机器学习

主讲教师：刘峤

第6章 Emsemble Methods

Bagging and Boosting

Ensemble learning

- In statistics and machine learning
  - ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone.
- The term ensemble is usually reserved for methods that generate multiple hypotheses using the same base learner.
- The broader term of multiple classifier systems also covers hybridization of hypotheses that are not induced by the same base learner.

Ensemble theory

- **An ensemble is itself a supervised learning algorithm**
  - The trained ensemble represents a single hypothesis.
  - It is not necessarily contained within the hypothesis space of the models from which it is built.
  - **Empirically**, ensembles tend to yield better results when there is a significant diversity among the models.
  - Many ensemble methods, therefore, **seek to promote diversity** among the models they combine.
  - Although perhaps non-intuitive, more random algorithms (like random decision trees) can be used to produce a stronger ensemble than very deliberate algorithms (like entropy-reducing decision trees)

Multiple classifier systems

- Multiple classifier systems:
  - approaches to combine several machine learning techniques into one predictive model in order to
    - → decrease the variance (bagging)
    - → decrease the bias (boosting) or
    - → improving the predictive force (stacking)
- Why multiple classifier systems ?
  - The main causes of error in learning are due to noise, bias and variance. meta-algorithms helps to minimize these factors. improve the stability and the accuracy of Machine Learning algorithms.

Bias/Variance Decomposition

- **Squared loss of model on test case i:**

$$[\mathrm{Learner}(x_i, \mathcal{D}) - \mathrm{Truth}(x_i)]^2$$

- **Expected prediction error:**

$$E\left\{[\mathrm{Learner}(x_i, \mathcal{D}) - \mathrm{Truth}(x_i)]^2\right\}$$

$$= \mathrm{Noise}^2 + \mathrm{Bias}^2 + \mathrm{Variance}$$

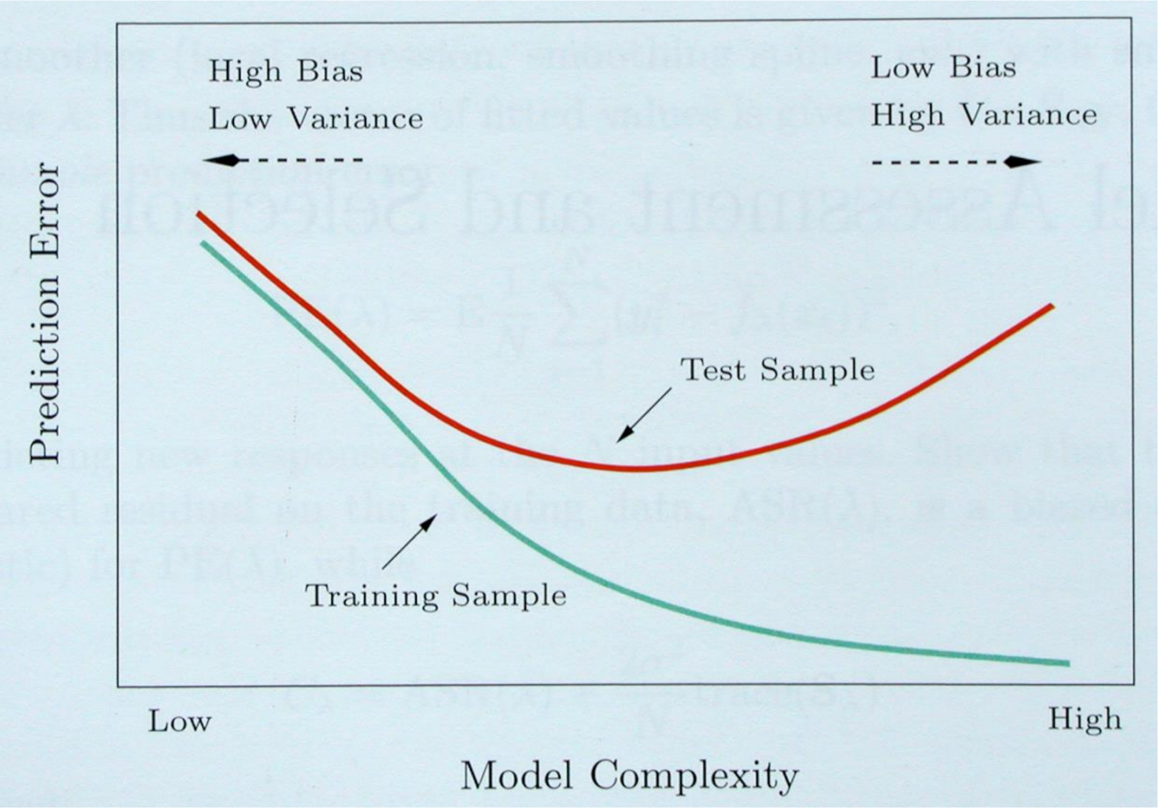
- $\mathrm{Noise}^2 =$ **lower bound on performance**
- $\mathrm{Bias}^2 =$ **(expected error due to model mismatch)^2**
- $\mathrm{Variance} =$ **variation due to train sample and randomization**

