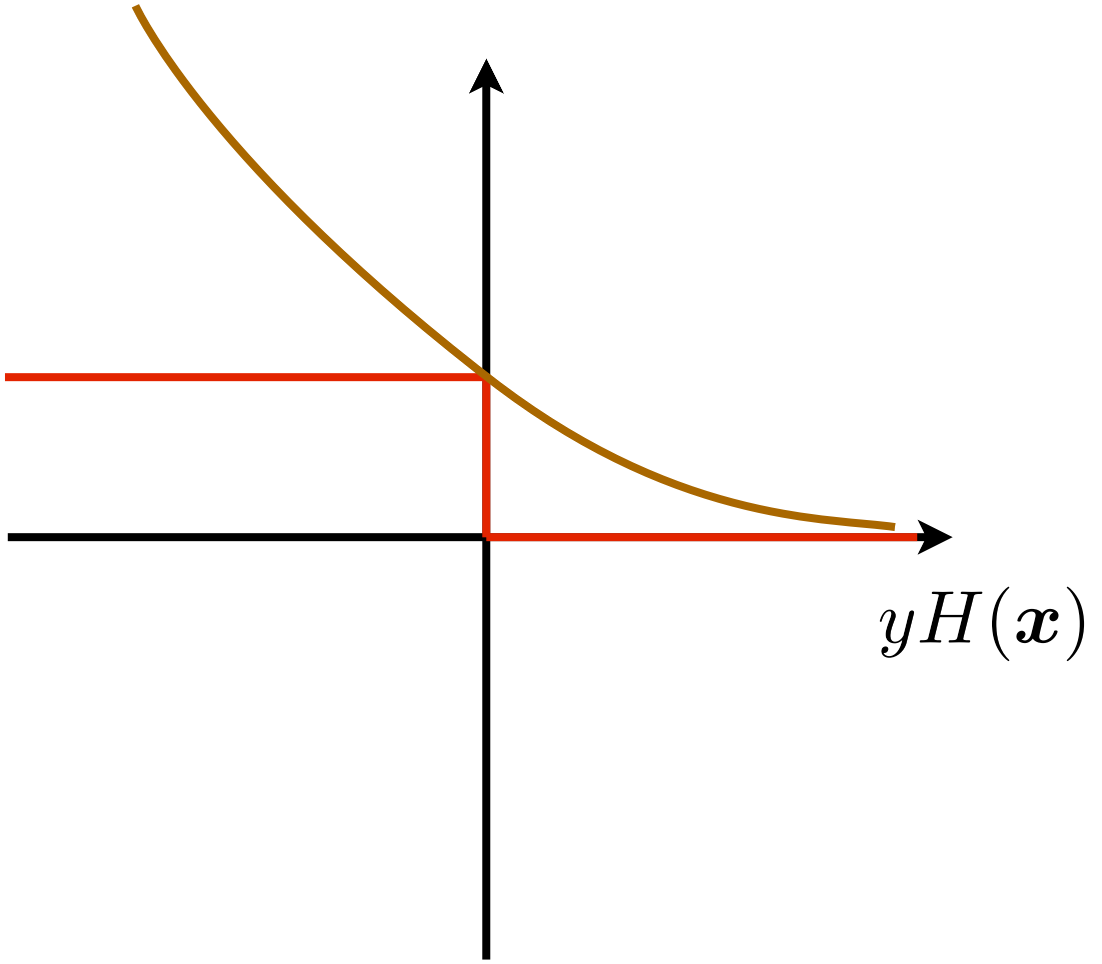
Gradient boosting (for classification)

0-1 loss

$$\min I(yH(\boldsymbol{x}) \leq 0)$$

logistic regression

$$\min \log(1 + e^{-yH(\boldsymbol{x})})$$



Sequential ensemble methods

Gradient boosting (for classification)

