Review of Algorithms and Implementation in R

Commonly-used Algorithms

Unsupervised

K-Means*

Hierarchical clustering*

PCA

Self Organizing Maps

Gaussian Mixture Models

Supervised

Support Vector Machines

Naïve Bayes

K-Nearest Neighbors*

Regularized Regression*

Decision Trees*

Neural Networks

Ensemble*

Bagging

- Random Forest

Boosting

- XGBoost

Stacking

- SuperLearner

*Reviewed in more detail in later slides

Critical thinking is not optional….

Credit: XKCD

Consider needs of research question….

FIGURE 2.7. *A representation of the tradeoff between flexibility and interpretability, using different statistical learning methods. In general, as the flexibility of a method increases, its interpretability decreases.*

What types of epidemiologic questions/tasks benefit from high flexibility?

What types require more interpretability?

What does interpretability mean in the context of machine learning?

Source: ISLR

Consider particulars of the data…

Are data highly correlated?

Do you anticipate non-linear effects?

Are you interested in interactions between features/exposures?

Clustering

Broad set of techniques for finding subgroups or clusters in a dataset-ISLR

Difference between clusters>>> difference within clusters

All observations are forced into a cluster

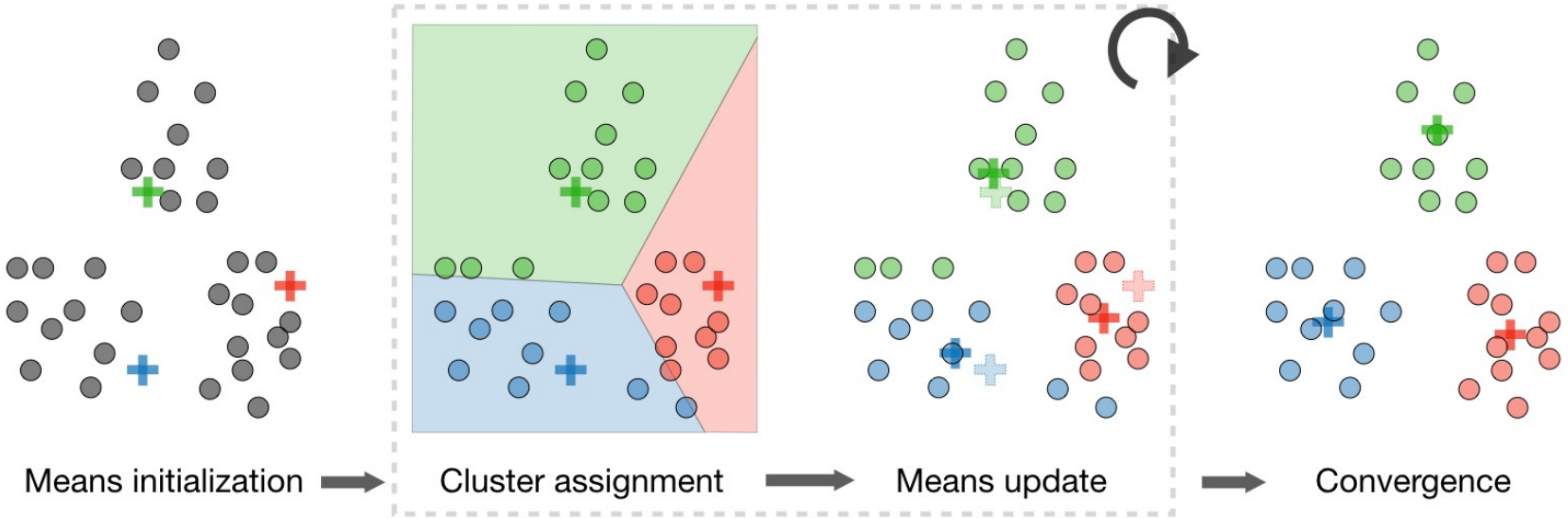
Two common algorithms: K-means and Hierarchical

Applications

Identify clusters of observations based on features: phenotypic subgroup identification

Identify clusters of features based on observations: identifying genetic expression patterns

K-means clustering: user must specify k



Source: CS-229 Course Material

Hierarchical Clustering

Builds nested clusters in a successive manner

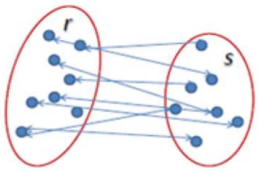
Agglomerative: each observation starts in its own cluster, and pairs of clusters are merged successively; better in discovering small clusters

Divisive: all observations start in the same cluster and splits are performed recursively; better in discovering large clusters

Merges and splits decided by cluster **dissimilarity.** Dissimilarity computed by **distance** and **linkage**.