Sources of "variance" in Supervised Learning

- noise in targets or input attributes
- bias (model mismatch)
- training sample
- randomness in learning algorithm
  - Eg. neural net weight initialization
- randomized subsetting of train set
  - Eg. cross validation, train and early stopping set

Bias/Variance Tradeoff

- $\text{Bias}^2 + \text{Variance}$ **is what counts for prediction**
- **Tradeoff:** $\text{Bias}^2 \ \ vs. \ \ \text{Variance}$



Hastie, Tibshirani, Friedman "Elements of Statistical Learning" 2001

Blending and Bagging

Aggregation Models

- **aggregation: combine hypotheses for better performance**
- 例如：若我们有H个学习器可用于股票价格涨跌的预测
  - 策略1：选择性能表现最好的学习器
  - 策略2：让H个学习器进行无差别投票
  - 策略3：投票时给不同的学习器不同的权重
  - 策略4：有条件地combine各学习器的预测结果
    - 若 H_i 满足某些特定条件，则赋予其较多的投票权
- 这几种实际工作中常用的策略有何关联？对其进行形式化

Aggregation with Math Notations

- T个学习器: $g_1(x), \cdots, g_T(x)$
  - 策略1：选择性能表现最好的学习器

$$G(\mathbf{x}) = g_{t^*}(\mathbf{x}) \ \text{ with } t^* = \underset{t \in \{1,2,\ldots,T\}}{argmin} E_{val}(g_t(\mathbf{x}'))$$

  - 策略2：让T个学习器进行无差别投票

$$G(\mathbf{x}) = sign\left(\sum_{t=1}^{T} 1 \cdot g_t(\mathbf{x})\right)$$

  - 策略3：投票时给不同的学习器不同的权重

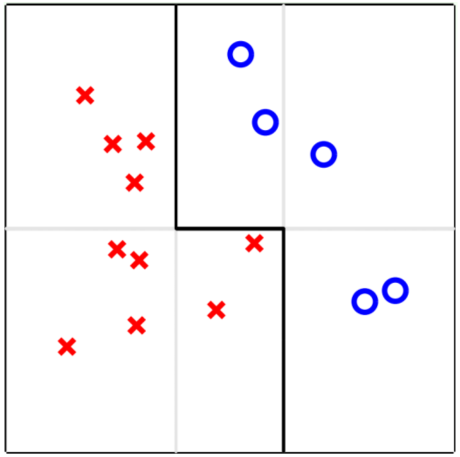
$$G(\mathbf{x}) = sign\left(\sum_{t=1}^{T} \alpha_t \cdot g_t(\mathbf{x})\right) \ \text{ with } \alpha_t \geq 0$$

  - 策略4：有条件地combine各学习器的预测结果

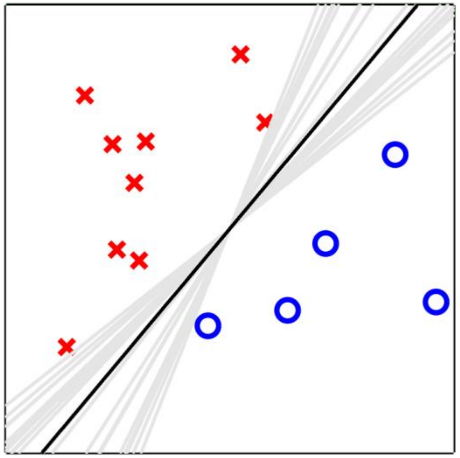
$$G(\mathbf{x}) = sign\left(\sum_{t=1}^{N} q_t(\mathbf{x}) \cdot g_t(\mathbf{x})\right) \ \text{ with } q_t(\mathbf{x}) \geq 0$$

Why Might Aggregation Work?

