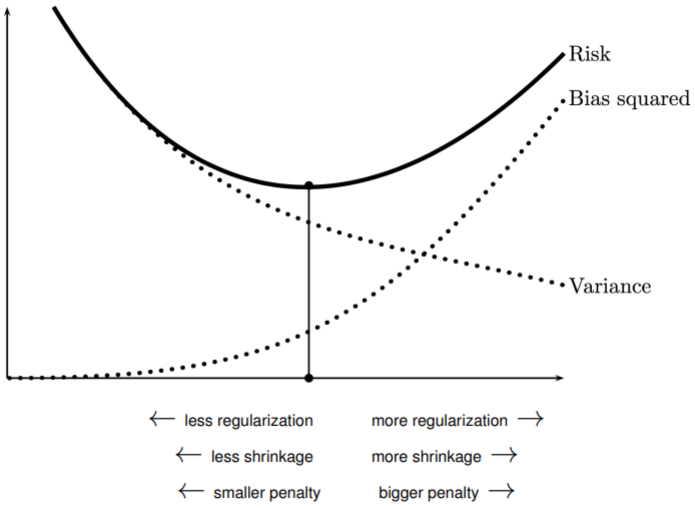
## 1 Bias-variance tradeoff

$$
\begin{aligned}
\mathrm{E}(\theta-\widehat{\theta})^2 &= \mathrm{E}(\theta-\mathbb{E}\widehat{\theta}+\mathbb{E}\widehat{\theta}-\widehat{\theta})^2\\
&= \mathbb{E}(\theta-\mathbb{E}\widehat{\theta})^2-2\mathrm{E}\{(\theta-\mathbb{E}\widehat{\theta})(\widehat{\theta}-\mathbb{E}\widehat{\theta})\}+\mathbb{E}(\widehat{\theta}-\mathbb{E}\widehat{\theta})^2\\
&= \mathbb{E}(\theta-\mathbb{E}\widehat{\theta})^2-2(\theta-\mathbb{E}\widehat{\theta})\mathrm{E}(\widehat{\theta}-\mathbb{E}\widehat{\theta})+\mathbb{E}(\widehat{\theta}-\mathbb{E}\widehat{\theta})^2\\
&= \mathbb{E}(\theta-\mathbb{E}\widehat{\theta})^2+\mathbb{E}(\widehat{\theta}-\mathbb{E}\widehat{\theta})^2\\
&= \mathrm{Bias}(\widehat{\theta})^2+\mathrm{Variance}(\widehat{\theta})
\end{aligned}
$$



## 2 Cross Validation

| | Iteration | | | | | |
|---|---|---|---|---|---|---|
| Obs | 1 | 2 | 3 | 4 | . . . | n |
| 1 | valid | train | train | train | . . . | train |
| 2 | train | valid | train | train | . . . | train |
| 3 | train | train | valid | train | . . . | train |
| 4 | train | train | train | valid | . . . | train |
| . . . | . . . | . . . | . . . | . . . | . . . | . . . |
| *n* | train | train | . . . | . . . | . . . | valid |
| MSE | *MSE*$_1$ | *MSE*$_2$ | *MSE*$_3$ | *MSE*$_4$ | . . . | *MSE*$_n$ |

LOOCV test error: $CV_{(n)} = \frac{1}{n}\sum_i MSE_i$

| | Iteration | | | | | | |
|---|---|---|---|---|---|---|---|
| Obs | 1 | 2 | 3 | 4 | . . . | *k* | |
| 1 | valid | train | train | train | . . . | train | |
| 2 | valid | train | train | train | . . . | train | fold 1 |
| 3 | valid | train | train | train | . . . | train | |
| 4 | train | valid | train | train | . . . | train | |
| . . . | . . . | . . . | . . . | . . . | . . . | . . . | |
| *n* − 2 | train | train | . . . | . . . | . . . | valid | |
| *n* − 1 | train | train | . . . | . . . | . . . | valid | fold *k* |
| *n* | train | train | . . . | . . . | . . . | valid | |
| MSE | *MSE*$_1$ | *MSE*$_2$ | *MSE*$_3$ | *MSE*$_4$ | . . . | *MSE*$_k$ | |

K-Fold: $CV_{(k)} = \sum_b \frac{n_b}{n} MSE_b$ where $n_b$ is the total observations in the b-th fold, and n is the total observations in the entire dataset. Suppose the fitted values can be written $\hat{Y} = HY$. The leave-one-out-cross-validation error is

$$R_{LOOCV} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \widehat{Y}_{(-i)}\right)^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{Y_i - \widehat{Y}_i}{1-H_{ii}}\right)^2$$

where $H_{ii}$ is the i-th diagonal entry. This is the case for least squares and regularized multiple regression:
$H = X\left(X^TX\right)^{-1}X^T$ or $H = X\left(X^TX + \lambda I\right)^{-1}X^T$

