Lecture 7: Machine Learning and the microbiome

UNSUPERVISED LEARNING: CLUSTERING

How do we understand clustering?

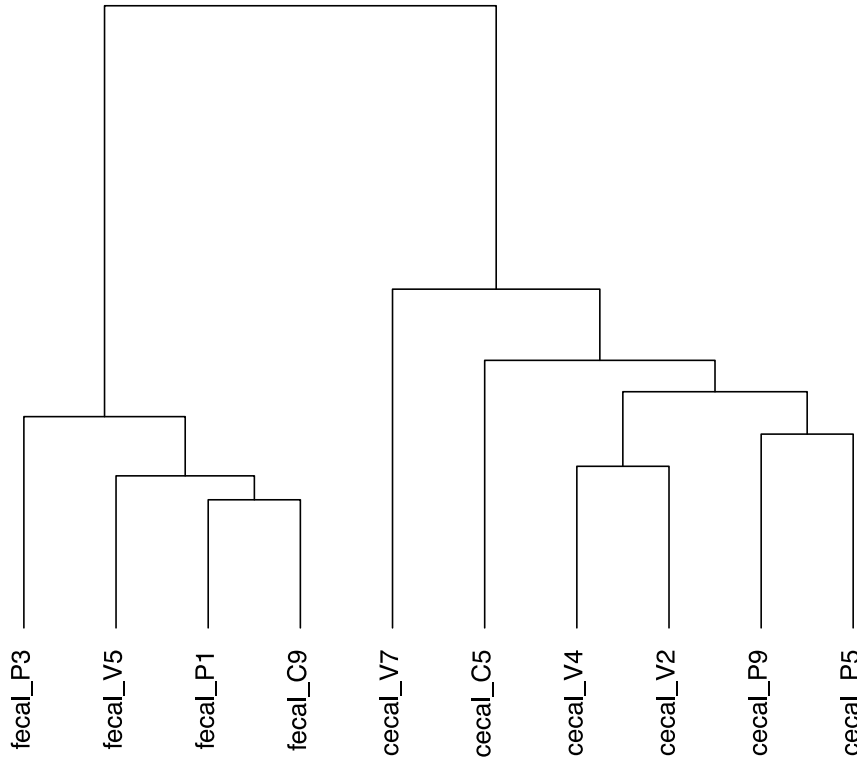
- What does it mean for the data to be clustered?
- What meaning do the clusters have?
- How do you know the data can be clustered?

Definition

- Clustering analysis – methodologies for describing proximity between objects
- Hierarchical clustering – a set of descriptive techniques for grouping objects by similarity
- Discrete clustering – a set of techniques for assessing membership of objects in one of several groups.

Hierarchical clustering

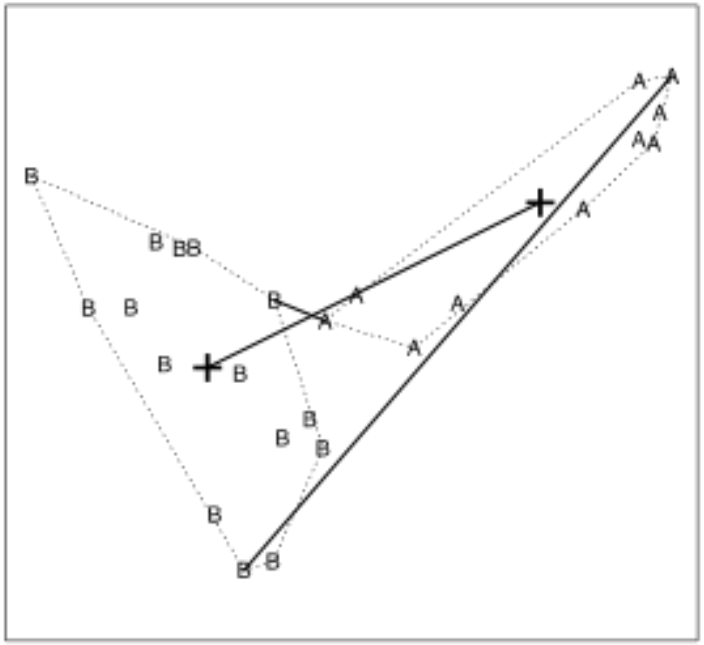
- Organize objects in dendrograms (usually binary);
- Objects more similar are closer to each other on the dendrogram
- For any set of objects one can find a dendrogram! Everything clusters!
- How to tell if the clustering is meaningful?



Hierarchical clustering algorithm

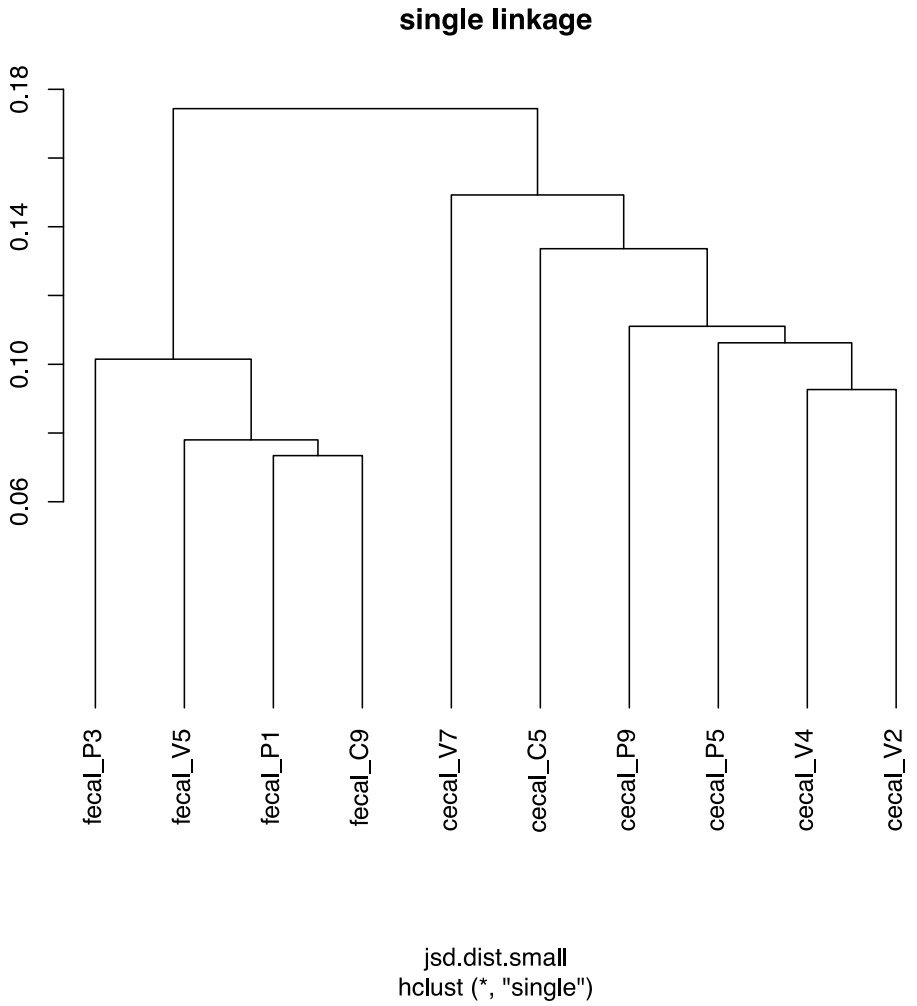
- Start with a dissimilarity matrix
- Join the 2 most closely related objects
- Remove the joined objects from the matrix
- Add a new object that represents the joint group (complete, average, single)
- Repeat until no objects remain in the matrix

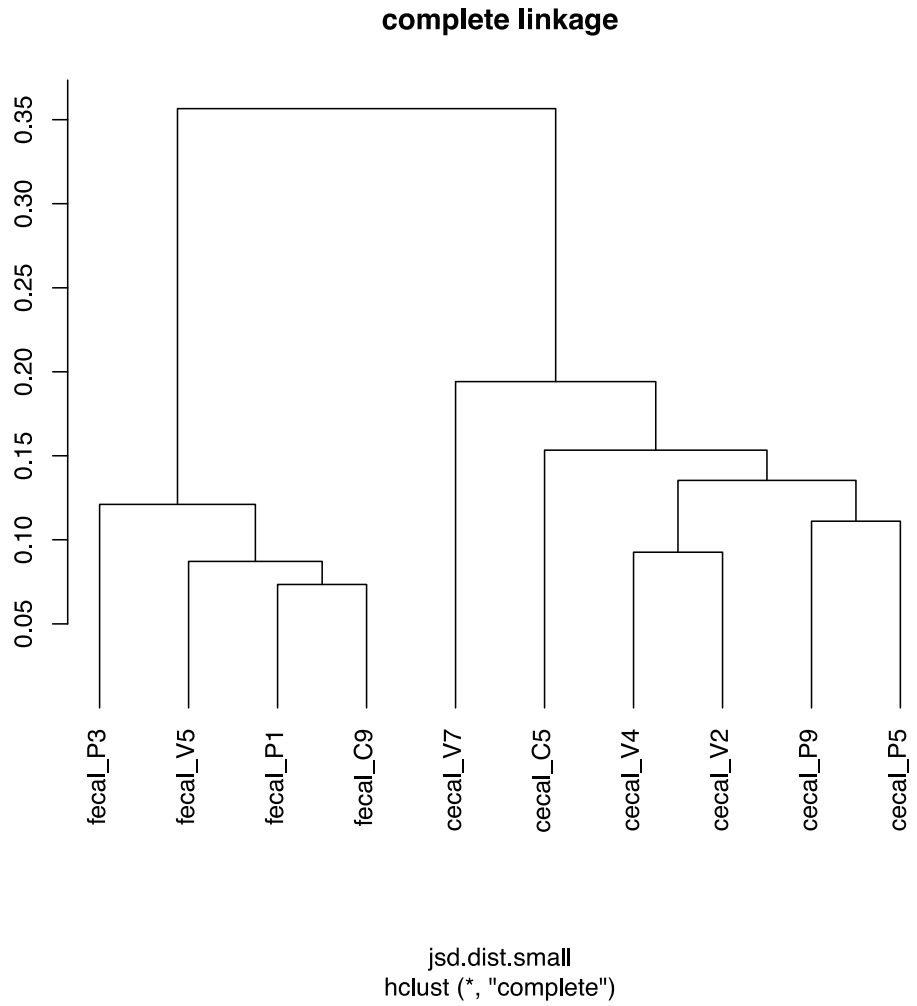
Linkage types



- Complete linkage: distance from the furthest objects apart
- Average linkage: average distance between objects
- Single linkage: distance from the closest objects apart