0-1 loss

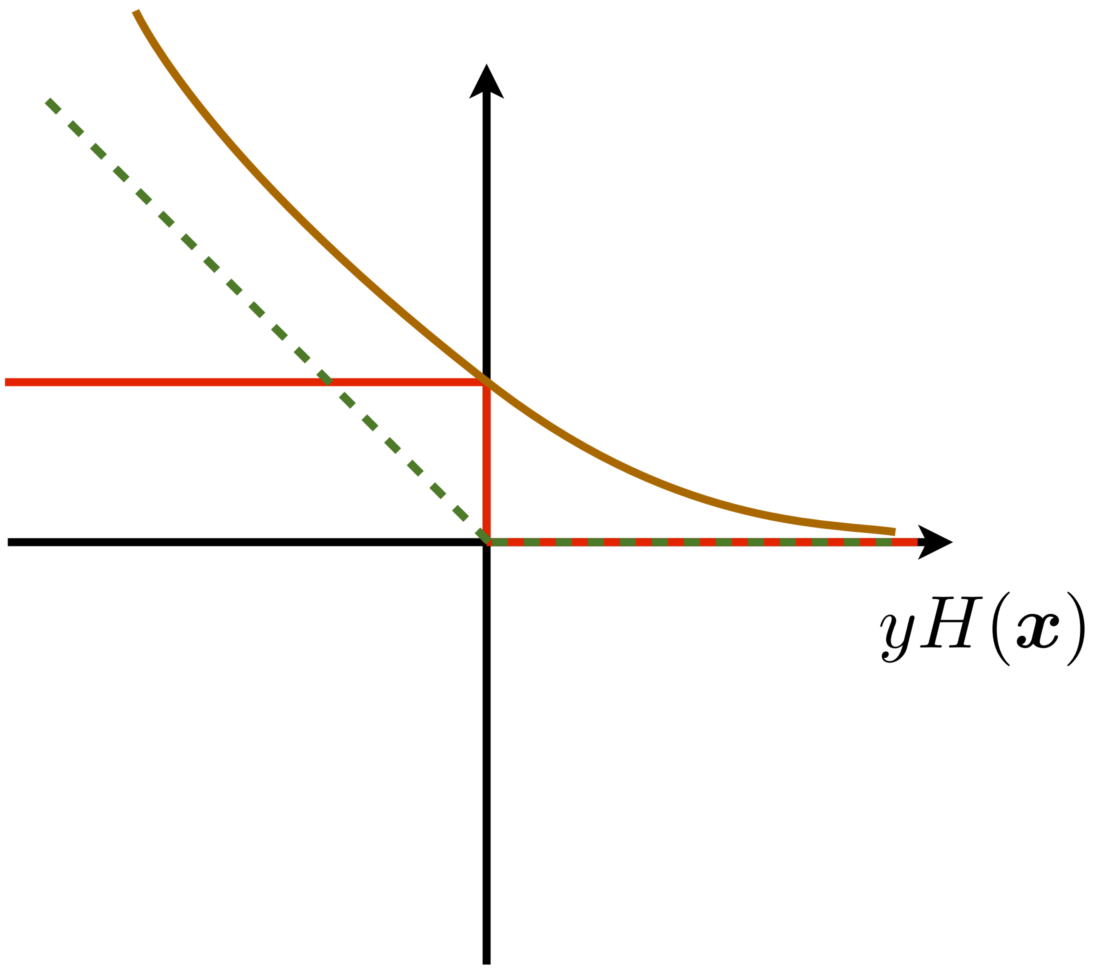
$\min I(yH(\boldsymbol{x}) \leq 0)$

logistic regression

$\min \log(1 + e^{-yH(\boldsymbol{x})})$

perceptron

$\min \max\{-yH(\boldsymbol{x}), 0\}$



Sequential ensemble methods

Gradient boosting (for classification)