Multiple distance metrics for calculating dissimilarity

- Euclidian
- Squared Euclidian
- Manhattan
- Maximum
- Mahalanobis

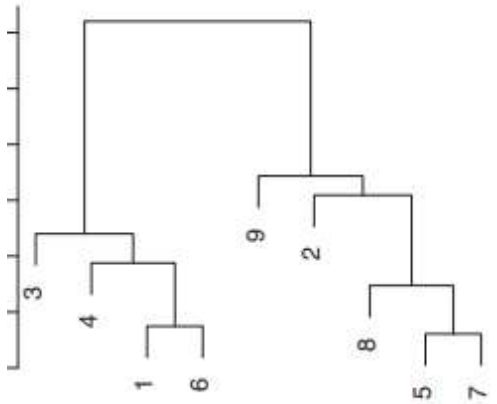


Multiple criterion for linkage

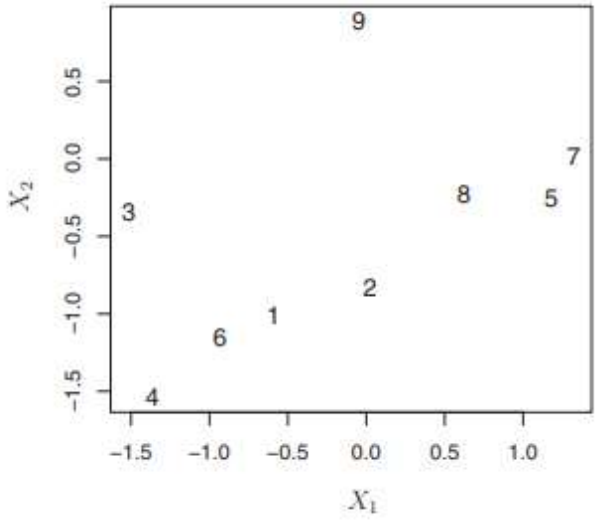
- Complete
- Single
- Average

Dendrogram: visualization of hierarchical clustering

Dendrogram after clustering



Raw Data



Source ISLR

Again, need to determine optimal number of clusters

Evaluating Number of Clusters

Goal: minimize the intra-cluster variation (i.e. homogeneous clusters)

Elbow: Plot the within-cluster sum of squares. Optimal clusters is identified at the bend in the plot

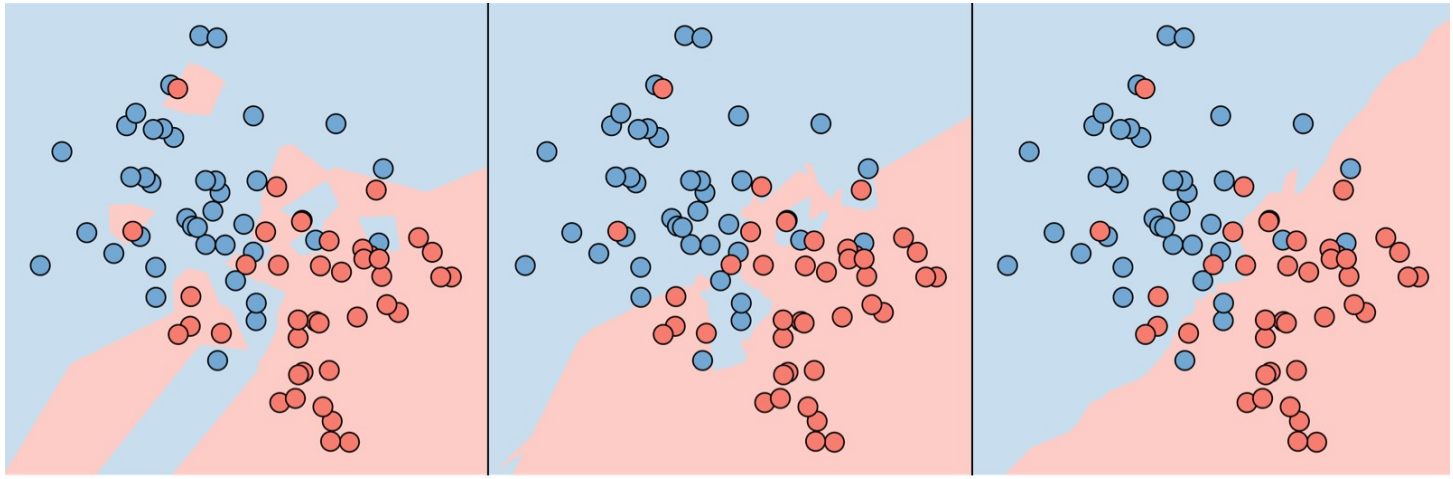
Silhouette: measure of the quality of the clustering, maximize s

$s = \frac{b-a}{\max(a,b)}$, where a is mean distance between a sample and other samples within the same class and b is the mean distance between a sample and other points in the next nearest cluster

Gap-statistic: compares total within cluster variation with their expected values under the null reference distribution. Requires boostrapping to generate reference distribution. Maximize the gap-statistic to identify optimal number of clusters.