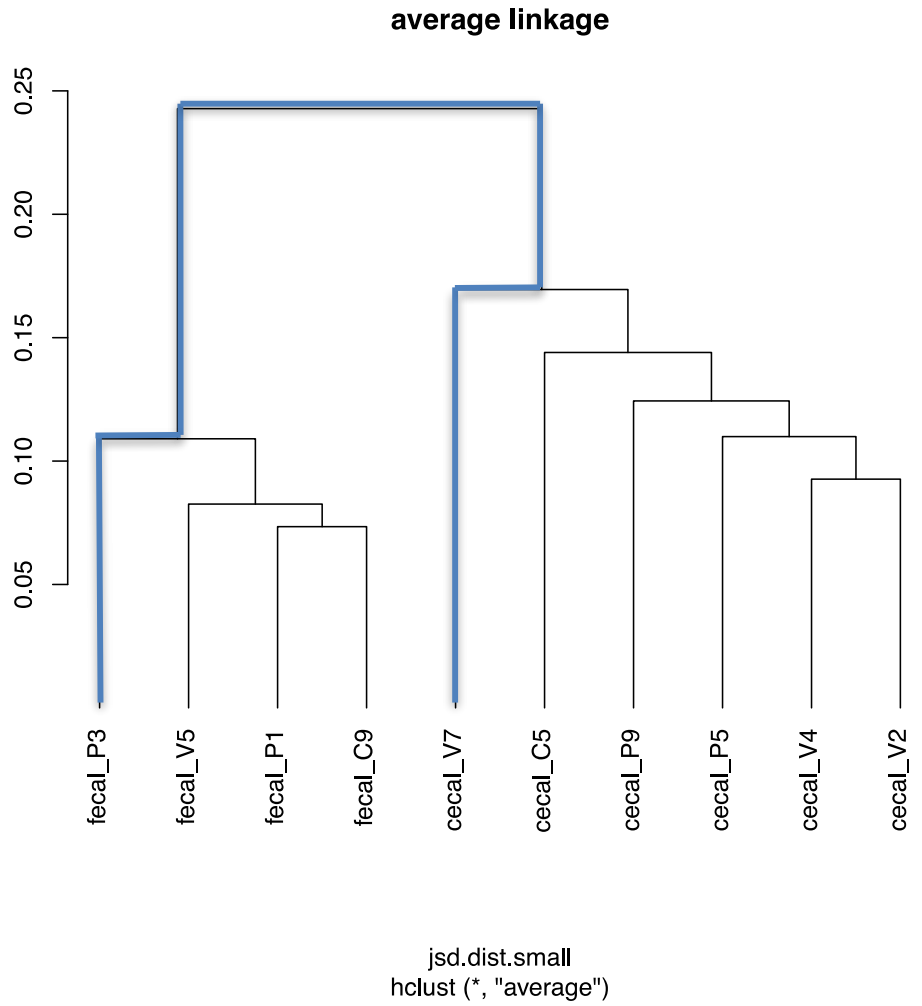
Hierarchical clustering example





Cophenetic distance

- Distance induced by the dendrogram is called *cophenetic* distance.
- This distance may be different from the original distance used to construct the dendrogram.



Clustering tools in R

`in base R`

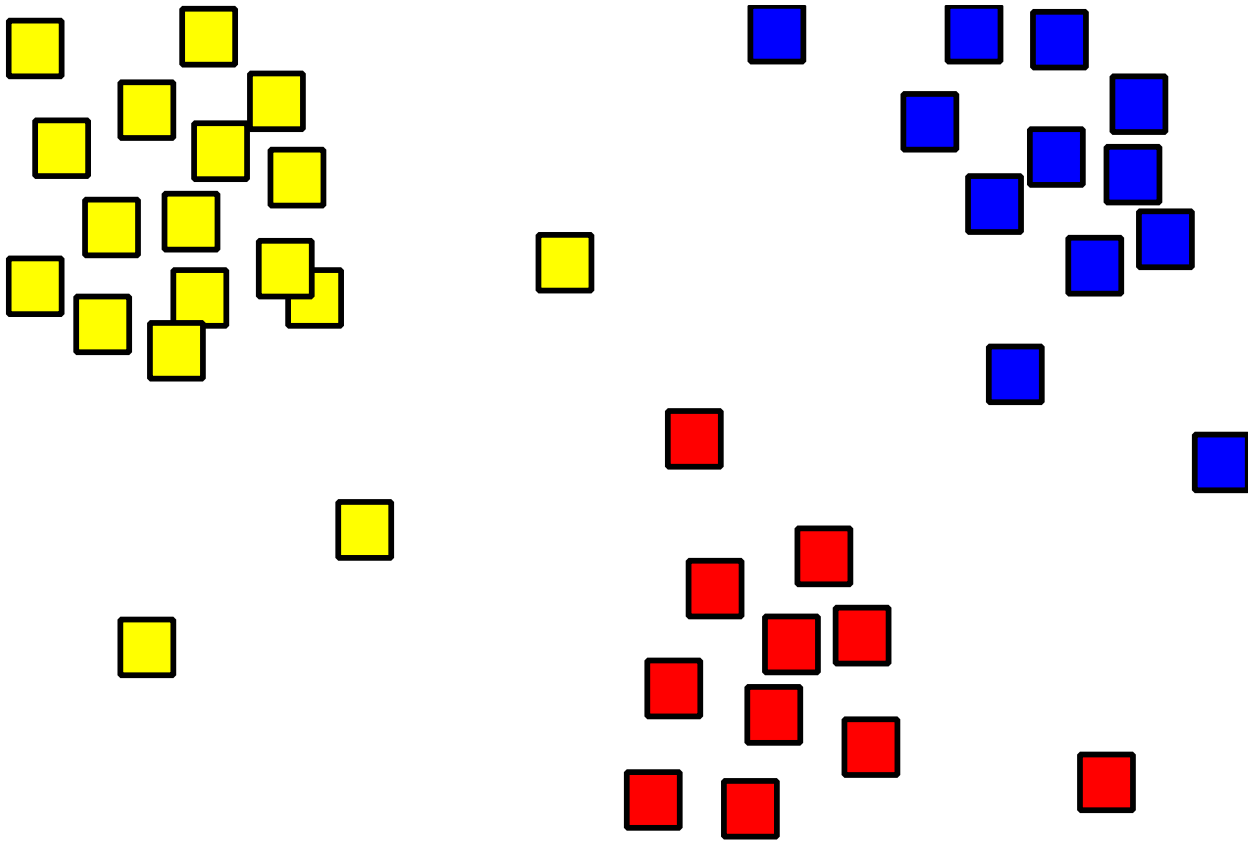
- `hclust` performs hierarchical clustering
- `cophenetic` computes cophenetic distance on the dendrogram

other packages

- CRAN Task View: Cluster Analysis & Finite Mixture Models
- https://cran.r-project.org/web/views/Cluster.html

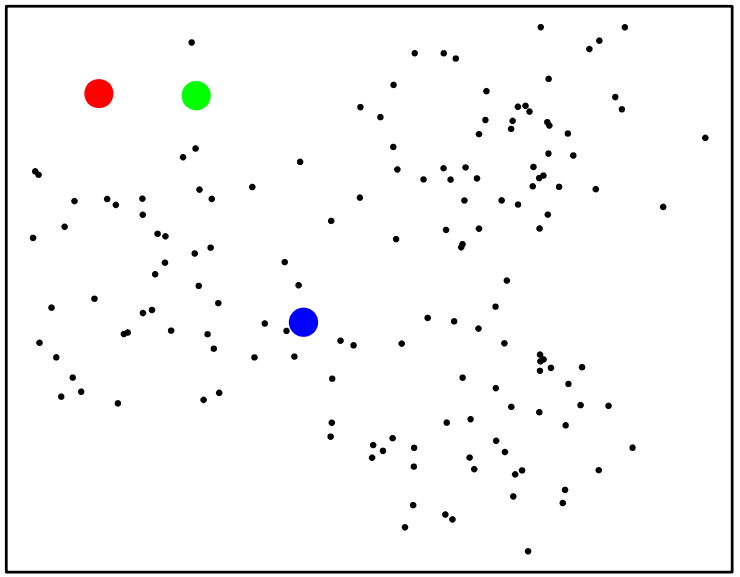
Discrete clustering

- Examples
  - K-means clustering
  - PAM (partitioning around medoids) clustering

