Data Management for Data Science

Lecture 19: Unsupervised Learning/Ensemble Learning

Prof. Asoc. Endri Raço

Today

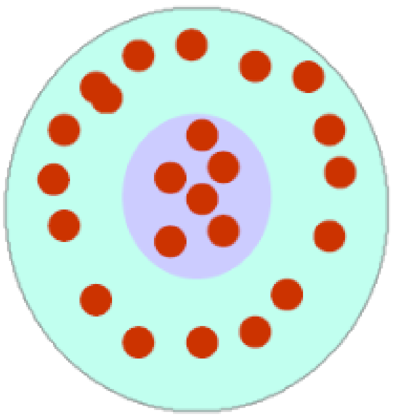
1. Unsupervised Learning/Clustering

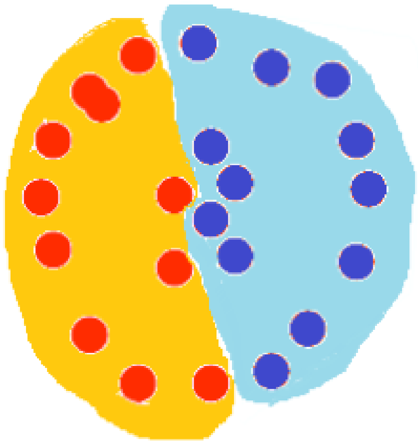
2. K-Means

3. Ensembles and Gradient Boosting

What is clustering?

- The organization of unlabeled data into similarity groups called clusters.
- A cluster is a collection of data items which are “similar” between them, and “dissimilar” to data items in other clusters.





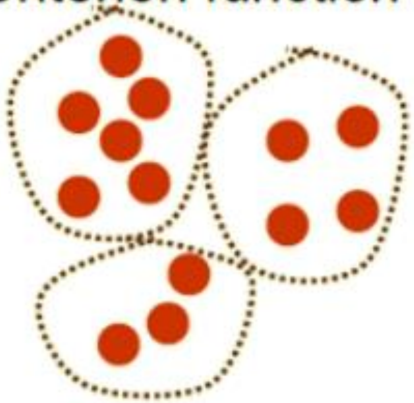
What do we need for clustering?

1. Proximity measure, *either*
   - similarity measure $\boldsymbol{s}(\boldsymbol{x}_i,\boldsymbol{x}_k)$: large if $\boldsymbol{x}_i,\boldsymbol{x}_k$ are similar
   - dissimilarity(or distance) measure $\boldsymbol{d}(\boldsymbol{x}_i,\boldsymbol{x}_k)$: small if $\boldsymbol{x}_i,\boldsymbol{x}_k$ are similar

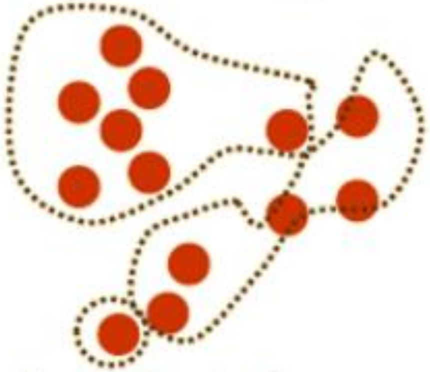
*large **d**, small **s***

*large **s**, small **d***

2. Criterion function to evaluate a clustering



good clustering



bad clustering

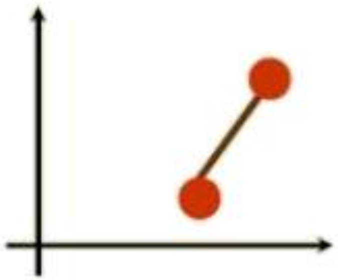
3. Algorithm to compute clustering
   - For example, by optimizing the criterion function

Distance (dissimilarity) measures

- Euclidean distance

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{k=1}^{d} \left(x_i^{(k)} - x_j^{(k)}\right)^2}$$

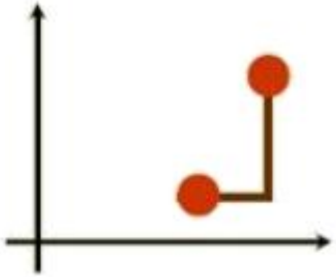
  - translation invariant



- Manhattan (city block) distance

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^{d} \left|x_i^{(k)} - x_j^{(k)}\right|$$

  - approximation to Euclidean distance, cheaper to compute



- They are special cases of **Minkowski distance:**

$$d_p(\mathbf{x}_i, \mathbf{x}_j) = \left(\sum_{k=1}^{m} \left|x_{ik} - x_{jk}\right|^p\right)^{\frac{1}{p}}$$

(p is a positive integer)

Cluster evaluation (a hard problem)