- aggregation: can we do better with many (possibly weaker) hypotheses?



- mix different weak hypotheses uniformly
  - G(x) will be stronger
  - feature transform?



- mix different random-PLA hypo. Uniformly
  - G(x) will be moderate
  - Regularization?

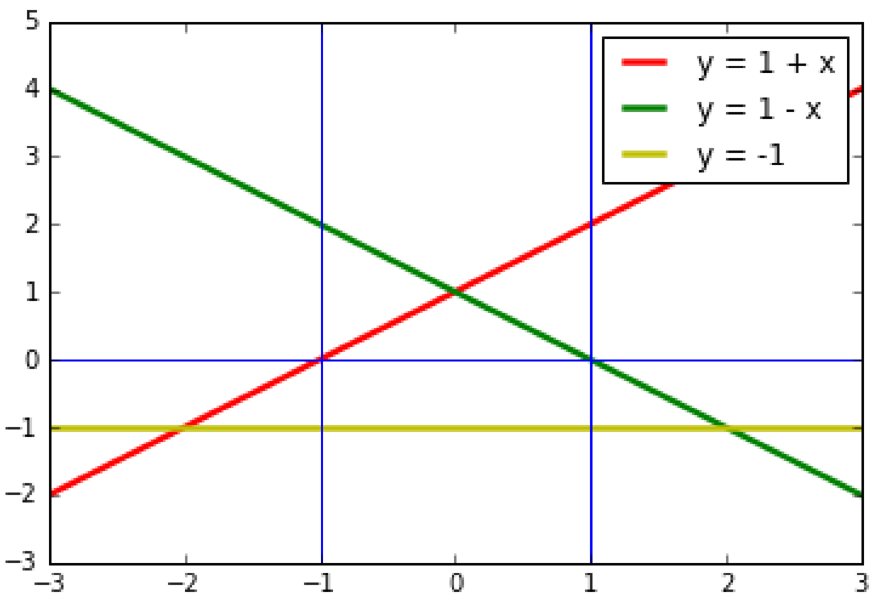
proper aggregation → better performance

Quize

Consider three decision stump hypotheses from $\mathbb{R}$ to $\{-1, +1\}$: $g_1(x) = \text{sign}(1 - x)$, $g_2(x) = \text{sign}(1 + x)$, $g_3(x) = -1$. When mixing the three hypotheses uniformly, what is the resulting $G(x)$?

1. $2\,[\![|x| \le 1]\!] - 1$
2. $2\,[\![|x| \ge 1]\!] - 1$
3. $2\,[\![x \le -1]\!] - 1$
4. $2\,[\![x \ge +1]\!] - 1$

Reference Answer: 1



The region that gets two positive votes from g1 and g2 is |x| <= 1, and thus G(x) is positive within the region only. We see that the three decision stumps gt can be aggregated to form a more sophisticated hypothesis G .

(1) Uniform Blending

Uniform Blending for Classification/Regression

- **Uniform Blending: known $g_t(x)$, each with 1 ballot**

$$G(\mathbf{x}) = sign\left(\sum_{t=1}^{T} 1 \cdot g_t(\mathbf{x})\right)$$

- **Uniform Blending for Regression**

$$G(\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T} g_t(\mathbf{x})$$

- **Diverse hypotheses:**
  - **Empirically, ensembles tend to yield better results when there is a significant diversity among the models**
  - **Uniform blending can be better than any single hypothesis**