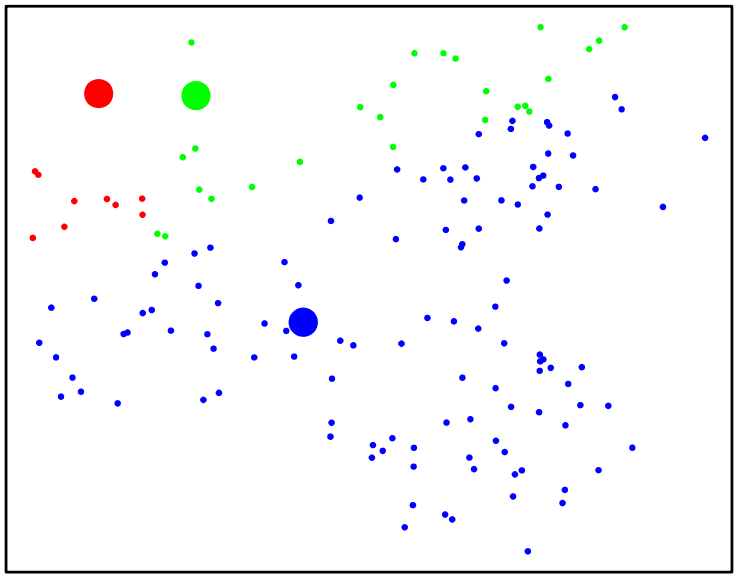


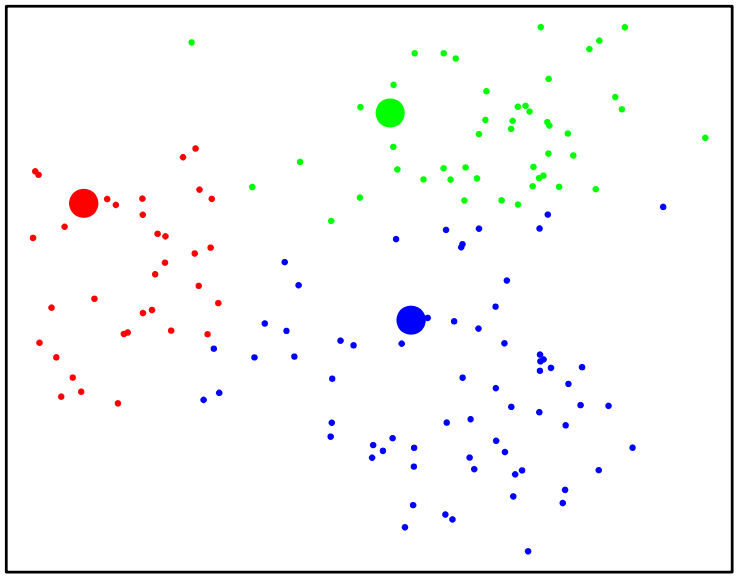
K-means clustering

- **Initialize**: Pick K random points as cluster centers
- **Iterate**:
  - Assign points to closest cluster center
  - Update cluster center location to the mean of the assigned points
- **Stop** when no points change cluster assignment (convergence)