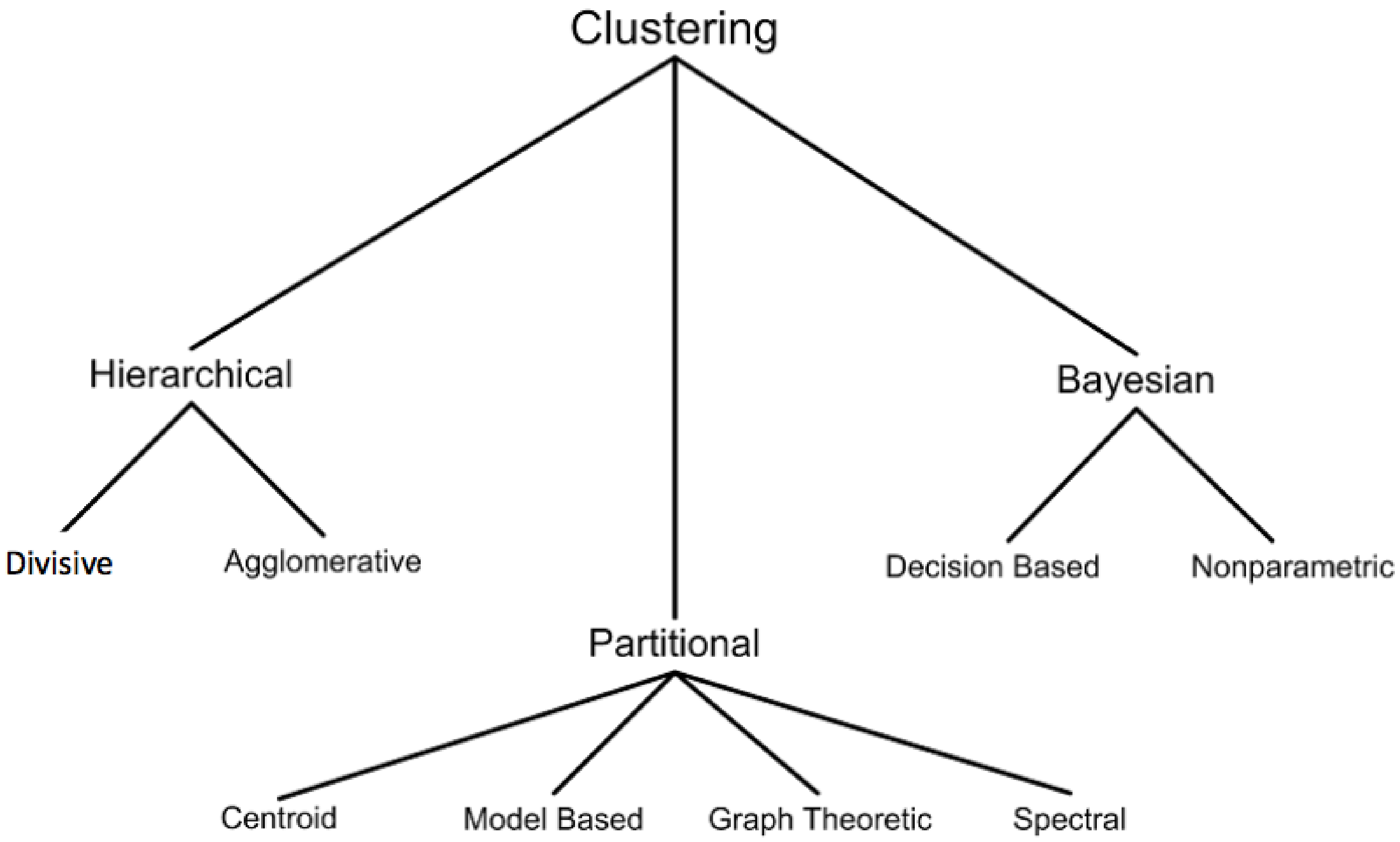
- **Intra-cluster cohesion** (compactness):
  - Cohesion measures how near the data points in a cluster are to the cluster centroid.
  - Sum of squared error (SSE) is a commonly used measure.
- **Inter-cluster separation** (isolation):
  - Separation means that different cluster centroids should be far away from one another.
- In most applications, expert judgments are still the key

How many clusters?

3 clusters or 2 clusters?

- Possible approaches
  1. fix the number of clusters to ***k***
  2. find the best clustering according to the criterion function (number of clusters may vary)