K-nearest neighbors

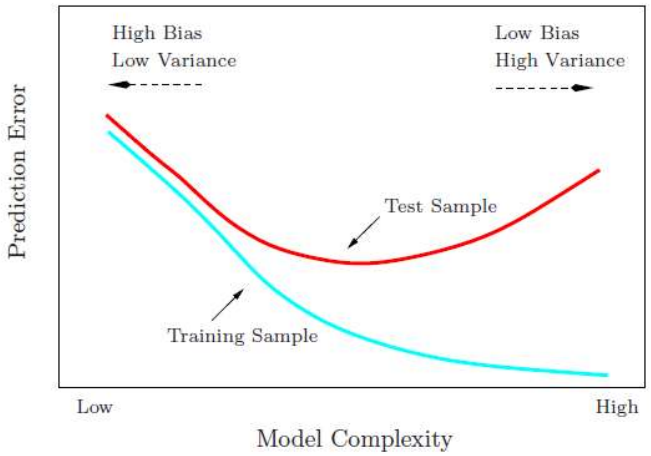
Non-parametric approach where the predicted value of an observation is determined by the nature of its k-neighbors from the training set.



Source: CS229 Course Material

Regularized Regression (shrinkage)

Addresses limitation of regression: Increase flexibility/complexity by including power terms, interactions, etc. decreases bias but can increase variance (overfitting)

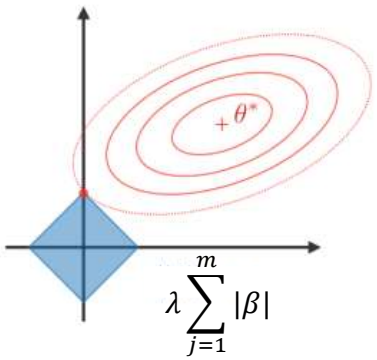
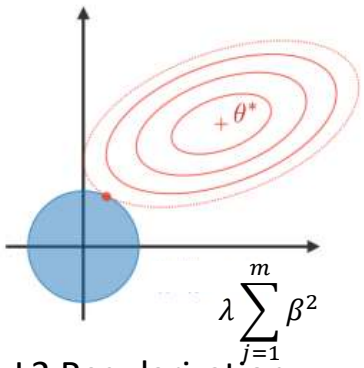
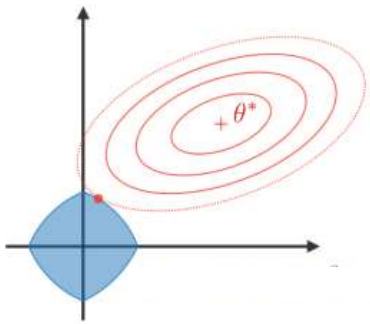


$$\sum_{i=1}^{N}\left(y_i - \theta_0 - \sum_{j=1}^{p}\theta_j \cdot x_{ij}\right)^2 + \lambda\|\theta\|$$

Penalty function

Residual sum of squares
Typical least squares loss function

Difference in the algorithms: penalty

LASSO: least absolute shrinkage operator	**Ridge**	**Elastic Net**
• Shrinks coefficients to 0 • Good for variable selection	Makes coefficients smaller	Tradeoff between variable selection and small coefficients
		
L1 Regularization	L2 Regularization	Combines L1 and L2 Regularization

Elastic Net:

$$\frac{\sum_{i=1}^{n}(y_i - x_i^J \hat{\beta})^2}{2n} + \lambda \left(\frac{1-\alpha}{2} \sum_{j=1}^{m} \hat{\beta}_j^2 + \alpha \sum_{j=1}^{m} |\hat{\beta}_j| \right)$$

Alpha is the hyperparameter that controls the loss function that differentiates LASSO, Ridge and Elastic Net

Varying Lambda (λ)

Controls the penalty function

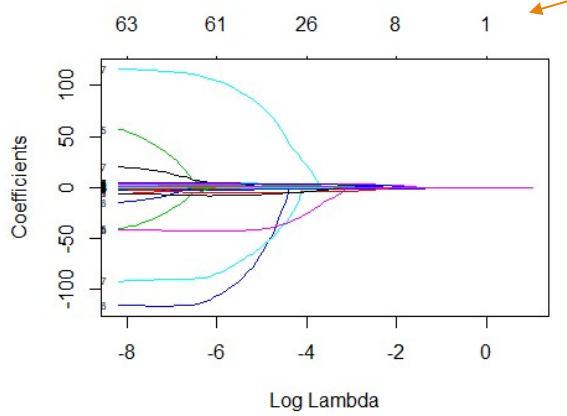
λ = 0: penalty=0 and produces standard regression coefficients