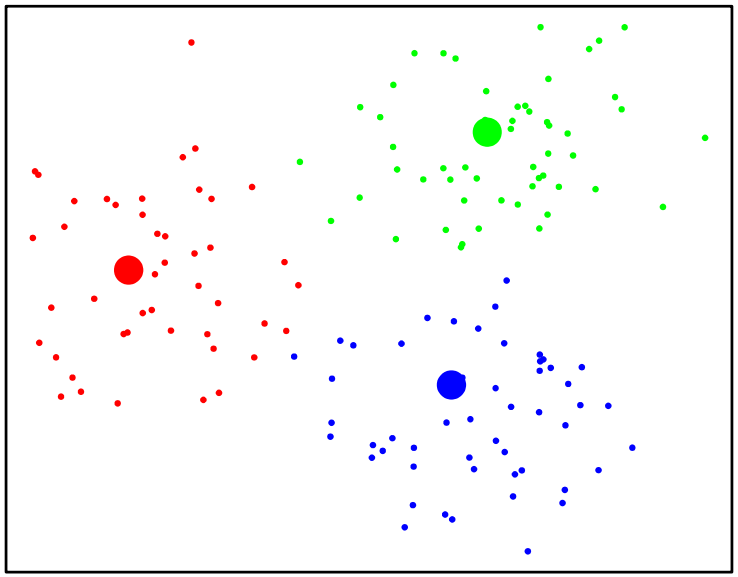


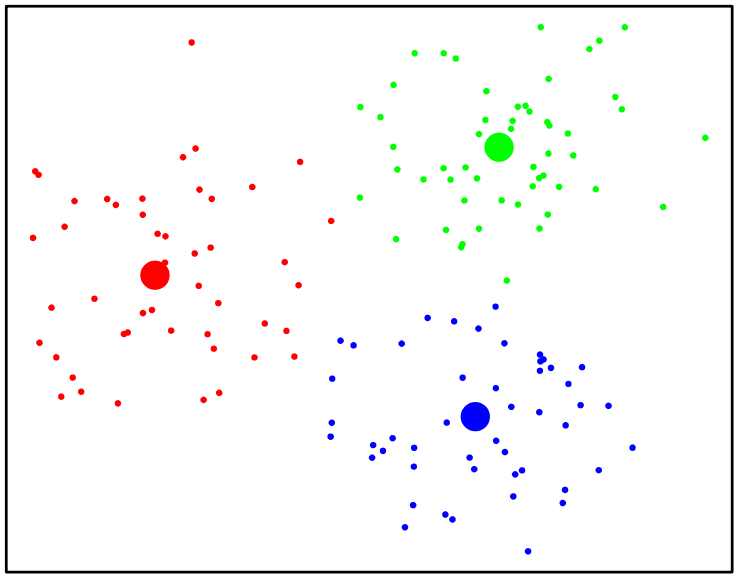
K-means clustering



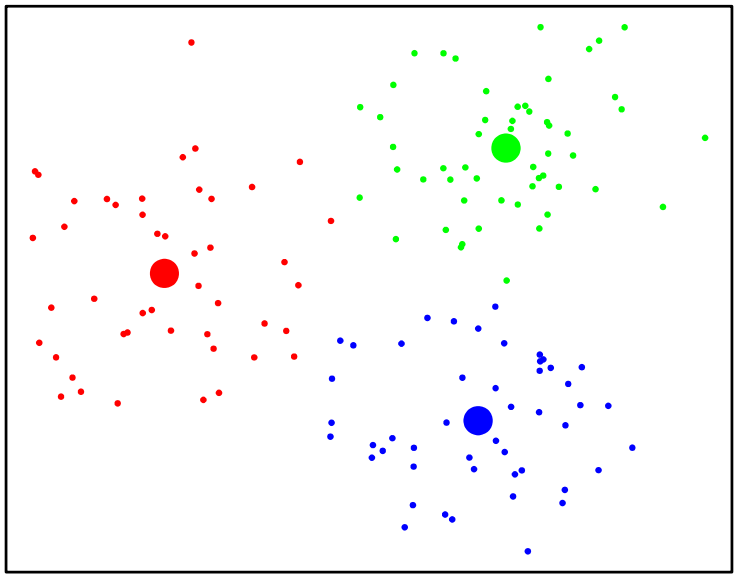


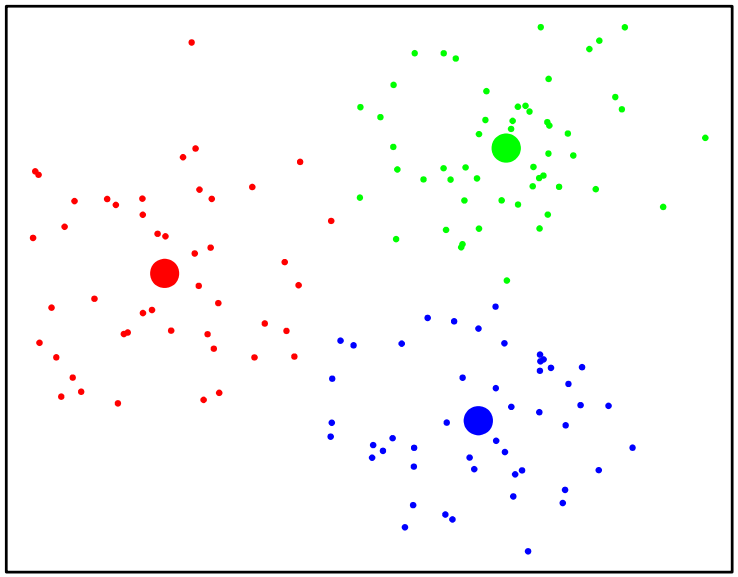
K-means clustering





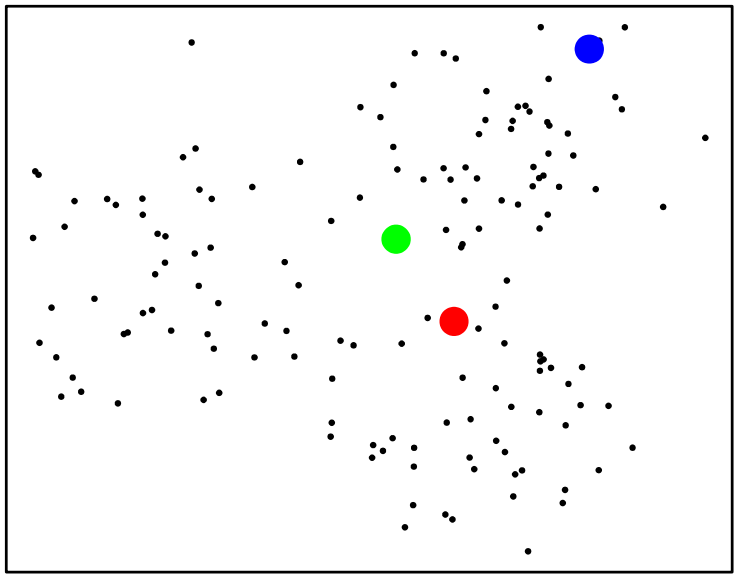
K-means clustering



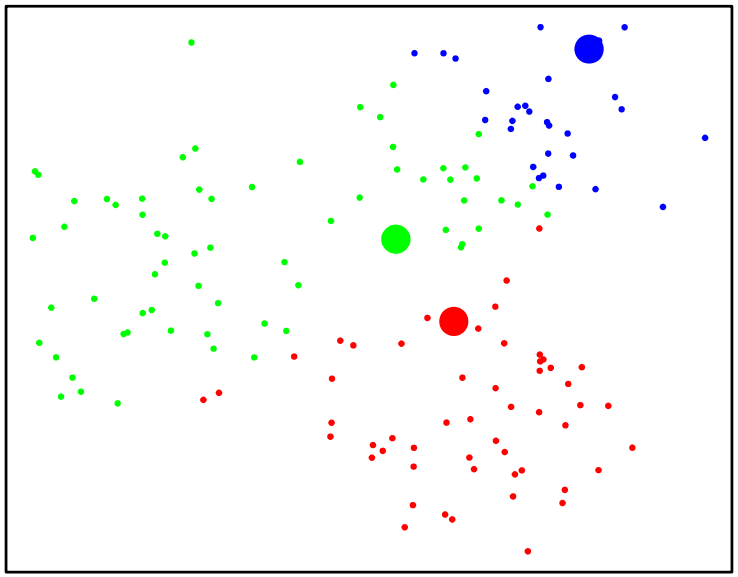


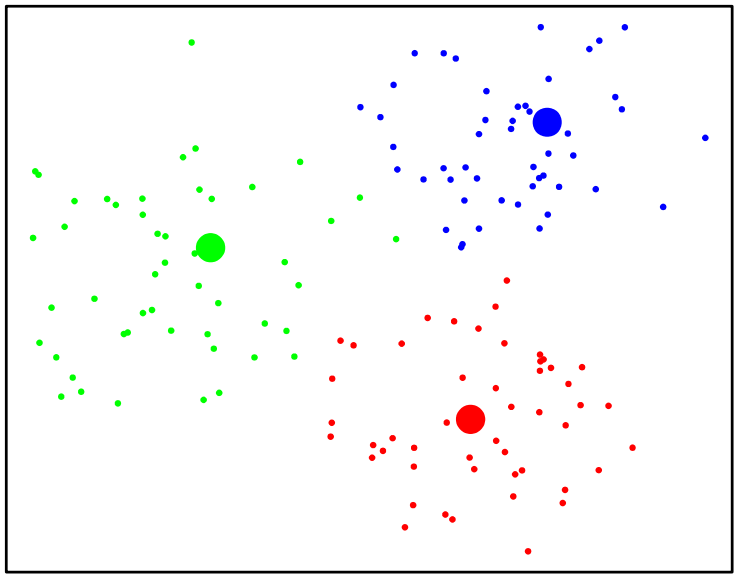
Partitioning around medoids clustering

- **Initialize**: Select K of the points to be the centers of the clusters
- **Iterate**:
  - Assign points to the closest cluster center.
  - For each cluster center:
    - Replace center with point that minimizes total distance within the cluster
- **Stop** when no cluster center has changed

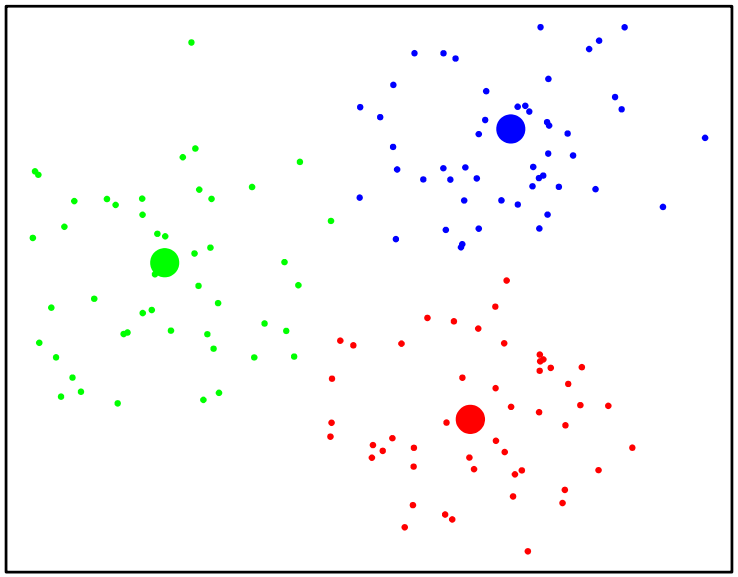


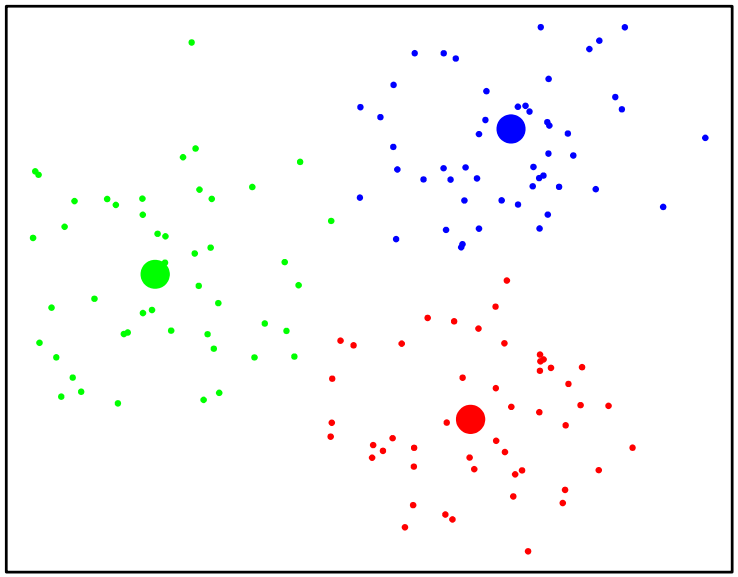
K-medoids





K-medoids





In R:

- Libraries: `cluster` and `clusterSim`
- `pam`: partitioning around medoids algorithm
- `clusGap`: gap statistic

How to select the number of clusters?

- Use measures of how good the clusters describe the structure of the data for varying number of clusters.
  - F-statistic: Calinski-Harabasz index
  - Silhouette method
  - Gap statistic: a metric based on within group distances defined using permutations

F-statistic

- Let
  - SSW is the sum of squares within clusters;
  - SSB is the sum of squares between clusters.
- F [CH-index]= (SSB/(K-1))/(SSW/(n-K))
  - Ratio of average between cluster distance and average within cluster distance
- Larger index value indicates better clustering:
  - When distance between clusters is maximized so is the F index;
  - When within cluster variability is low the index is higher.

Silhouette

- For each point i let:
  - a(i) is average distance to other objects within the same cluster;
  - b(i) distance to the closest object outside the cluster.
- $s(i) = [b(i) - a(i)]/\max(a(i), b(i))$
- $-1 \leq s(i) \leq 1$
- s(i) closer to 1 indicates best clustering; when a(i) is vanishingly small and b(i) is much larger than a(i).

Gap Statistic: How many clusters?

$$D_r = \sum_{i,i' \in C_r} d_{ii'}$$

sum of the pairwise distances for all points in cluster *r*

$$W_k = \sum_{r=1}^{k} \frac{1}{2n_r} D_r.$$

W_k - within-cluster dispersion

$$\mathrm{Gap}_n(k) = E_n^*\{\log(W_k)\} - \log(W_k),$$

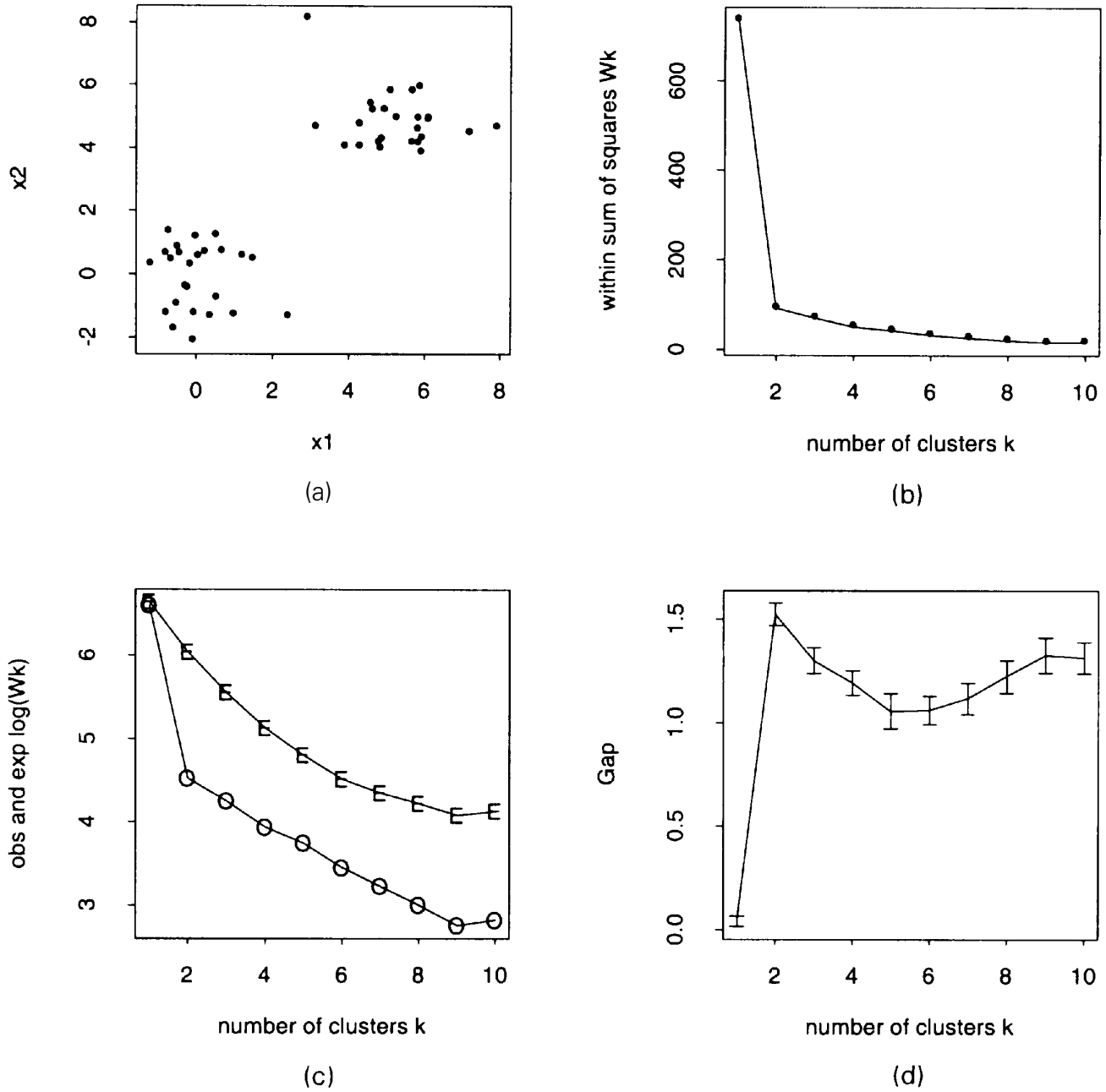
Gap Statistic:

- A statistic representing the strength of clusters, relative to null reference distribution of clustering strength, for a given number of clusters, k
- E^*_n denotes expectation for sample size n from reference
- W_k decreases monotonically as number of clusters, k, increases
- from some k onwards the decrease flattens markedly
- reference distribution defined by resampling

Tibshirani, R., Walther, G., & Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society, B*.

Gap Statistic: How many clusters?