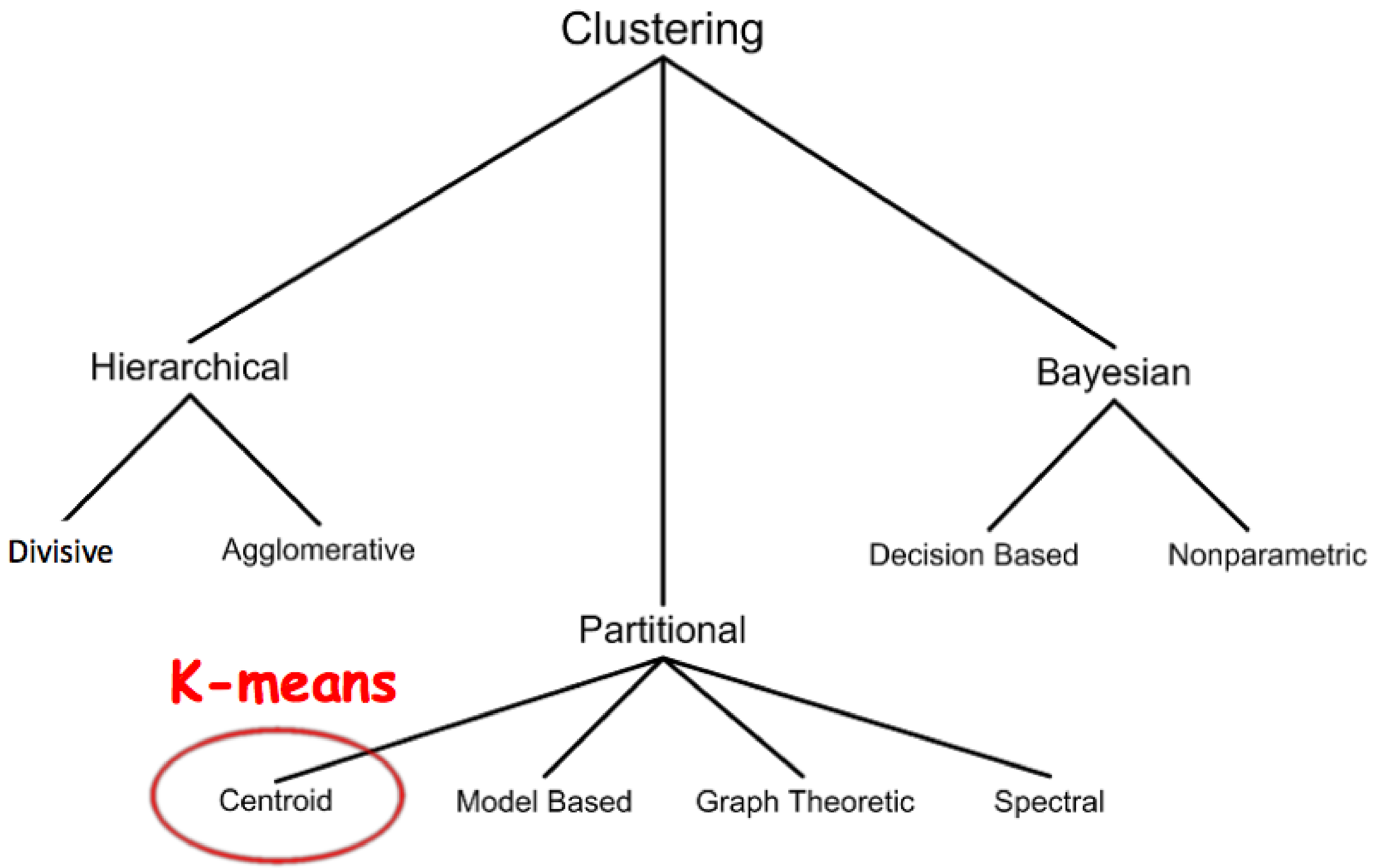
Clustering techniques



Clustering techniques

- **Hierarchical** algorithms find successive clusters using previously established clusters. These algorithms can be either agglomerative ("*bottom-up*") or divisive ("*top-down*"):
  1. Agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters;
  2. Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters.
- **Partitional** algorithms typically determine all clusters at once, but can also be used as divisive algorithms in the hierarchical clustering.
- **Bayesian** algorithms try to generate a *posteriori distribution* over the collection of all partitions of the data.

Clustering techniques



K-means

- K-means (MacQueen, 1967) is a partitional clustering algorithm
- Let the set of data points D be $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$,

  where $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ir})$ is a vector in $X \subseteq R^r$, and r is the number of dimensions.
- The k-means algorithm partitions the given data into k clusters:
  - Each cluster has a cluster **center**, called **centroid**.
  - k is specified by the user

K-means algorithm

- Given *k*, the *k-means* algorithm works as follows:
  1. Choose *k* (random) data points (seeds) to be the initial centroids, cluster centers
  2. Assign each data point to the closest centroid
  3. Re-compute the centroids using the current cluster memberships
  4. If a convergence criterion is not met, repeat steps 2 and 3

K-means convergence (stopping criterion)

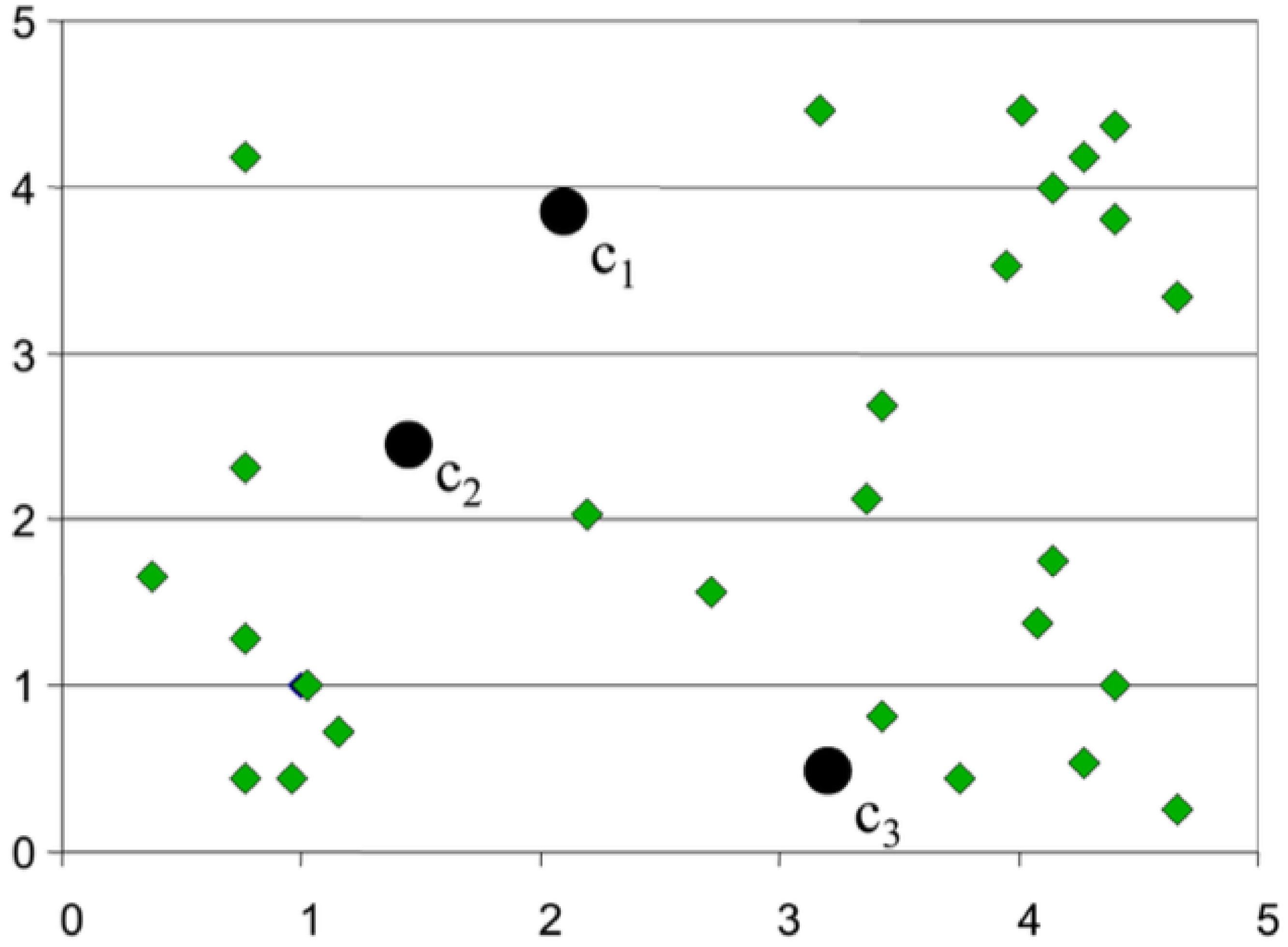
- no (or minimum) re-assignments of data points to different clusters, *or*
- no (or minimum) change of centroids, *or*
- minimum decrease in the **sum of squared error** (SSE),