λ -> ∞: penalty=large and regression coefficients shrunk toward 0

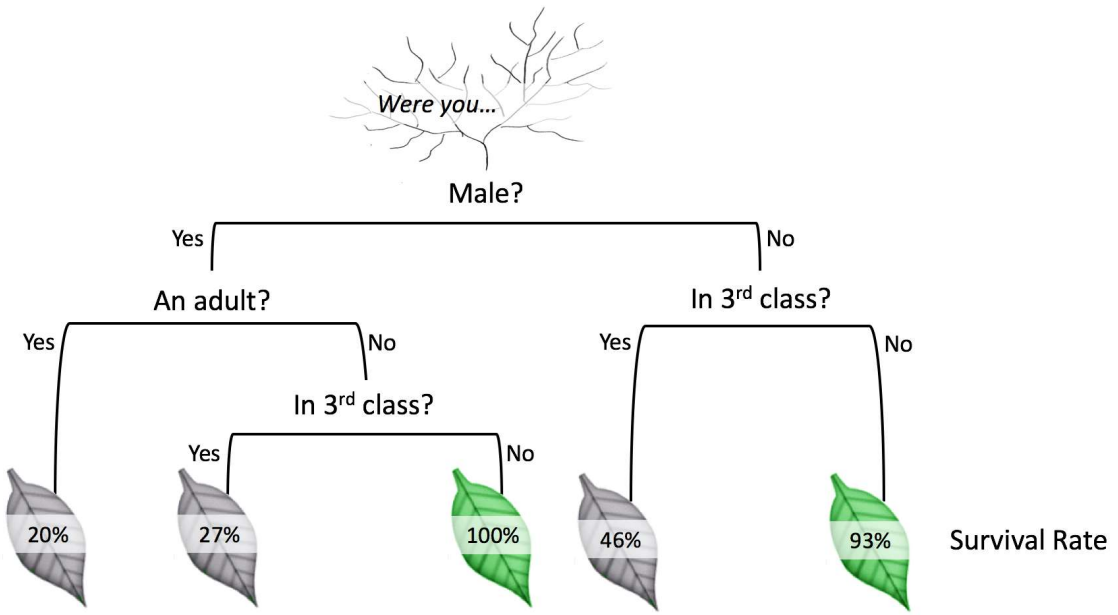
λ : chosen through **cross-validation**

Number of features

Plot of coefficients from LASSO when varying λ



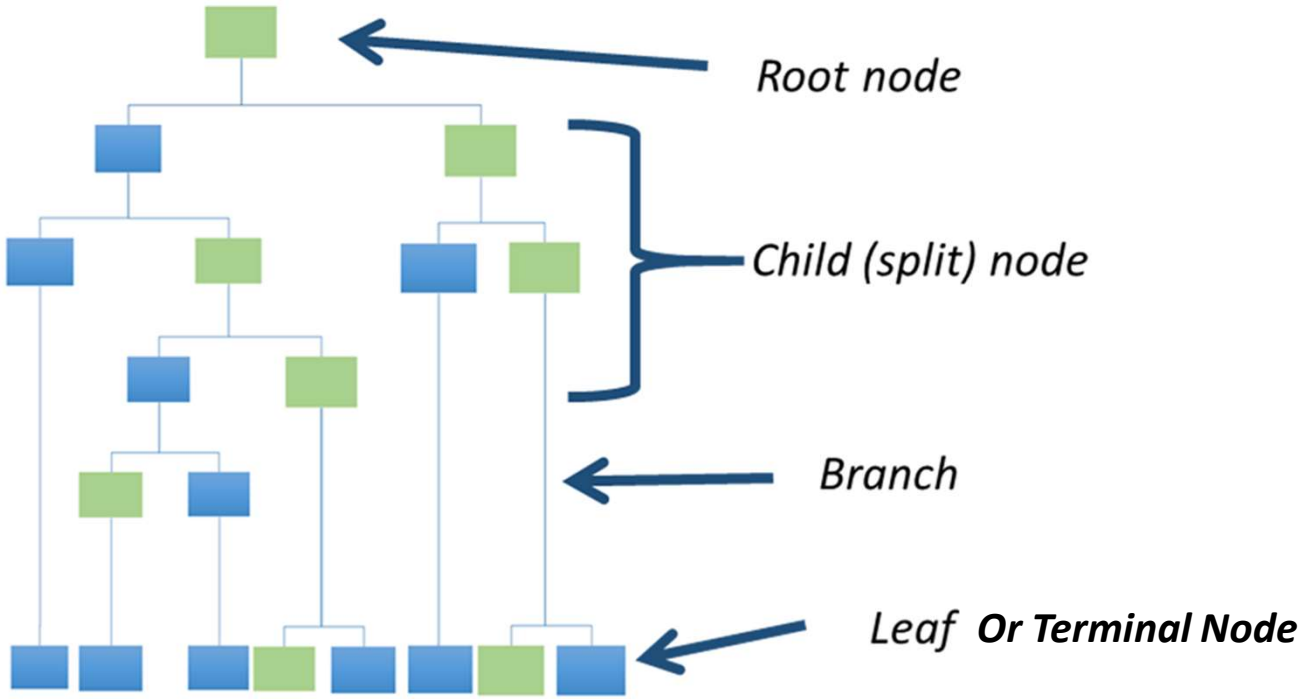
Intro to Tree-Based Methods: Titanic Example