## 3 Tree

Tree Build: (1) Cycle through predictors $X_k$ for $k = 1, \dots, p$. For each $X_k$:
- (Quantitative $X_k$) Consider cutpoints s (unique values of $X_k$) that divide up the region into two parts:

$$R_1(k,s) = \{i \mid X_{ik} < s\} \quad \text{and} \quad R_2(k,s) = \{i \mid X_{ik} \geq s\}$$

- Evaluate (for regression trees):

$$Q_k(s) = \sum_{i:i\in R_1(k,s)} (y_i - \bar{y}_{R_1})^2 + \sum_{i:i\in R_2(k,s)} (y_i - \bar{y}_{R_2})^2$$

Find the value of s that minimizes $Q_k(s)$. Call this $s_k$.
(2) Find the predictor $X_k$ with the minimum $Q_1(s_1), Q_2(s_2), \dots, Q_p(s_p)$. Make the first binary partition along predictor $X_k$ at cut point $s_k$.
Cost-complexity pruning:

$$C(T) = \sum_{m=1}^{|T|}\sum_{i\in R_m}\left(y_i - \widehat{y}_{R_m}\right)^2 + \alpha|T|$$

$\alpha = 0$ implies the full tree, Larger $\alpha$ implies higher penalty for complexity of model. With increasing $\alpha$:
(1) Grow a big tree on a training set. (2) Obtain a nested set of subtrees $T_L \subset \cdots \subset T_2 \subset T_1 \subset T$ corresponding to a sequence of $\alpha$ values. (3) Use K -fold cross-validation to identify the subtree that does best.
Classification Tree Impurity measures: Define node proportion of class k: $\widehat{p}_{mk} = \frac{1}{N_m}\sum_{x_i\in R_m} I(y_i = k)$ and $k(m) = \arg\max_k \widehat{p}_{mk}$
- Misclassification error: $1 - \widehat{p}_{mk(m)}$
- Gini index: $\sum_{k=1}^{K}\widehat{p}_{mk}\left(1-\widehat{p}_{mk}\right)$
- Entropy: $-\sum_{k=1}^{K}\widehat{p}_{mk}\log\widehat{p}_{mk}$

Bagging
Regression trees: Create B bootstrap samples, grow tree (without pruning) using each $\widehat{f}^{*1}, \widehat{f}^{*2}, \dots \widehat{f}^{*B}$. For prediction at x , we take an average: $\widehat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B}\widehat{f}^{*b}(x)$
Classification trees: $\widehat{f}_{\text{bag}}(x)$ is decided by majority vote.
OOB Estimation: For each bagged tree, we can make predictions for the OOB observations. At the end, we can aggregate over all predictions for the i-th observation to arrive at a OOB prediction $\widehat{y}_i$. We can compute prediction error based on these OOB predictions $\widehat{y}_1, \dots, \widehat{y}_n$.
Random Forests:
(1) For b=1 to B:
(a) Draw a bootstrap sample $Z^*$ of size n from the training data
(b) Grow a random-forest tree $T_b$ to the bootstrapped data, recursively repeating following steps, until minimum node size reached: i. Select m variables at random from the p variables ii. Pick the best variable/split-point among the m iii. Split the node into two children nodes

(2) Output the ensemble fo trees $\{T_b\}_{b=1}^{B}$. To make a prediction at a new point x :

Regression: Average $\hat{f}_{rf}^{B}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$

Classification: Majority vote of the individual trees

## 4 PCA

Algorithm (1) Center the data: $x_i \mapsto x_i - \frac{1}{n}\sum_{j=1}^{n} x_j = x_i - \bar{x}$ (2) Compute the $d \times d$ sample covariance $S = \frac{1}{n}\sum_{i=1}^{n} x_i x_i^T$. Note that $\frac{1}{n}\sum_{i}(x_{ij} - \bar{x})^2$ is the sample variance of j th coordinate of data.(3) Find the first k eigenvectors of S ,$\phi_1, \ldots, \phi_k$ ∈R$^d$, $S\phi_j = \lambda_j \phi_j$(4) Project the data onto those k vectors: $x_i \mapsto \bar{x} + \left(\phi_1^T x_i\right)\phi_1 + \ldots + \left(\phi_k^T x_i\right)\phi_k$

PCA is an unsupervised method

- Finds directions of greatest variation in the data
- The directions are called the principal vectors; the weightings on the vectors are called the principal components
- The first few vectors may be interpretable
- Orthogonality makes interpretation difficult for the higher components
- Can be used for visualization or dimensionality reduction