$$SSE = \sum_{j=1}^{k} \sum_{\mathbf{x} \in C_j} d(\mathbf{x}, \mathbf{m}_j)^2$$

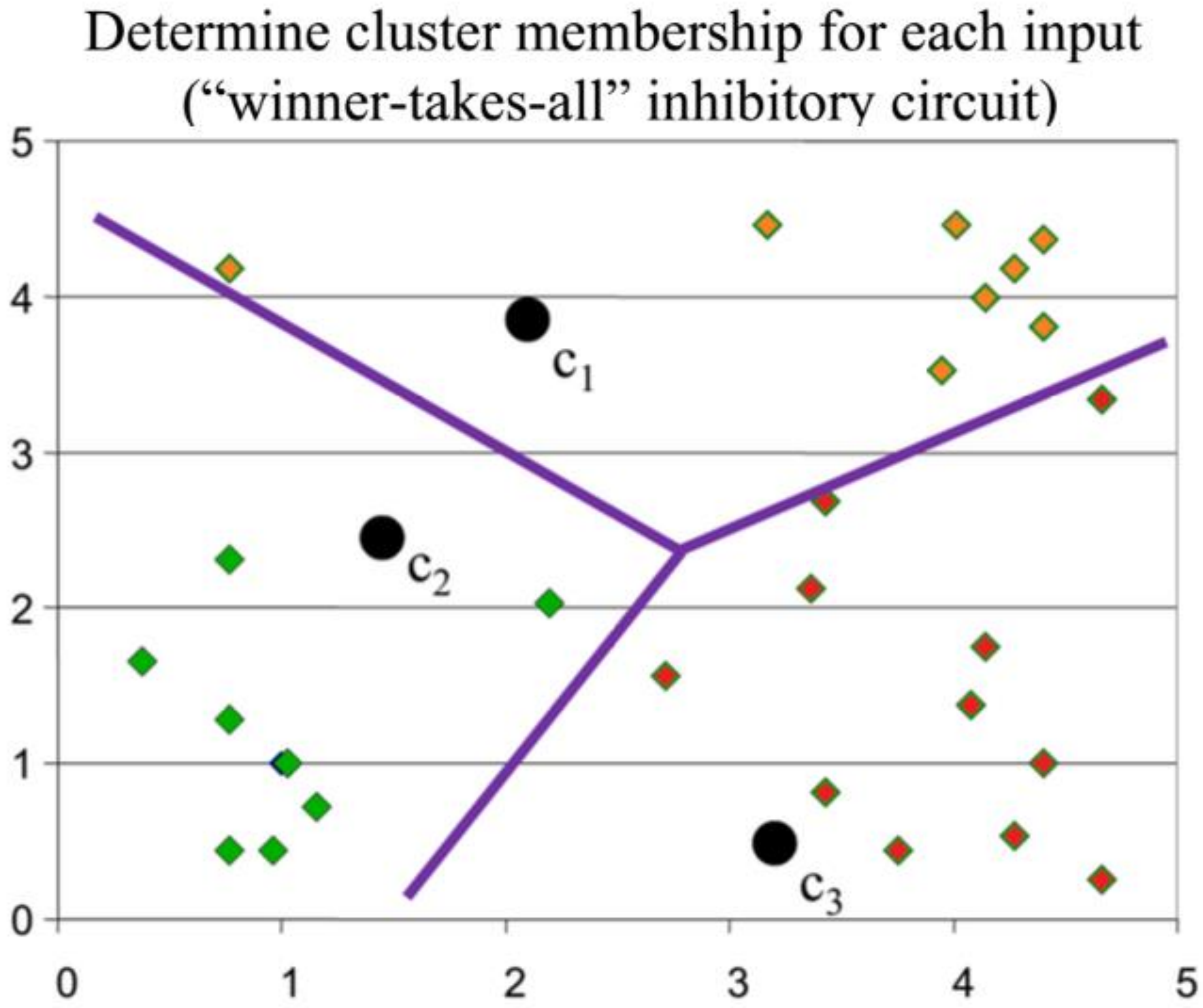
  - C_j is the jth cluster,
  - $\mathbf{m}_j$ is the centroid of cluster C_j (the mean vector of all the data points in C_j),
  - $d(\mathbf{x}, \mathbf{m}_j)$ is the (Eucledian) distance between data point $\mathbf{x}$ and centroid $\mathbf{m}_j$.