0-1 loss

$$\min I(yH(\boldsymbol{x}) \leq 0)$$

logistic regression

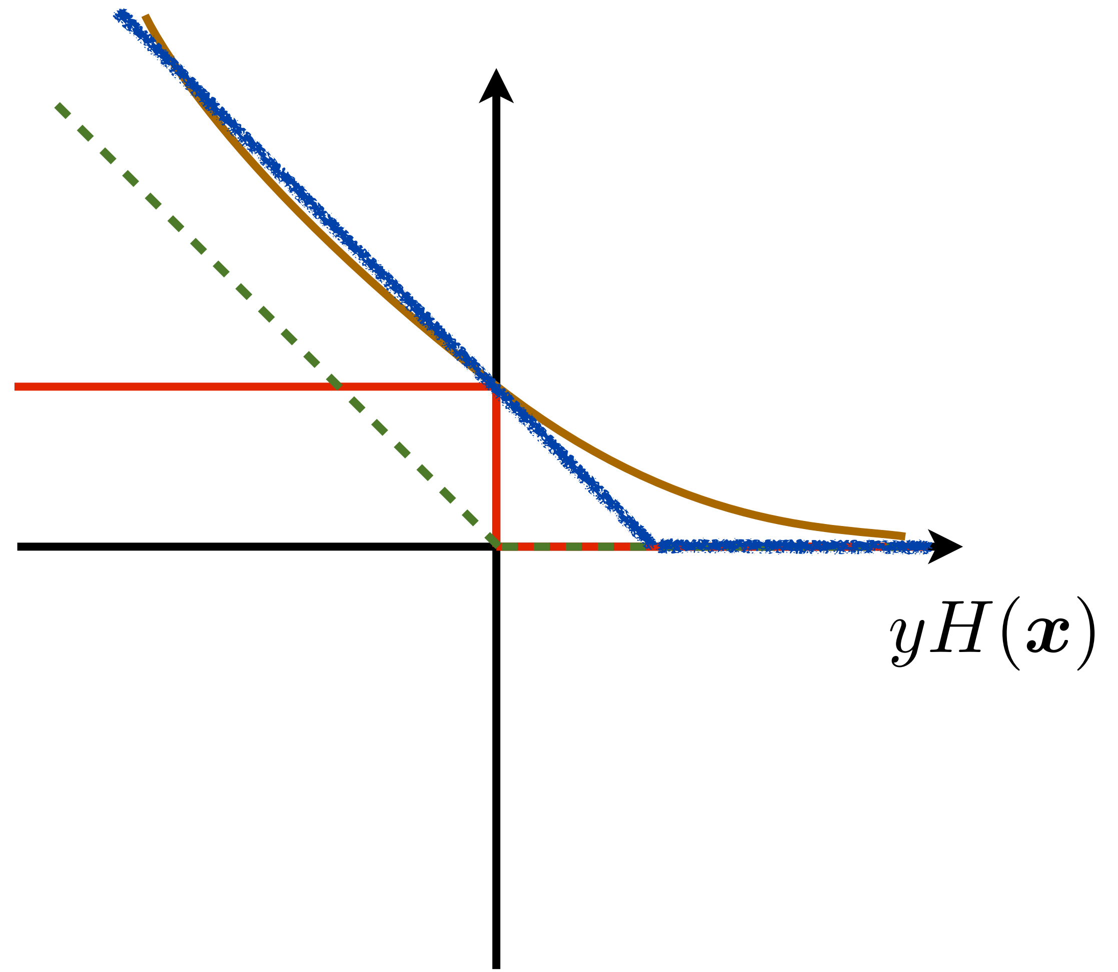
$$\min \log(1 + e^{-yH(\boldsymbol{x})})$$

perceptron

$$\min \max\{-yH(\boldsymbol{x}), 0\}$$

hinge loss

$$\min \max\{1 - yH(\boldsymbol{x}), 0\}$$



Sequential ensemble methods

Gradient boosting (for classification)