Simple Example



Tibshirani, R., Walther, G., & Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society, B*.

Gap Statistic: How many clusters?

Gap Statistic, final notes:

- Must choose:
  - clustering algorithm
  - range of *k*
- We will observe gap statistic behavior graphically (sanity check)
- Also choose formal definition for optimal clustering:

$$\hat{k} = \text{smallest } k \text{ such that } \text{Gap}(k) \geqslant \text{Gap}(k+1) - s_{k+1}.$$

where s_{k+1} is standard error of gap statistic at *k + 1*

Tibshirani, R., Walther, G., & Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society, B*.

SUPERVISED LEARNING: CLASSIFICATION

Main elements of predictive modeling

1. **Model selection**

   Out of many possible models find the ones that are most probably accurate (and also have other desired properties).

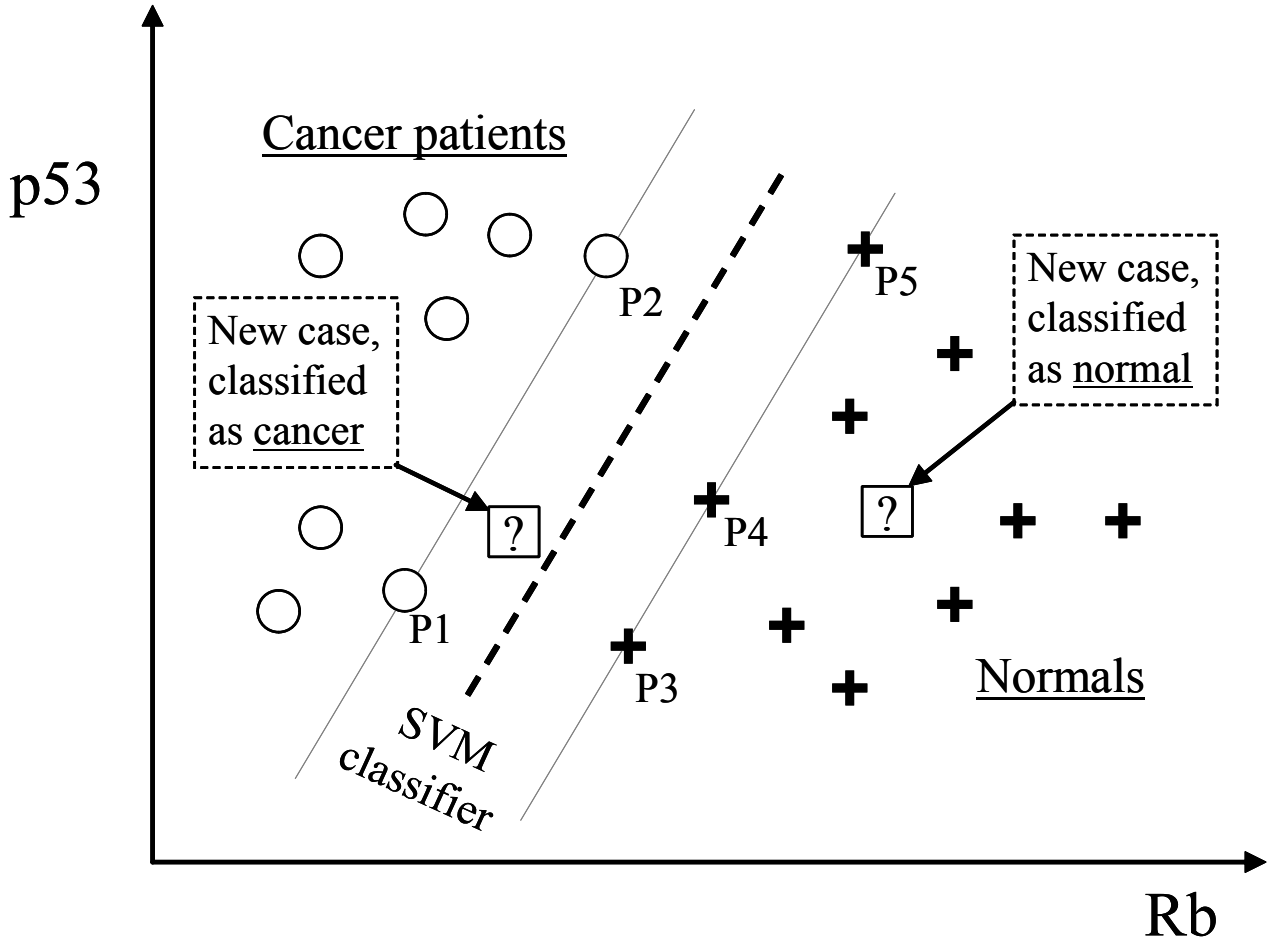
2. **Error estimation**

   Estimate how accurate the final model will be in future applications (i.e., in the population where we sampled from).

Very important Model Selection + Error Estimation approach:

Repeated Nested n-Fold Cross Validation (RNCV)

Supervised learning: a geometrical interpretation



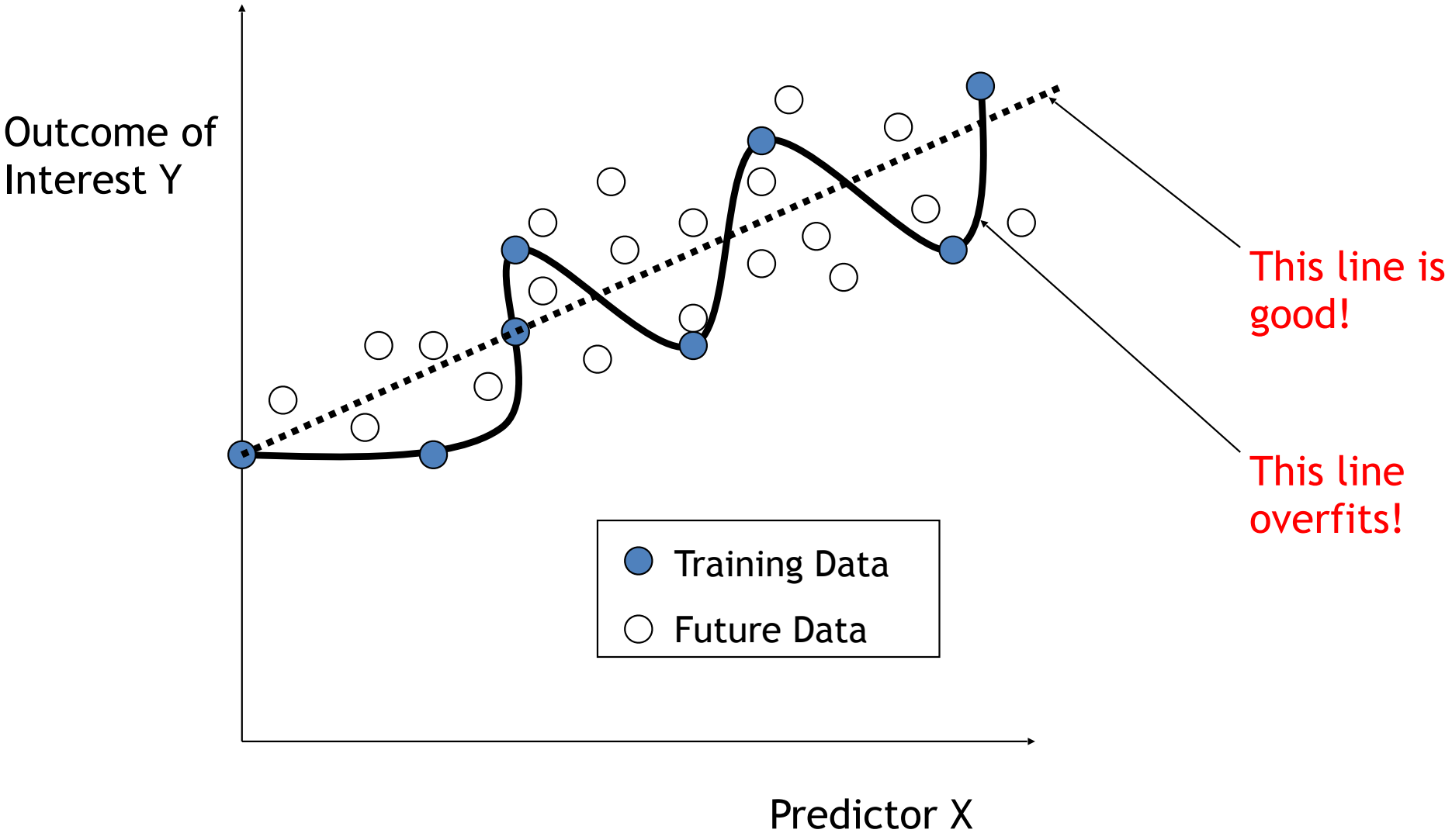
“Curses” of High-dimensionality (especially with small samples):

- Some methods may not compute at all
  - e.g. classical multiple regression
- Some methods give bad results
  - KNN
  - Decision trees
- Very slow analysis
- Very expensive/cumbersome clinical application
- Very easy to overfit if you’re not diligent