K-means clustering example: step 1

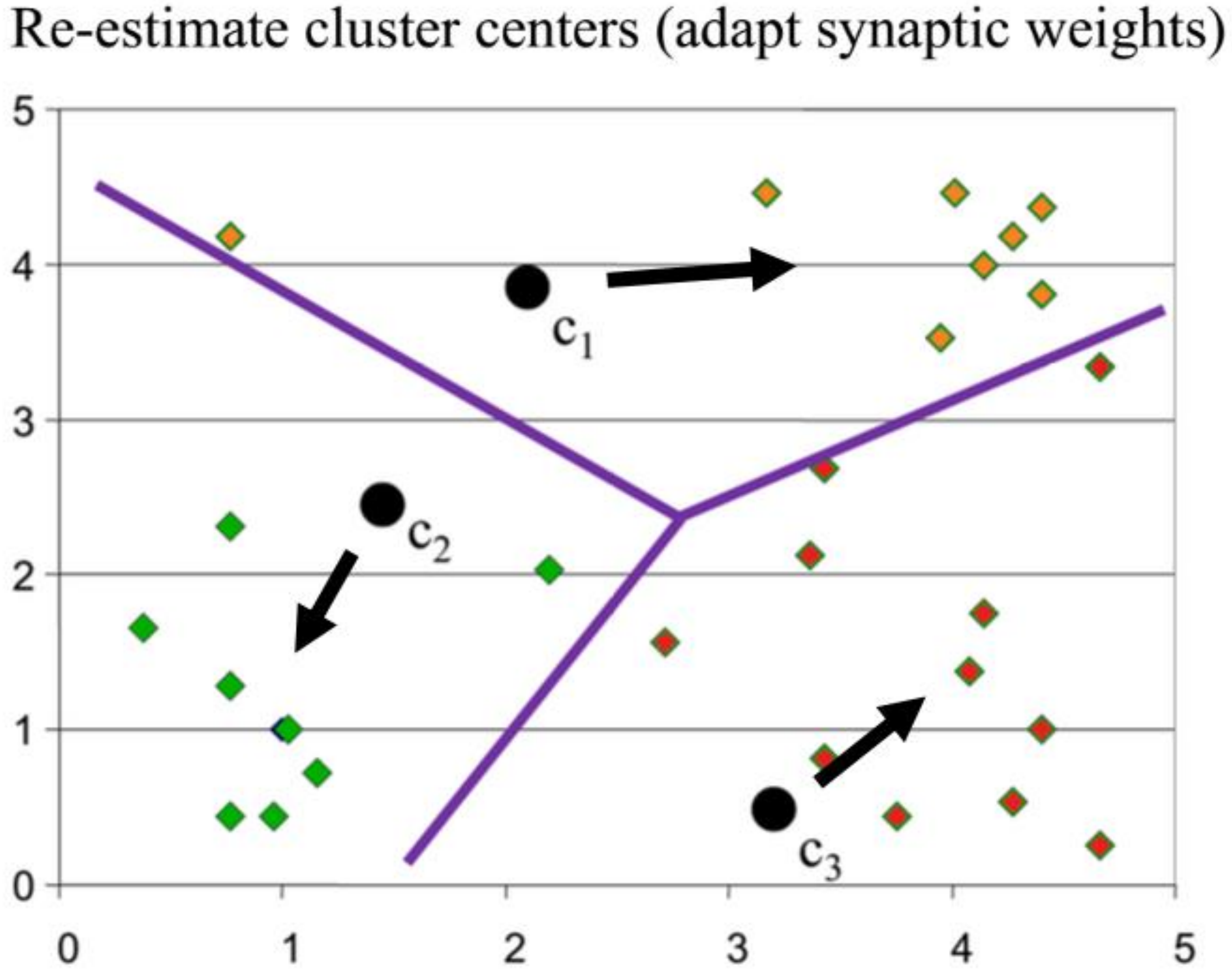
Randomly initialize the cluster centers (synaptic weights)