0-1 loss

$\min I(yH(\boldsymbol{x}) \leq 0)$

logistic regression

$\min \log(1 + e^{-yH(\boldsymbol{x})})$

perceptron

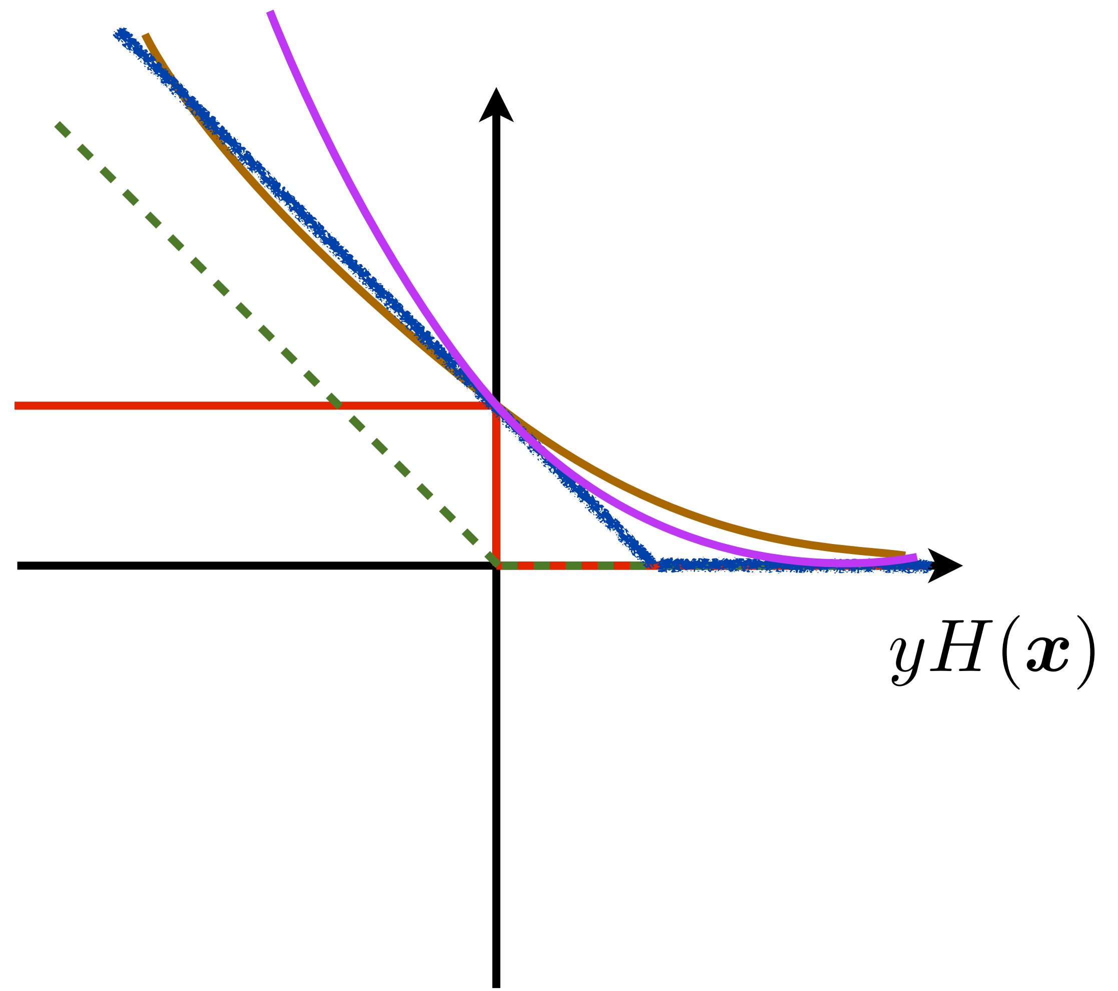
$\min \max\{-yH(\boldsymbol{x}), 0\}$

hinge loss

$\min \max\{1 - yH(\boldsymbol{x}), 0\}$

exponential loss

$\min e^{-yH(\boldsymbol{x})}$



Sequential ensemble methods

exponential loss

$$\min e^{-y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b)}$$

use (approximate) Newton's method to sequentially optimize exponential loss

$\Downarrow$

AdaBoost

Sequential ensemble methods

exponential loss

$$\min e^{-y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b)}$$

use (approximate) Newton's method to sequentially optimize exponential loss

L. Valiant
Turing Award 2010

is weak learnable class equals strong learnable class?