Source: algobeans.com

Structure of a Tree

Trees generated through recursive partitioning



Key Terms

“Greedy” Algorithm: makes the optimal choice at each step

- Makes “best” first split without consideration of subsequent splits

Node purity: homogeneity of a node in relation to the labels of the observations contained

- Goal is often to maximize node purity to obtain optimal classification or prediction

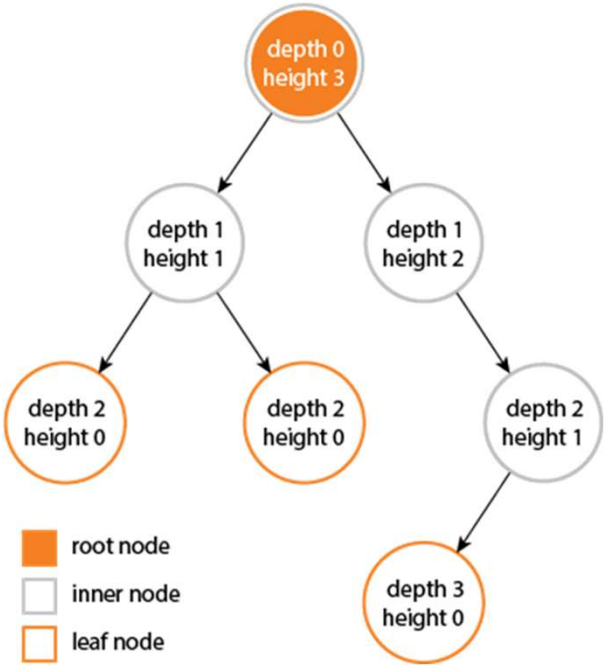
Measures of purity

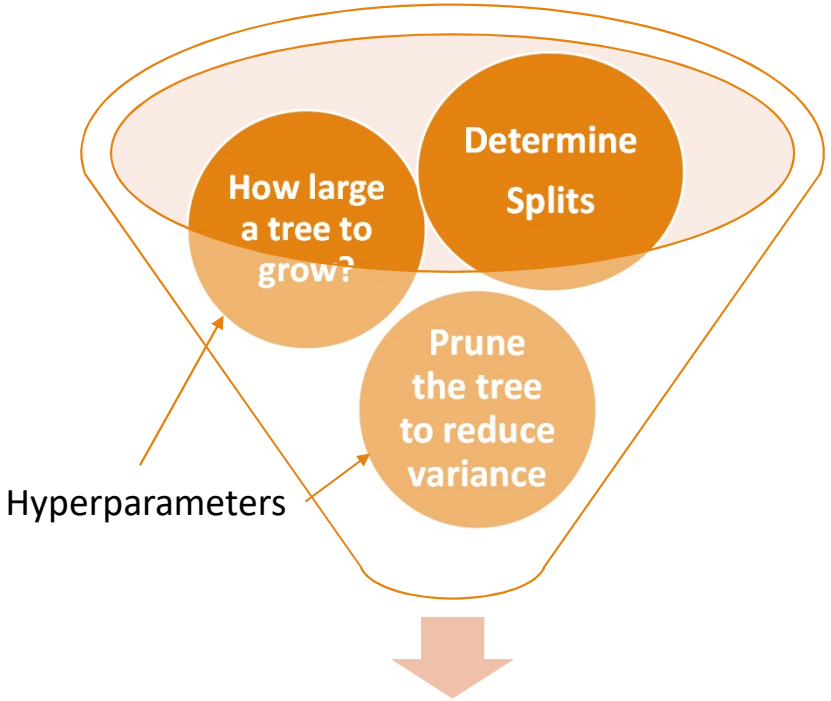
- Gini coefficient, entropy, variance, mean square error…. and others

Surrogate split: split using another feature that most closely resembles the consequences of the original split