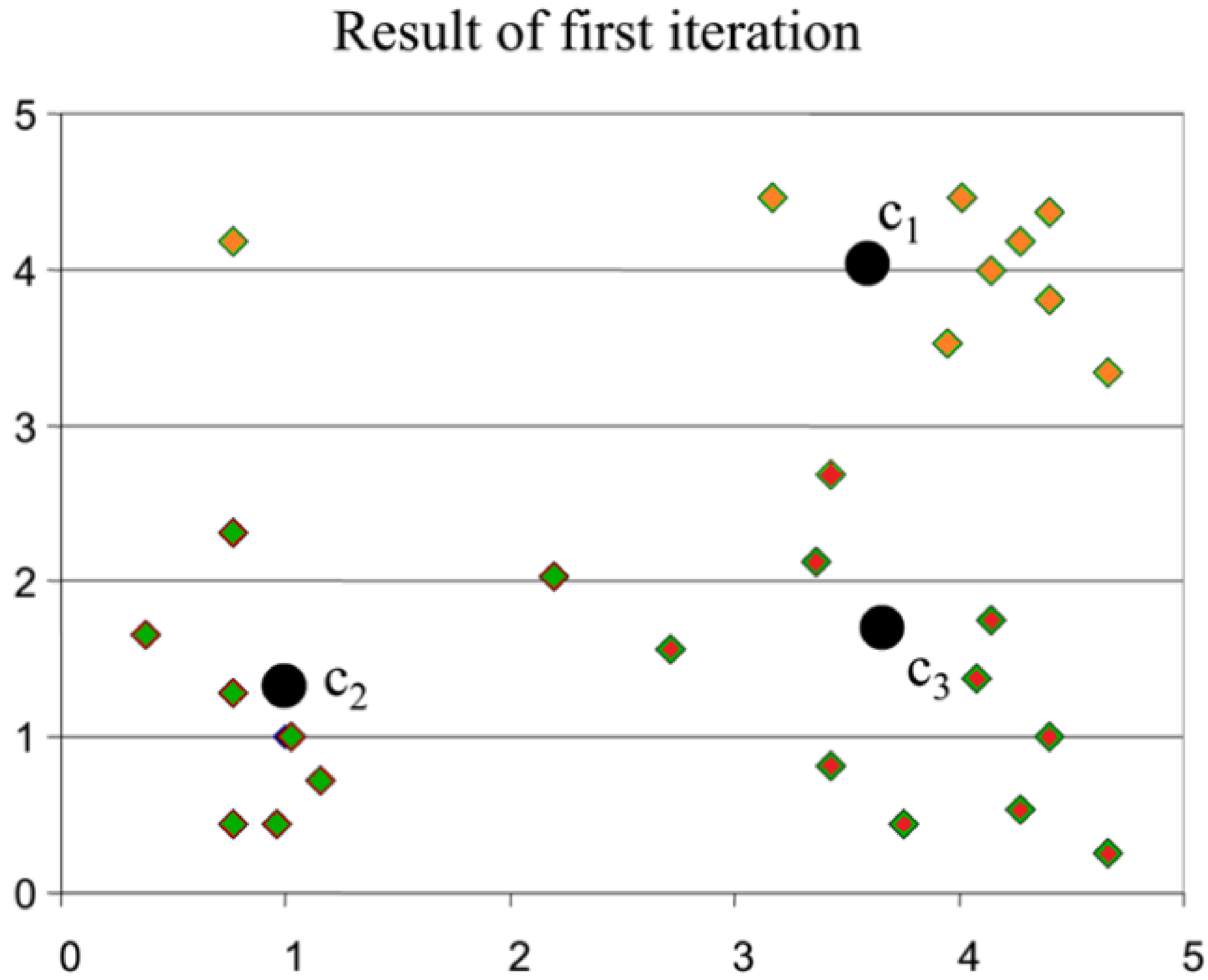
K-means clustering example: step 2



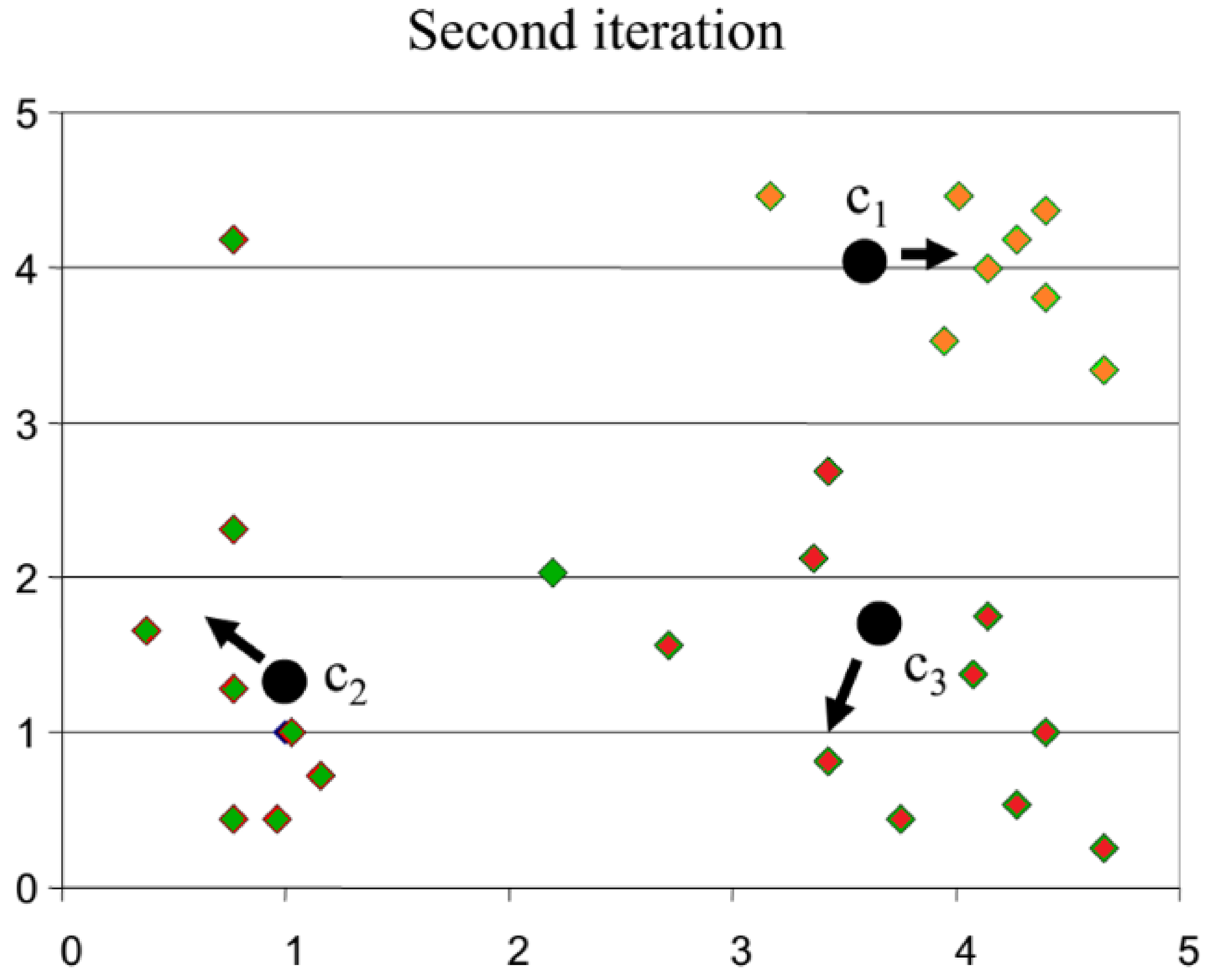
K-means clustering example: step 3



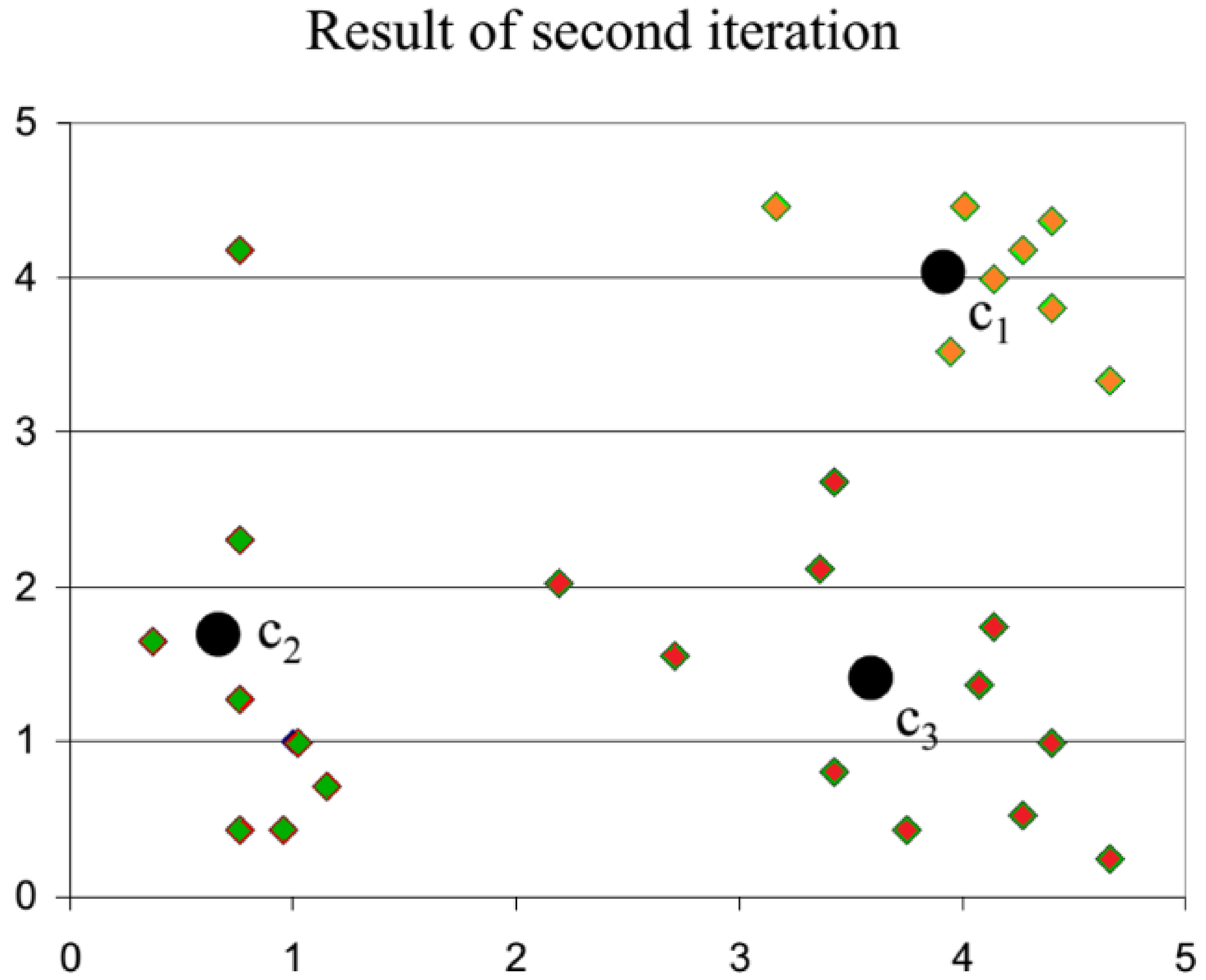
K-means clustering example



K-means clustering example



K-means clustering example



Why use K-means?

- Strengths:
  - Simple: easy to understand and to implement
  - Efficient: Time complexity: $O(tkn)$,

    where n is the number of data points,

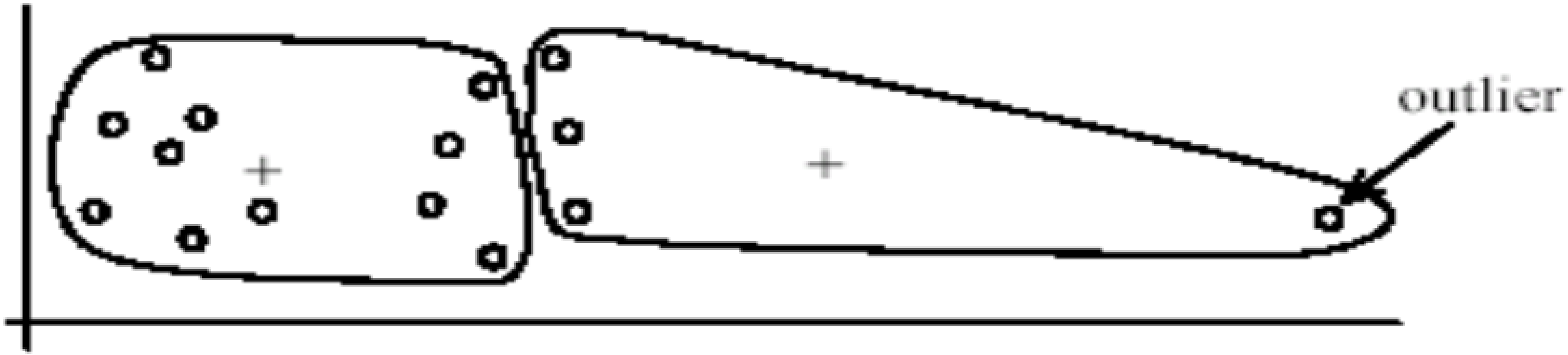
    k is the number of clusters, and

    t is the number of iterations.
  - Since both k and t are small. k-means is considered a linear algorithm.
- K-means is the most popular clustering algorithm.
- Note that: it terminates at a local optimum if SSE is used. The global optimum is hard to find due to complexity.

Weaknesses of K-means

- The algorithm is only applicable if the mean is defined.