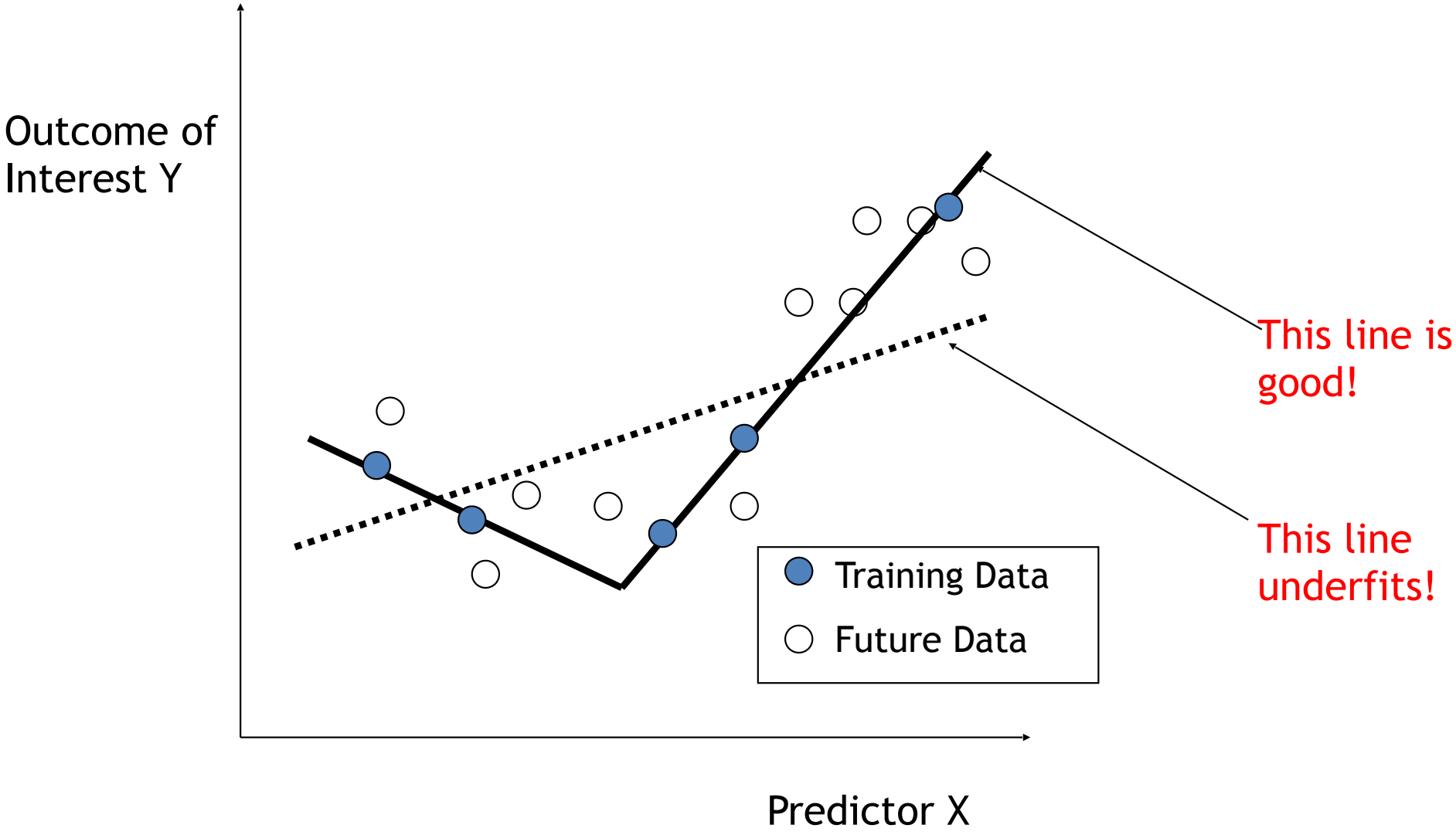
Two (very real and very unpleasant) problems: Over-fitting & Under-fitting

- Over-fitting (a model to your data)
  - building a model that is good in original data but fails to generalize well to fresh data
- Under-fitting (a model to your data)
  - building a model that is poor in both original data and fresh data

Over/under-fitting are directly related to the complexity of the decision surface and goodness of fit



Over/under-fitting are directly related to the complexity of the decision surface and goodness of fit



Over/under-fitting are directly related to the complexity of the decision surface and goodness of fit

Successful data analysis methods balance training data fit with complexity

What is the relationship between overfitting and high dimensionality?

1. **Overfitting**: when we create a model that accurately captures characteristics of our discovery dataset but fails to perform well in the populations where the discovery data was sampled from.
2. All else being equal, high dimensionality makes overfitting easier to occur… Our data is usually high-dimensional…

Cross Validation: A general method for balancing tendency to under/over-fit

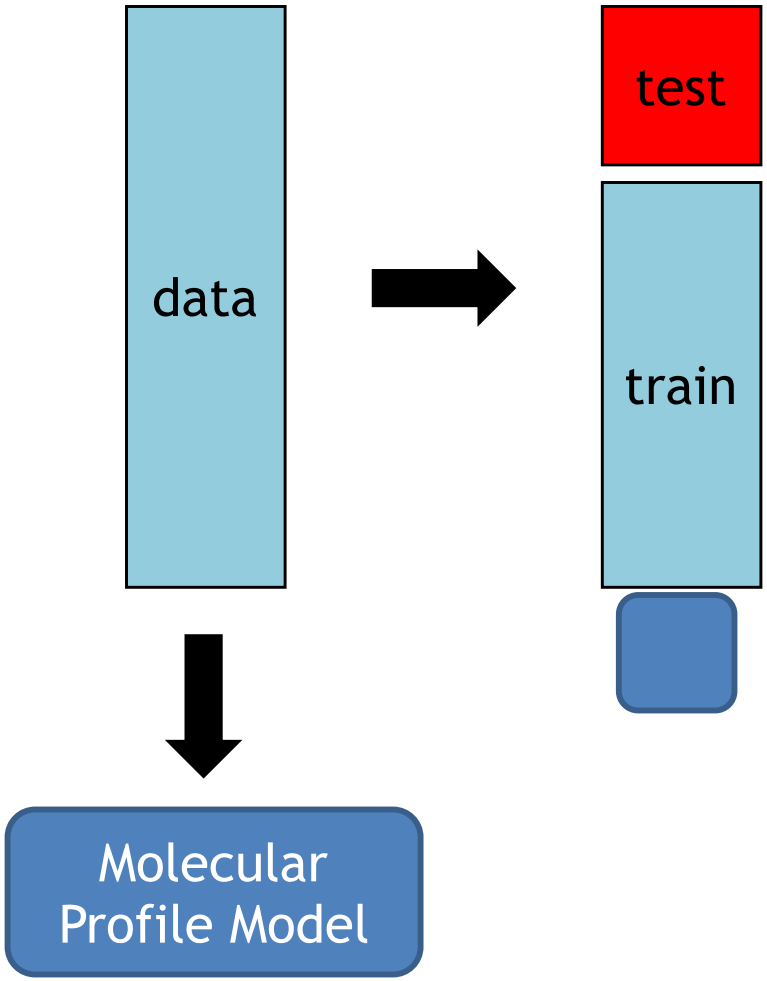
```
Define sets of model parameter values to evaluate
for each parameter set do
    for each resampling iteration do
        Hold–out specific samples
        [Optional] Pre–process the data
        Fit the model on the remainder
        Predict the hold–out samples
    end
    Calculate the average performance across hold–out predictions
end
Determine the optimal parameter set
Fit the final model to all the training data using the optimal parameter set
```

Cross Validation: A general method for balancing tendency to under/over-fit

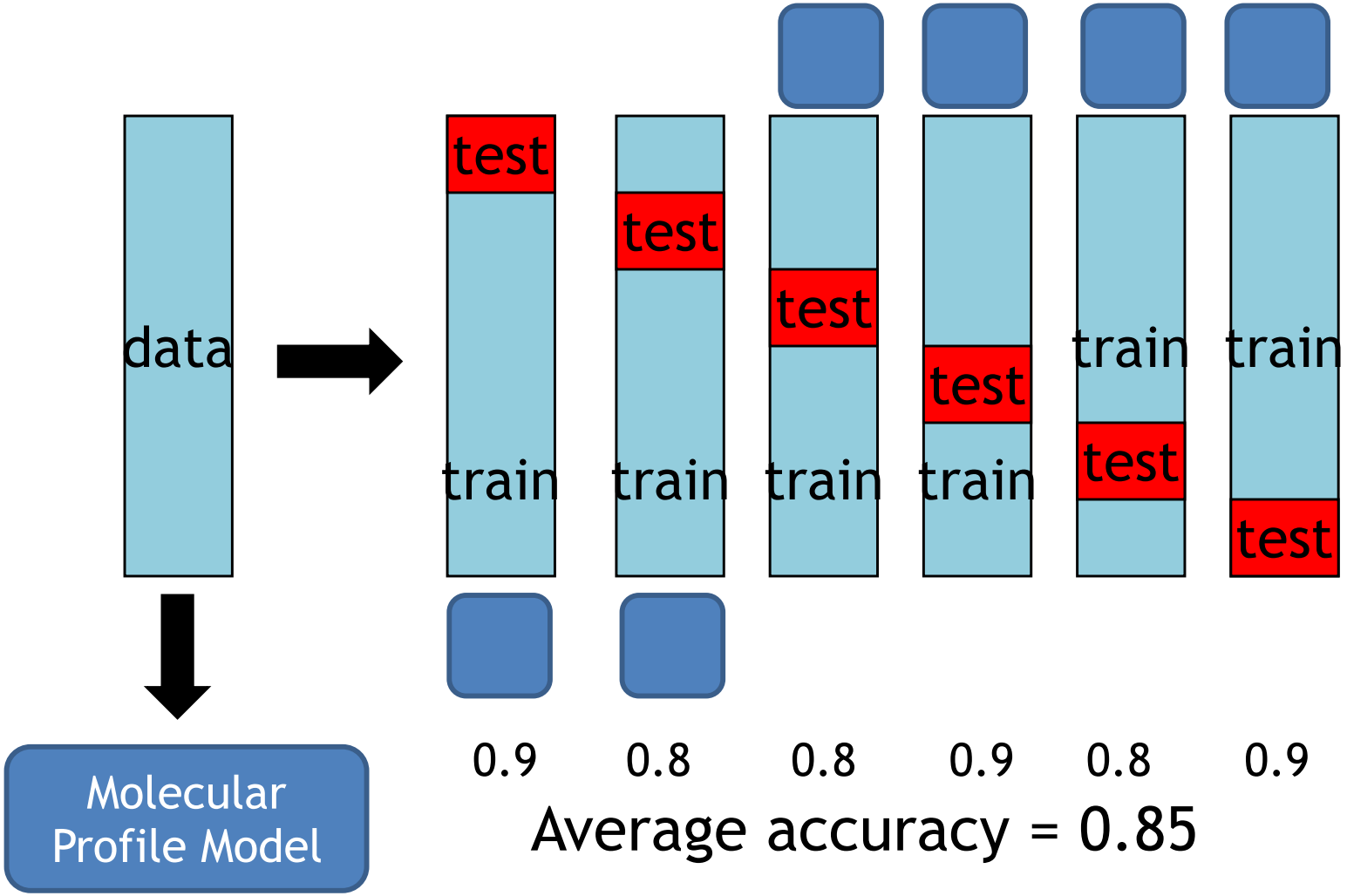
```
Define sets of model parameter values to evaluate
for each parameter set do
    for each resampling iteration do
        Hold–out specific samples
        [Optional] Pre–process the data
        Fit the model on the remainder
        Predict the hold–out samples
    end
    Calculate the average performance across hold–out predictions
end
Determine the optimal parameter set
Fit the final model to all the training data using the optimal parameter set
```

A common mistake is to do this outside the loops! Why is that bad?