- Often how tree-based methods handle missing data

Growing a Tree: Analytic Considerations



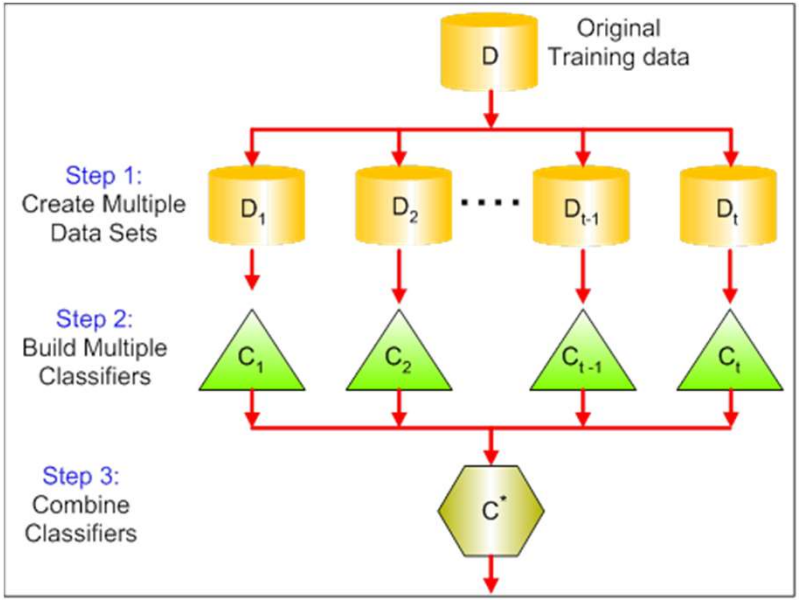


Constructing a tree-based model

Ensemble Methods

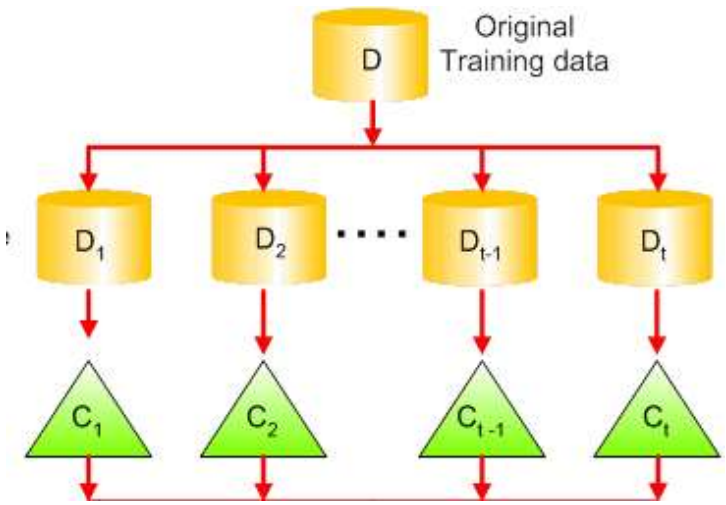
IMPROVE PREDICTION

Ensembles **combine** several base models in order to produce an optimal prediction