AdaBoost

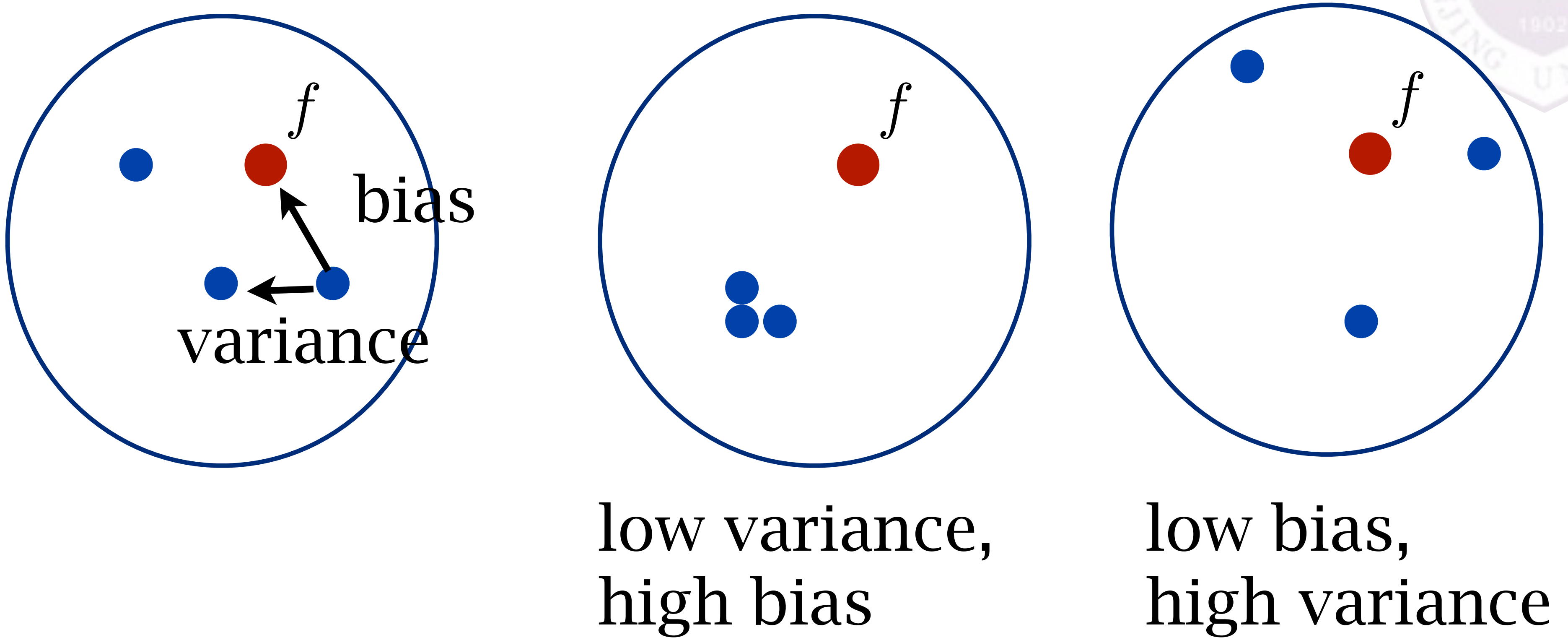
(Gödel Prize 2003)

yes! The proof is the boosting algorithm

R. Schapire

AdaBoost is the first practical boosting algorithm

Bias-variance analysis



parallel ensemble: reduce variance

sequential ensemble: reduce bias and variance

Applications

KDDCup: data mining competition organized by ACM SIGKDD

KDDCup 2009: to estimate the churn, appetency and up-selling probability of customers.

An Ensemble of Three Classifiers for KDD Cup 2009: Expanded Linear Model, Heterogeneous Boosting, and Selective Naïve Bayes

Hung-Yi Lo, Kai-Wei Chang, Shang-Tse Chen, Tsung-Hsien Chiang, Chun-Sung Ferng, Cho-Jui Hsieh, Yi-Kuang Ko, Tsung-Ting Kuo, Hung-Che Lai, Ken-Yi Lin, Chia-Hsuan Wang, Hsiang-Fu Yu, Chih-Jen Lin, Hsuan-Tien Lin, Shou-de Lin {D96023, B92084, B95100, B93009, B95108, B92085, B93038, D97944007, R97028, R97117, B94B02009, B93107, CJLIN, HTLIN, SDLIN}@CSIE.NTU.EDU.TW
Department of Computer Science and Information Engineering, National Taiwan University
Taipei 106, Taiwan

KDDCup 2010: to predict student performance on mathematical problems from logs of student interaction with Intelligent Tutoring Systems.