Theoretical Analysis of Uniform Blending

- **Uniform Blending for Regression** $G(\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T} g_t(\mathbf{x})$

$$
\begin{aligned}
\operatorname{avg}\{(g_t(\mathbf{x}) - f(\mathbf{x}))^2\} &= \operatorname{avg}\left(g_t^2 - 2g_t f + f^2\right) \\
&= \operatorname{avg}\left(g_t^2\right) - 2Gf + f^2 \\
&= \operatorname{avg}(g_t^2) - G^2 + (G - f)^2 \\
&= \operatorname{avg}(g_t^2) - 2G^2 + G^2 + (G - f)^2 \\
&= \operatorname{avg}(g_t^2 - 2g_t G + G^2) + (G - f)^2 \\
&= \operatorname{avg}\{(g_t - G)^2\} + (G - f)^2
\end{aligned}
$$

$$
\operatorname{avg}\{E_{out}(g_t)\} = \operatorname{avg}\{\mathbf{E}(g_t - G)^2\} + E_{out}(G) \geq E_{out}(G)
$$

Some Special g_t

- **consider a virtual iterative process that for** $t = 1, 2, \ldots, T$
  1. **request size-*N* data** $\mathcal{D}_t$ **from** P^N **(i.i.d.)**
  2. **obtain** g_t **by** $\mathcal{A}(\mathcal{D}_t)$

$$\bar{g} = \lim_{T\to\infty} G = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} g_t = \mathop{\mathbf{E}}_{\mathcal{D}} \{\mathcal{A}(\mathcal{D})\}$$

$$\operatorname{avg}\{E_{out}(g_t)\} = \operatorname{avg}\{\mathbf{E}(g_t - \bar{g})^2\} + E_{out}(\bar{g})$$

- expected performance of A =
  - expected deviation to consensus (**variance**) +
  - performance of consensus (**bias**)

uniform blending: reduces variance for more stable performance

Quize

Consider applying uniform blending : $G(\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T} g_t(\mathbf{x})$

on linear regression hypotheses : $g_t(\mathbf{x}) = \mathbf{w}_t \cdot \mathbf{x}$. Which of

the following property best describes the resulting G(**x**)?

1. a constant function of **x**
2. a linear function of **x**
3. a quadratic function of **x**
4. none of the other choices

$$G(\mathbf{x}) = \left(\frac{1}{T}\sum_{t=1}^{T} \mathbf{w}_t\right) \cdot \mathbf{x}$$

Reference Answer: 2

(2) Linear Blending

Linear Blending

- **Linear Blending: known $g_t(x)$, each to be given α_t ballot**

$$G(\mathbf{x}) = sign\left(\sum_{t=1}^{T} \alpha_t \cdot g_t(\mathbf{x})\right) \quad \text{with } \alpha_t \geq 0$$

- **Computing good α_t :** $\min\limits_{\alpha_t \geq 0} E_{in}(\alpha)$

- **Linear blending for regression**

$$\min_{\alpha_t \geq 0} \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \sum_{t=1}^{T} \alpha_t g_t(\mathbf{x}_i)\right)^2$$

- **Linear Regression + transformation**

$$\min_{\mathbf{w}_i} \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{k} \mathbf{w}_j \Phi_j(\mathbf{x}_i)\right)^2$$

- **Linear blending = LinModel + hypotheses as transform + constraints**

Constraint on α_t

- linear blending = LinModel + hypotheses as transform + constraints

$$\min_{\alpha_t \geq 0} \frac{1}{N} \sum_{i=1}^{N} err\left(y_i, \sum_{t=1}^{T} \alpha_t g_t(\mathbf{x}_i)\right)$$

- Linear blending for binary classification

$$\text{if } \alpha_t < 0 \;\Rightarrow\; \alpha_t g_t(\mathbf{x}) = |\alpha_t|(-g_t(\mathbf{x}))$$

  - negative α_t for g_t $\equiv$ positive $|\alpha_t|$ for $-g_t$

- in practice, often the constraints are ignorable
  - Linear blending = LinModel + hypotheses as transform

Linear blending in action

- **blending practically done with**

$$E_{val} \text{ (instead of } E_{in}) + g_t \text{ from minimum } E_{train}$$

- **Given :** $g_1, g_2, \ldots, g_T \text{ from } \mathcal{D}_{train}$
    - **transform** (x_i, y_i) **in** D_{val} **to (** $\mathbf{z}_i = \Phi(\mathbf{x}_i), y_i$ **)**
    - **where :** $\Phi(\mathbf{x}) = (g_1(\mathbf{x}), \ldots, g_T(\mathbf{x}))$
- **Any Blending (Stacking) : g could be any model**
    - **powerful, achieves conditional blending**
    - **but danger of overfitting, as always :-(**

Quize

Consider three decision stump hypotheses from $\mathbb{R}$ to $\{-1, +1\}$:
$g_1(x) = \text{sign}(1 - x)$, $g_2(x) = \text{sign}(1 + x)$, $g_3(x) = -1$. When $x = 0$, what is the resulting $\mathbf{\Phi}(x) = (g_1(x), g_2(x), g_3(x))$ used in the returned hypothesis of linear/any blending?