Simple Techniques to Combine: Classification

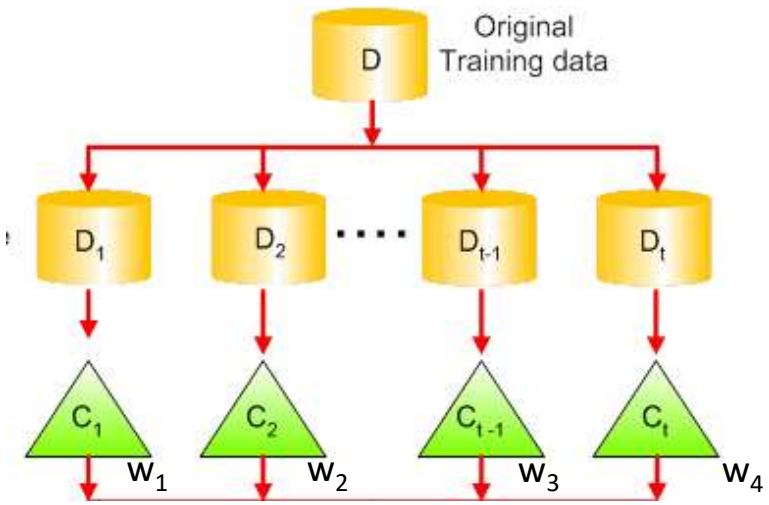
Majority Voting



C_1=1 C_2=1 C_3=0 C_4=1

C=1

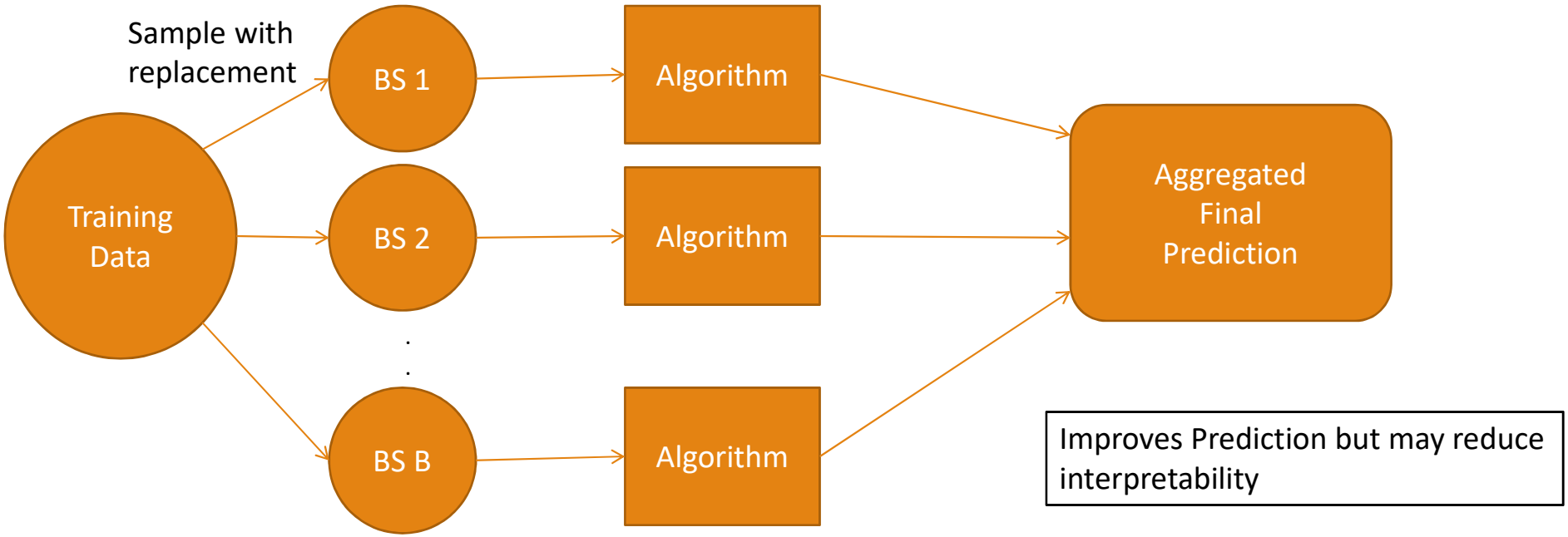
Weighted Voting



Final Prediction (C) a weighted combination with weights calculated based on predictive ability

Advanced Techniques: Bagging

Bagging: Bootstrap Aggregation; average results (or voting) across bootstrapped samples of data



Random Forest: Extension of Bagging

At each split, we choose a random sample of *m* predictors from the full set of *p* features

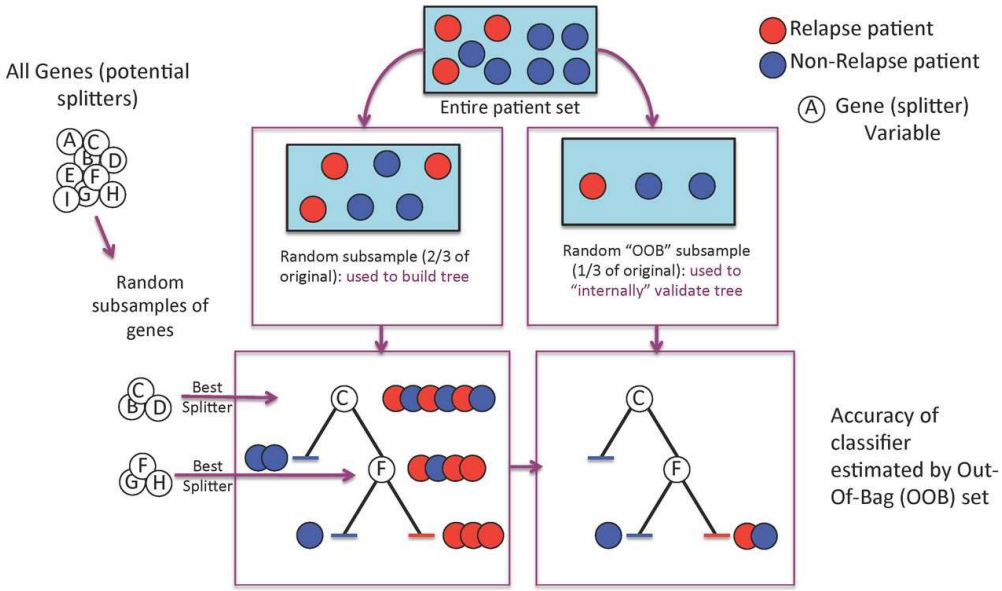
When m=p, then bagging and RF are equivalent

Randomly sampling from features de-correlates the trees

Provides opportunity for all features to contribute to prediction

Improves prediction overall, even though individual trees may be weaker

For prediction, want m to be small, typically $m=\sqrt{p}$



Source: BioInformatics Handbook

Variable Importance Factors

Measure of Individual Variable Contribution to the Overall Prediction

Accuracy-based Importance

Within OOB, record the prediction/classification error.

Permute the feature and recalculate prediction/classification

Difference in two errors are averaged over trees and normalized.

Node purity based Importance (Gini importance for classification)

Within OOB, total decrease in node impurities from splitting on variable averaged over all trees.