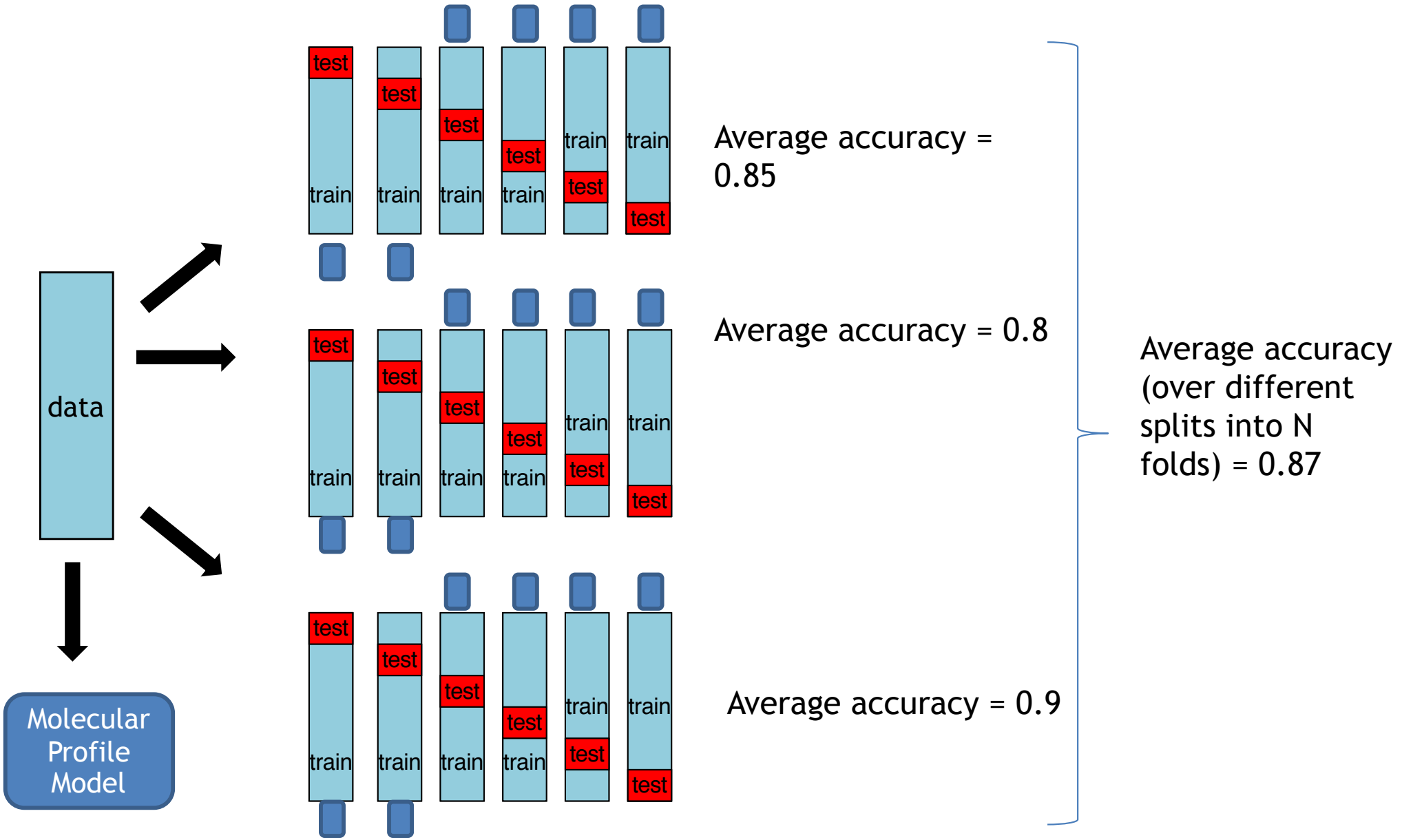
Hold-out validation method



N-Fold Cross-validation



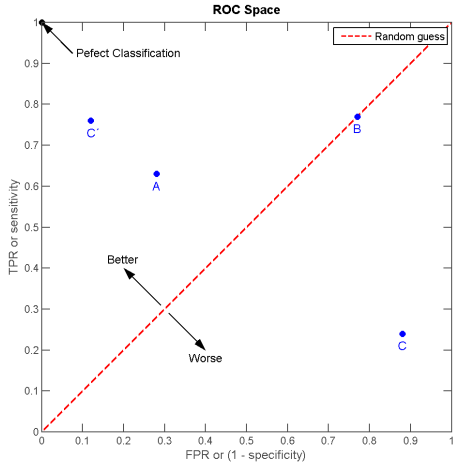
Repeated N-Fold Cross-validation



Measures of classification error

- Accuracy: proportion of correct classifications
  - The number of times the classifier gives the correct result divided by the total number of test cases.
- Area under receiver-operator characteristic curve (AUC).

		Truth		
		Positive	Negative	
Test outcome	Positive	True positive	False positive	**Precision:** #TP/ #PPositives
	Negative	False negative	True negative	**NPP:** #TN/ #PNegative
		Sensitivity: #TP/ #Positives	**Specificity:** #TN/ #Negatives	**Accuracy**



http://en.wikipedia.org/wiki/File:ROC_space-2.png

Comparison of State of the Art Methods for Microbiomic marker + Signature Discovery 1

A comprehensive evaluation of multicategory classification methods for microbiomic data

Alexander Statnikov[1,2,§], Mikael Henaff[1], Varun Narendra[1], Kranti Konganti[5], Zhiguo Li[1], Liying Yang[2], Zhiheng Pei[2,3], Martin J. Blaser[2,4], Constantin F. Aliferis[1,3,6], Alexander V. Alekseyenko[1,2,§]

Problem: It is currently unknown which classifiers perform best among the many available alternatives for classification with microbiomic data linking abundances of microbial taxa to phenotypic and physiological states, which can inform development of new diagnostic, personalized medicine, and forensic modalities

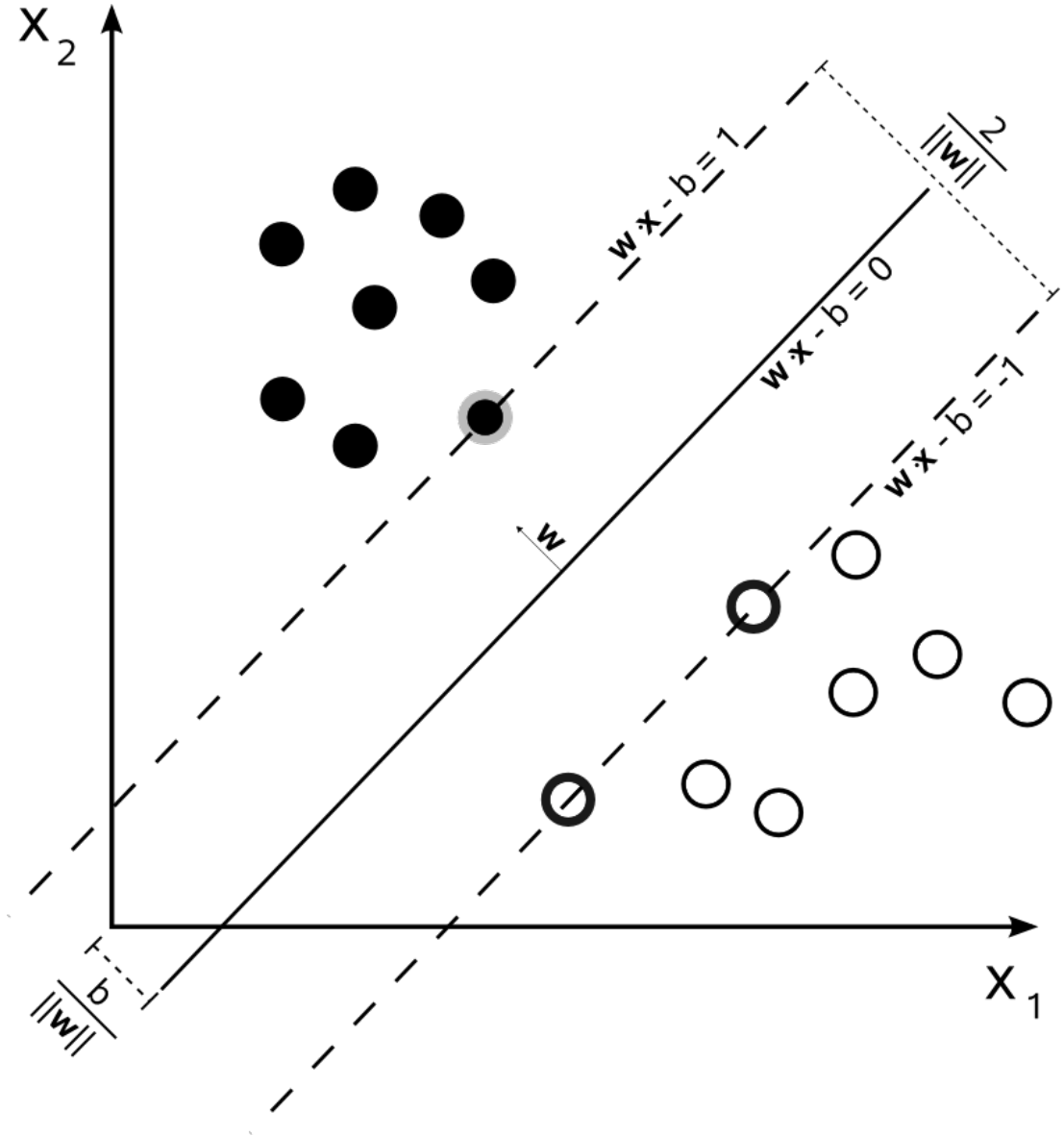
Results: In this work, we performed a systematic comparison of 18 major classification methods, 5 feature selection methods, and 2 accuracy metrics using 8 datasets spanning 1,802 human samples and various classification tasks: body site and subject classification and diagnosis.