1. $(+1, +1, +1)$
2. $(+1, +1, -1)$
3. $(+1, -1, -1)$
4. $(-1, -1, -1)$

Reference Answer: 2

（3） Bagging

Brief Summary

- blending: aggregate after getting g_t
- learning g_t for uniform aggregation: **diversity** important
  - diversity by different **models**: $g_1 \in \mathcal{H}_1, g_2 \in \mathcal{H}_2, \ldots, g_T \in \mathcal{H}_T$
  - diversity by different **parameters**: gradient descent with

$$\eta = 0.001, 0.01, \ldots, 10$$

  - diversity by **algorithmic randomness**:
    - Eg. random PLA with different random seeds
  - diversity by **data randomness**:
    - within-cross-validation hypotheses g_v

Revisit of Bias-Variance

$$\text{avg}\{E_{out}(g_t)\} = \text{avg}\{\mathbf{E}(g_t - \bar{g})^2\} + E_{out}(\bar{g})$$

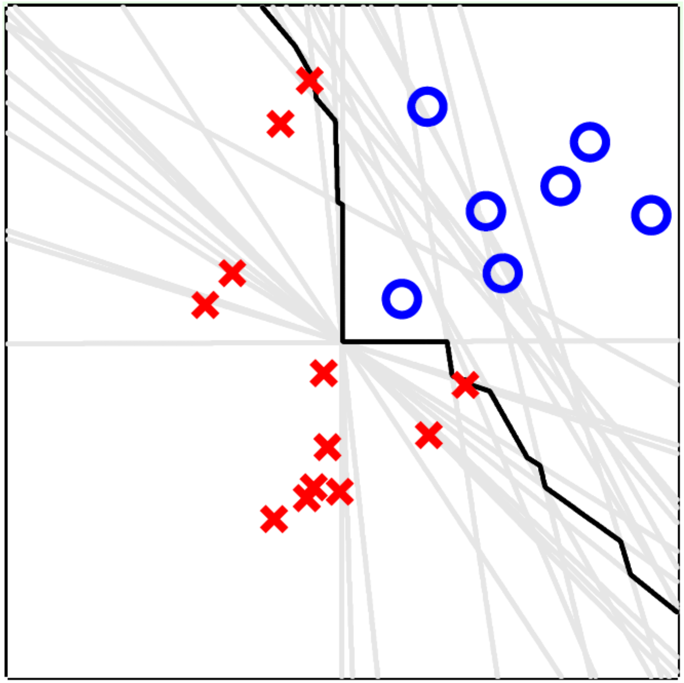
- expected performance of A = **variance + bias**
  - **variance :** expected deviation to consensus
  - **bias :** performance of consensus

- consensus more stable than direct A(D)
  – but comes from many more ***D_t*** than the ***D*** on hand
- want: approximate g by
  – finite (large) T
  – approximate $g_t = \mathcal{A}(\mathcal{D}_t)$ from $\mathcal{D}_t \sim P^N$ using only D
- bootstrapping: re-samples from D to simulate $\mathcal{D}_t$

Bootstrap Aggregation

- bootstrap sample $\mathcal{D}_t$
  - re-sample ***N*** examples from ***D*** uniformly with replacement
  - can also use arbitrary ***N'*** instead of original ***N***
- **Bootstrap aggregating (Bagging)**
  - consider a iterative process that for $t = 1, 2, \ldots, T$
    ① request size-N' data $\mathcal{D}_t$ from bootstrapping
    ② obtain g_t by $\mathcal{A}(\mathcal{D}_t)$: $\mathrm{G} = \mathrm{Uniform}(\{\ \mathrm{g}_t\})$
- Bagging : a simple meta algorithm on top of base algorithm A