  - For categorical data, *k*-mode - the centroid is represented by most frequent values.
- The user needs to specify *k*.
- The algorithm is sensitive to **outliers**
  - Outliers are data points that are very far away from other data points.
  - Outliers could be errors in the data recording or some special data points with very different values.

Outliers



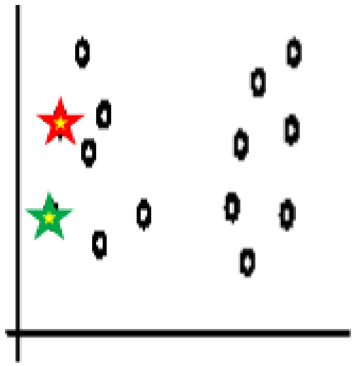
(A): Undesirable clusters

(B): Ideal clusters

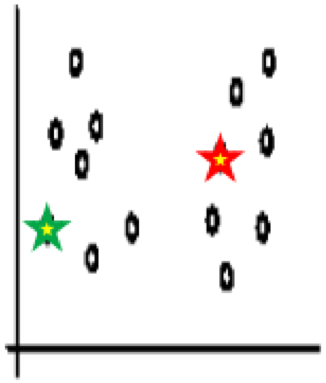
Dealing with outliers

- Remove some data points that are much further away from the centroids than other data points
  - To be safe, we may want to monitor these possible outliers over a few iterations and then decide to remove them.
- Perform random sampling: by choosing a small subset of the data points, the chance of selecting an outlier is much smaller
  - Assign the rest of the data points to the clusters by distance or similarity comparison, or classification