JMLR: Workshop and Conference Proceedings 1: 1-16 KDD Cup 2010

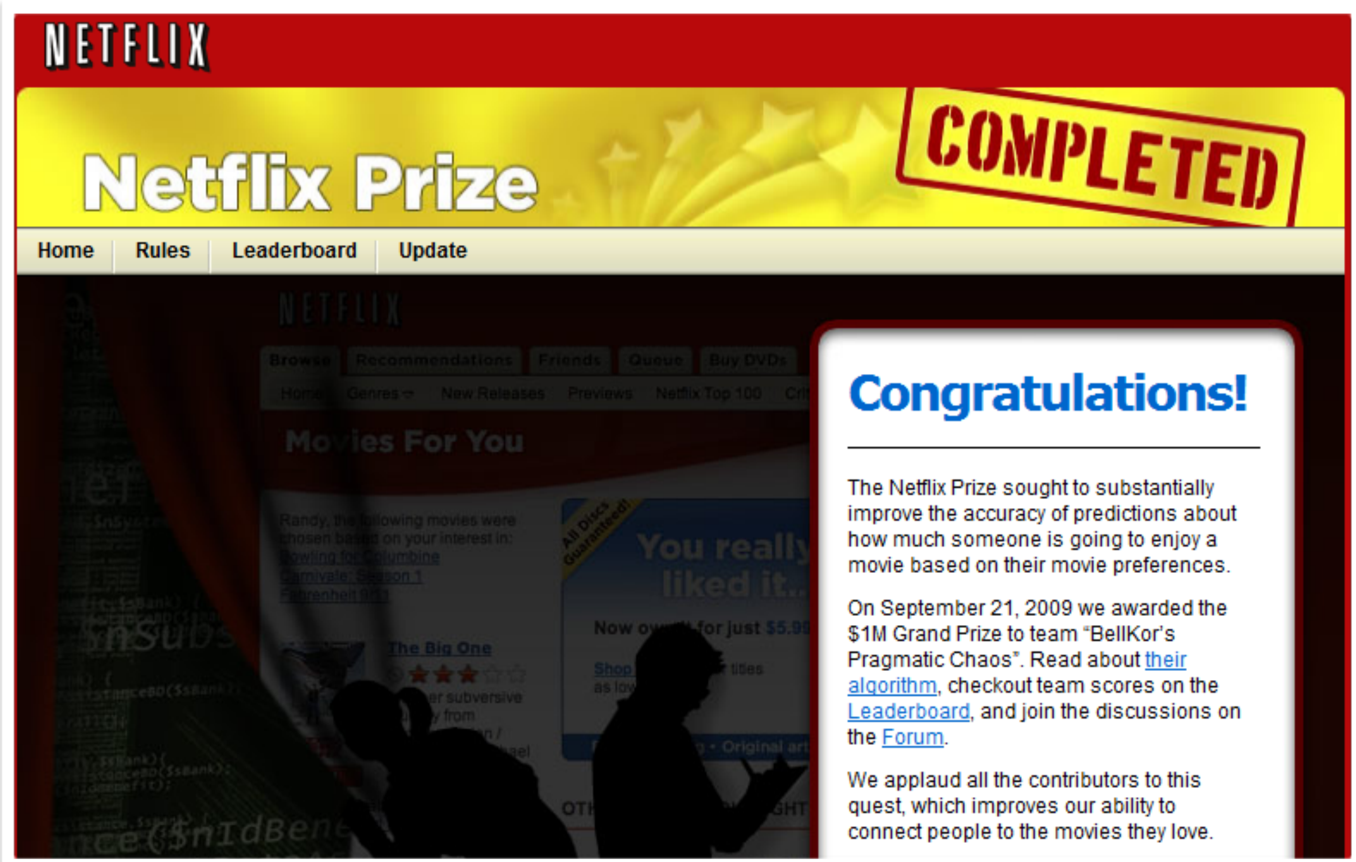
Feature Engineering and Classifier Ensemble for KDD Cup 2010

Hsiang-Fu Yu, Hung-Yi Lo, Hsun-Ping Hsieh, Jing-Kai Lou, Todd G. McKenzie, Jung-Wei Chou, Po-Han Chung, Chia-Hua Ho, Chun-Fu Chang, Yin-Hsuan Wei, Jui-Yu Weng, En-Syu Yan, Che-Wei Chang, Tsung-Ting Kuo, Yi-Chen Lo, Po Tzu Chang, Chieh Po, Chien-Yuan Wang, Yi-Hung Huang, Chen-Wei Hung, Yu-Xun Ruan, Yu-Shi Lin, Shou-de Lin, Hsuan-Tien Lin, Chih-Jen Lin
Department of Computer Science and Information Engineering, National Taiwan University
Taipei 106, Taiwan

KDDCup 2011, KDDCup 2012, and foreseeably, 2013, 2014 ...

Applications

Netflix Price: if one participating team improves Netflix's own movie recommendation algorithm by 10% accuracy, they would win the grand prize of $1,000,000.



习题

什么样的集成学习(ensemble learning)方法可能获得好的预测性能?

并行集成学习方法(parallel ensemble)为何可以并行进行训练?

作为0-1损失函数(0-1 loss)的近似，logistic regression loss、perception loss、hinge loss、exponential loss各有什么优缺点?