Bagging Pocket in Action



$$T_{Pocket} = 1000$$

$$T_{Bagging} = 25$$

- very diverse g_t from bagging
- proper non-linear boundary after aggregating binary classifiers
- bagging works reasonably well
  - if base algorithm sensitive to data randomness

Pocket Algorithm

initialize pocket weights $\hat{\mathbf{w}}$

For $t = 0, 1, \cdots$

1. find a (random) mistake of $\mathbf{w}_t$ called $(\mathbf{x}_{n(t)}, y_{n(t)})$
2. (try to) correct the mistake by

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + y_{n(t)}\mathbf{x}_{n(t)}$$

3. if $\mathbf{w}_{t+1}$ makes fewer mistakes than $\hat{\mathbf{w}}$, replace $\hat{\mathbf{w}}$ by $\mathbf{w}_{t+1}$

...until enough iterations

return $\hat{\mathbf{w}}$ (called $\mathbf{w}_{\text{POCKET}}$) as g

- **PLA算法最大的缺点是假设数据线性可分**
  - **若数据线性可分，则算法可证收敛**
  - **若数据不是线性可分，则算法无法收敛**

$$\mathbf{w}_g \leftarrow \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{n=1}^{N} ind\left\{y_i \neq sign(\mathbf{w}^T \mathbf{x}_i)\right\}$$ **NP-hard**

Quize

When using bootstrapping to re-sample N examples $\tilde{\mathcal{D}}_t$ from a data set $\mathcal{D}$ with N examples, what is the probability of getting $\tilde{\mathcal{D}}_t$ exactly the same as $\mathcal{D}$?