Advanced Techniques: Boosting

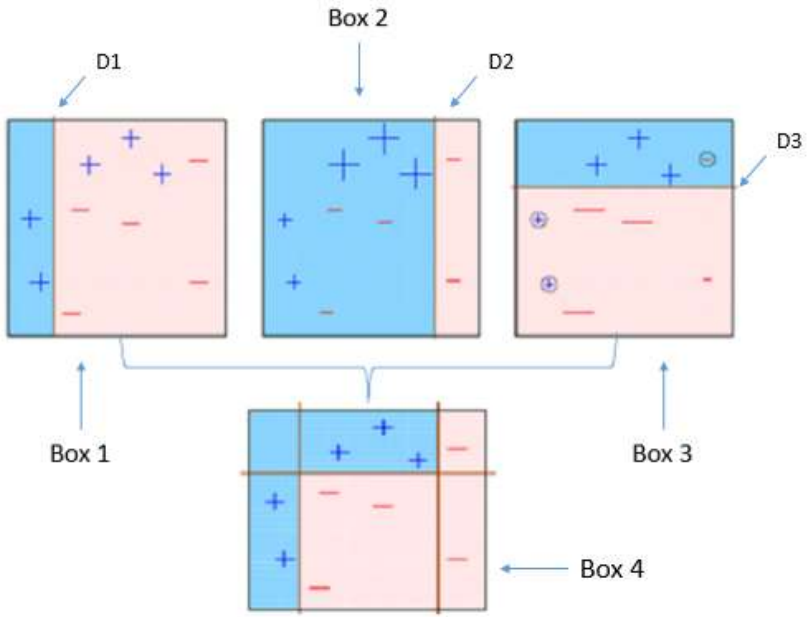
General technique that can be applied to diverse algorithms; popular with trees as base

Goal: Convert series of weak learners to a strong learner by learning algorithms sequentially

Example: Results from first algorithm provide information to second, etc.

Both provide an initial strong learner and then add to the initial learner with weaker learners

In adaptive boosting, information is weights of data points. If classified correctly, gets smaller weight so algorithm focuses on misclassified datapoints.



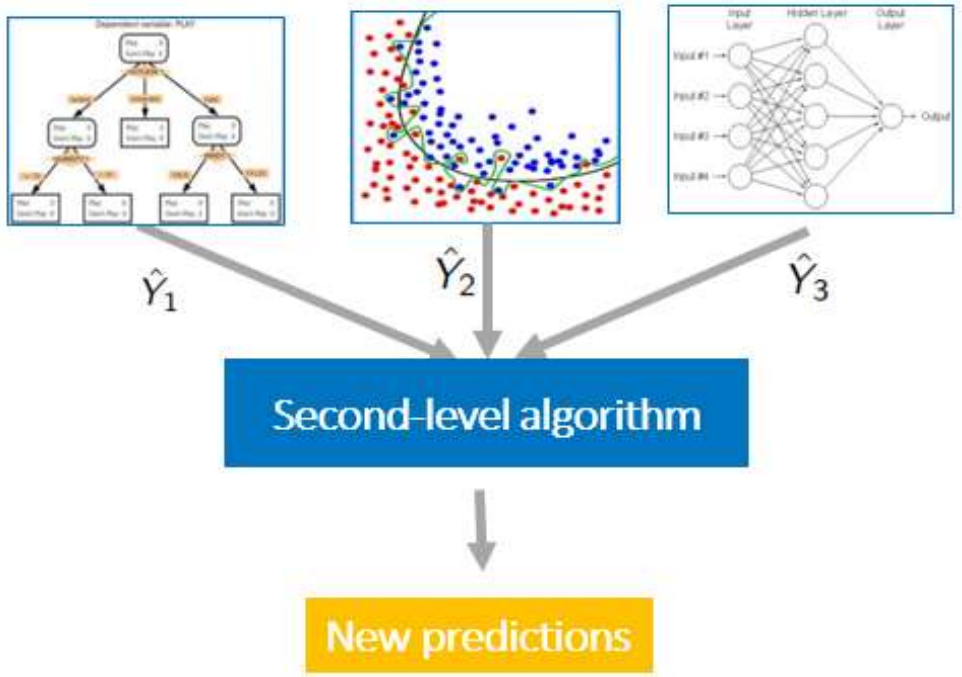
Source: Desarda Understanding AdaBoost

Advanced Techniques: Stacking

Rather than combining results of multiple learners, stacking uses predictions from learners as inputs to additional learners.

Optimal performance when diverse learners are used in the initial predictions

Common Example in Epidemiology: SuperLearner



Source: SAS Blogs

Overview of Super Learner

Data Partitioning

- Partition into K-folds
- Within each fold, partition into training and testing

Fit with library of algorithms

- Select library consisting of 'm' algorithms
- For each fold, fit each algorithm on the training set. Apply in test set. Calculate evaluation metric (L-2 squared error loss but equivalent to MSE)

Combine across folds

- Average evaluation metrics across all folds to obtain one measure for each algorithm
- Discrete SuperLearner: choose algorithm which minimizes loss

Combine across algorithms

- Regress actual outcome against predictions from each algorithm with certain constraints (depending upon type of data)
- Obtain weights for each algorithm by normalizing coefficient values

Obtain final predicted outcome

- Use weights in combination with predictions from each algorithm to generate predictions for newly observed data

Naimi and Balzer Eur J Epidemiol 2018; 33:459-464

Implementation in R