Conclusions: We found that random forests, support vector machines, kernel ridge regression, and Bayesian logistic regression with Laplace priors are the most effective machine learning techniques for performing accurate classification from these microbiomic data.

Classification techniques

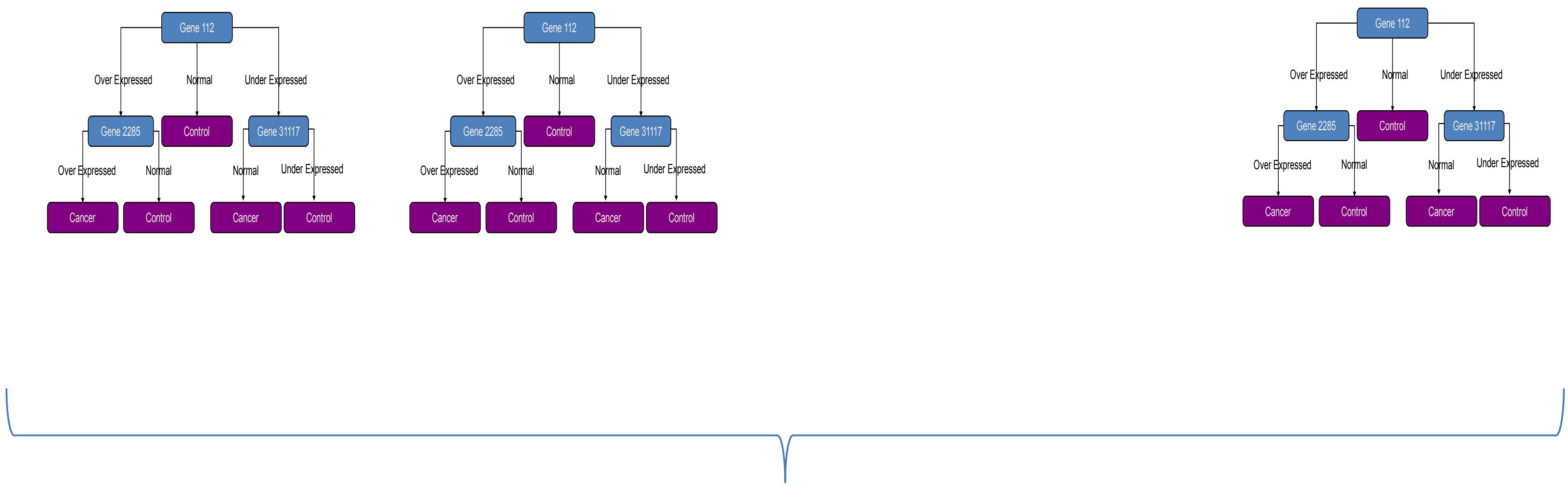
- Support Vector Machines
- Random Forests

Support Vector Machines



http://en.wikipedia.org/wiki/File:Svm_max_sep_hyperplane_with_margin.png

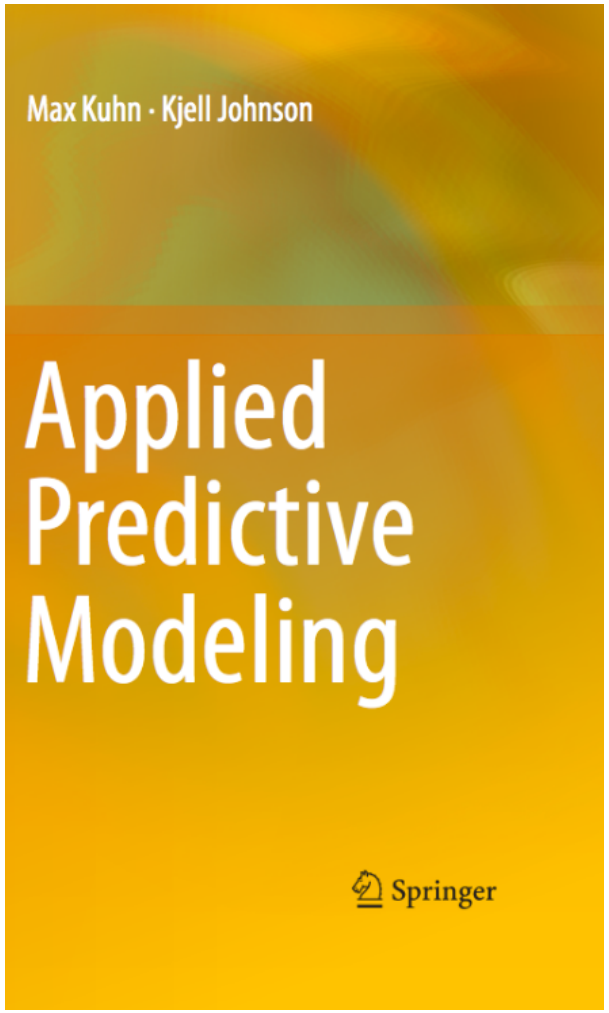
Random Forest

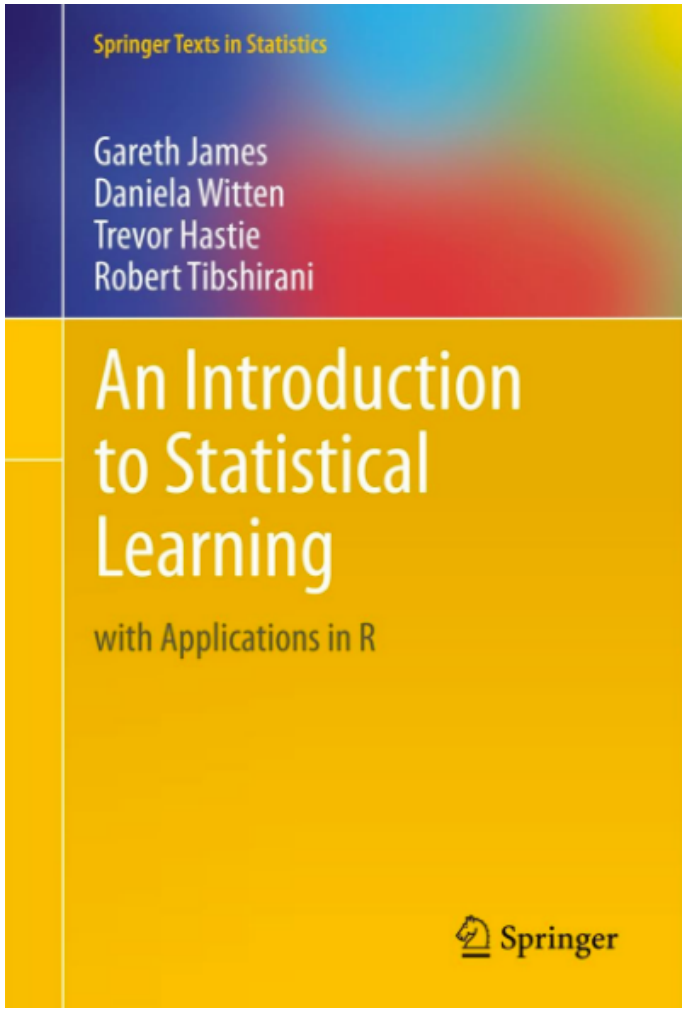


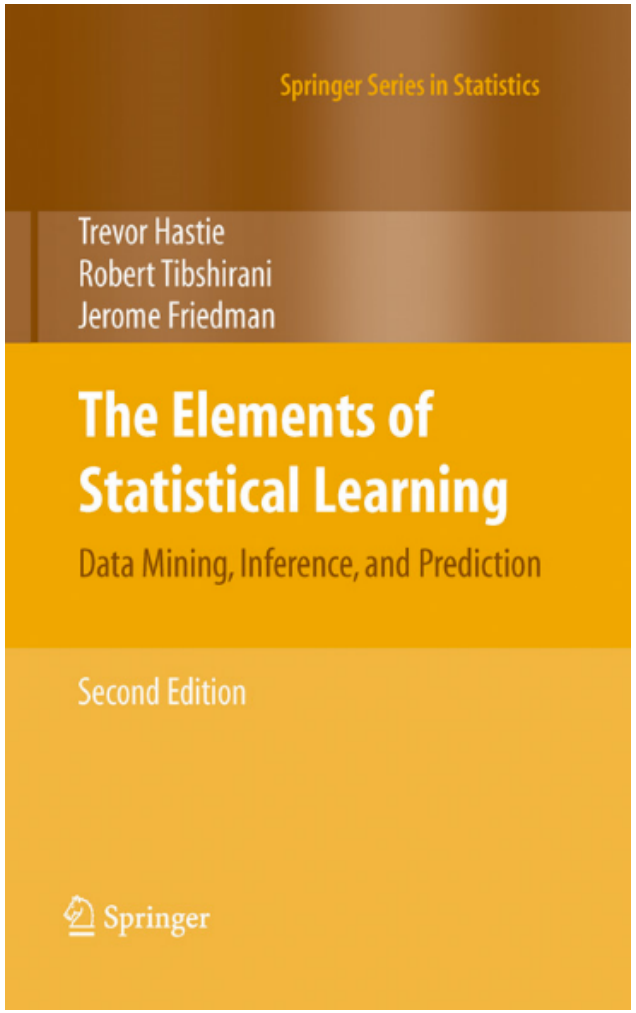
Random Forests v. SVM…

1. RFs perform well, on par with SVMs in terms of predictive accuracy (depends on data).
2. RFs are slower than SVMs for typical HD molecular datasets.
3. RFs do not require to set up variable selection, model selection and error estimation separately because it is embedded (CV still recommended, though).
4. RFs often produce large, complicated, hard to explain models ("Black box")

Recommended Texts







Recommended Packages

The caret package (short for Classification And REgression Training) is a set of functions that attempt to streamline the process for creating predictive models. The package contains tools for:

- data splitting
- pre-processing
- feature selection
- model tuning using resampling
- variable importance estimation

http://topepo.github.io/caret/modelList.html