1. $0 \;/N^N = 0$
2. $1 \;/N^N$
3. $N! \;/N^N$
4. $N^N/N^N = 1$

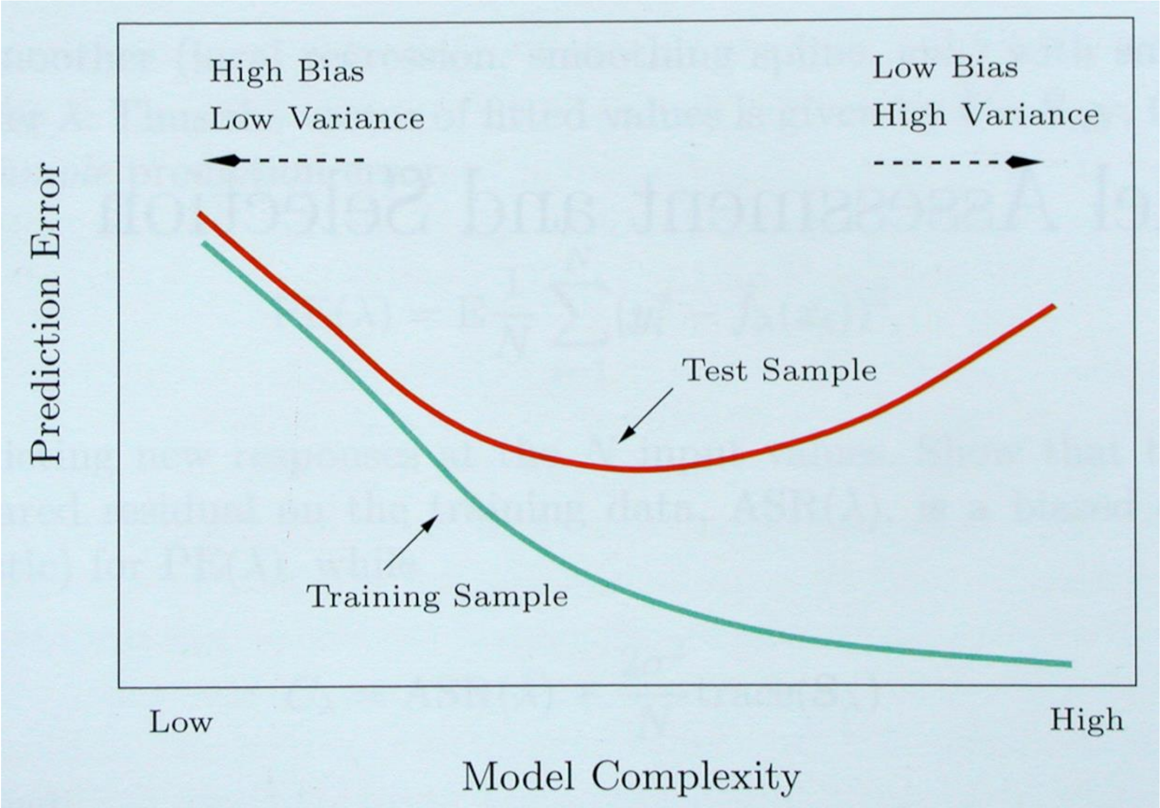
Reference Answer: 3

- **Consider re-sampling in an ordered manner for N steps**
  - **Then there are (N^N) possible outcomes Dt**
  - **each with equal probability**
  - **(N!) of the outcomes are permutations of the original D**

Roadmap

Bias/Variance Tradeoff

- $\text{Bias}^2 + \text{Variance}$ **is what counts for prediction**
- **Tradeoff:** $\text{Bias}^2 \;\; vs. \;\; \text{Variance}$



Hastie, Tibshirani, Friedman "Elements of Statistical Learning" 2001

Reduce Variance Without Increasing Bias

- Averaging reduces variance:

$$Var(\bar{\mathbf{x}}) = \frac{Var(\mathbf{x})}{N}$$

- Average models to reduce model variance
  - One problem: only one train set
  - where do multiple models come from?
- Bagging: Bootstrap Aggregation
  - Leo Breiman (1994) (1928 - 2005)
  - Bootstrap Sample:
    - draw sample of size |D| with replacement from D

Bagging: Bootstrap Aggregation

- **Best case:**

$$\mathrm{Var}(\mathrm{Bagging}(L(\mathbf{x},,\mathcal{D}))) = \frac{\mathrm{Var}(L(\mathbf{x},,\mathcal{D}))}{N}$$

- **In practice:**
  - **models are correlated, so reduction is smaller than 1/N**
  - **variance of models trained on fewer training cases usually somewhat larger**
  - **stable learning methods have low variance to begin with, so bagging may not help much**

Can Bagging Hurt?

- Each base classifier is trained on less data
  - Only about **63.2%** of the data points are in any bootstrap sample
  - Javed A. Aslam, et al. On Estimating the Size and Confidence of a Statistical Audit. Proceedings of the Electronic Voting Technology Workshop. Boston, MA, August 6, 2007.
- However the final model has seen all the data
  - On average a point will be in **>50%** of the bootstrap samples

Reduce $Bias^2$ and Decrease Variance?

- Bagging reduces variance by averaging
- Bagging has little effect on bias
- Can we average and reduce bias?

Yes : Boosting

Boosting

- Freund & Schapire
- Weak Learner: performance on any train set is slightly better than chance prediction
  - PAC : theory for *weak learners* in late 80’s （Valiant）
- intended to answer a theoretical question
  - not as a practical way to improve learning
- tested in mid 90's using not-so-weak learners
- works anyway!

Boosting

- Weight all training samples equally
- Train model on train set
- Compute error of model on train set
- Increase weights on train cases model gets wrong
- Train new model on re-weighted train set
- Re-compute errors on weighted train set
- Increase weights again on cases model gets wrong
- Repeat until tired (100+ iterations)
- Final model: weighted prediction of each model

Boosting vs. Bagging

- Bagging doesn't work so well with stable models.
  - Boosting might still help.
- Boosting might hurt performance on noisy datasets.
  - Bagging doesn't have this problem
- In practice bagging almost always helps.
- On average, boosting helps more than bagging, but it is also more common for boosting to hurt performance.
  - The weights grow exponentially.
- Bagging is easier to parallelize.

Bagging and Boosting

- Probably Approximately Correct (PAC, Kearns & Valiant)
- Ensemble : learners are trained using same learning techniques.
  - Bagging : bootstrap aggregating （random forest）
    - bootstrap : pull up by your own bootstraps
  - Boosting (adaboost)
    - Can a set of weak learners create a single strong learner?
- Hybrid: learners are trained using different learning techniques.
  - Stacking : combining multiple models （ meta learners ）

电子科技大学

UESTC

1956