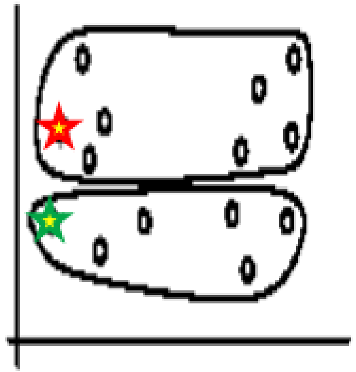
Sensitivity to initial seeds



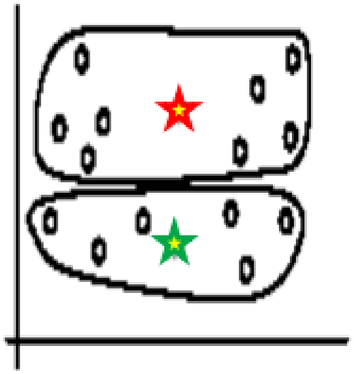
Random selection of seeds (centroids)



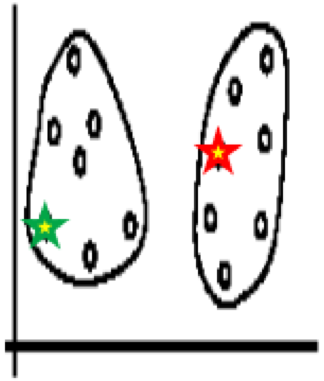
Random selection of seeds (centroids)



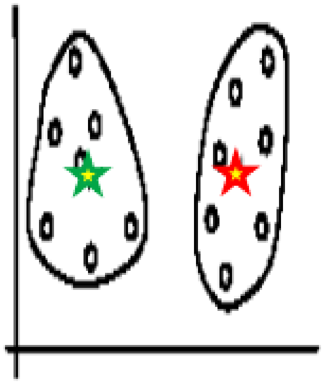
Iteration 1



Iteration 2



Iteration 1

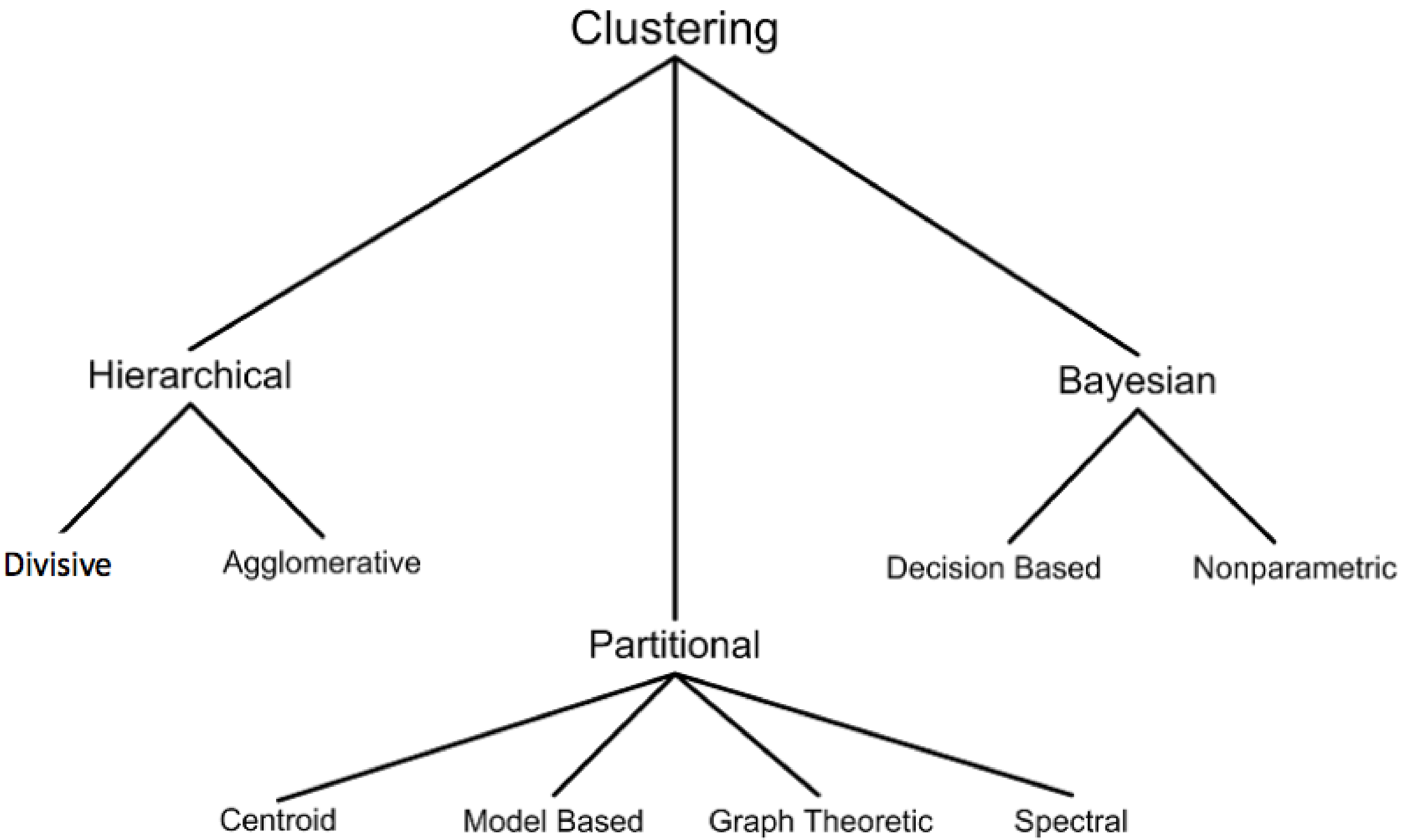


Iteration 2

K-means summary

- Despite weaknesses, *k*-means is still the most popular algorithm due to its simplicity and efficiency
- No clear evidence that any other clustering algorithm performs better in general
- Comparing different clustering algorithms is a difficult task. No one knows the correct clusters!

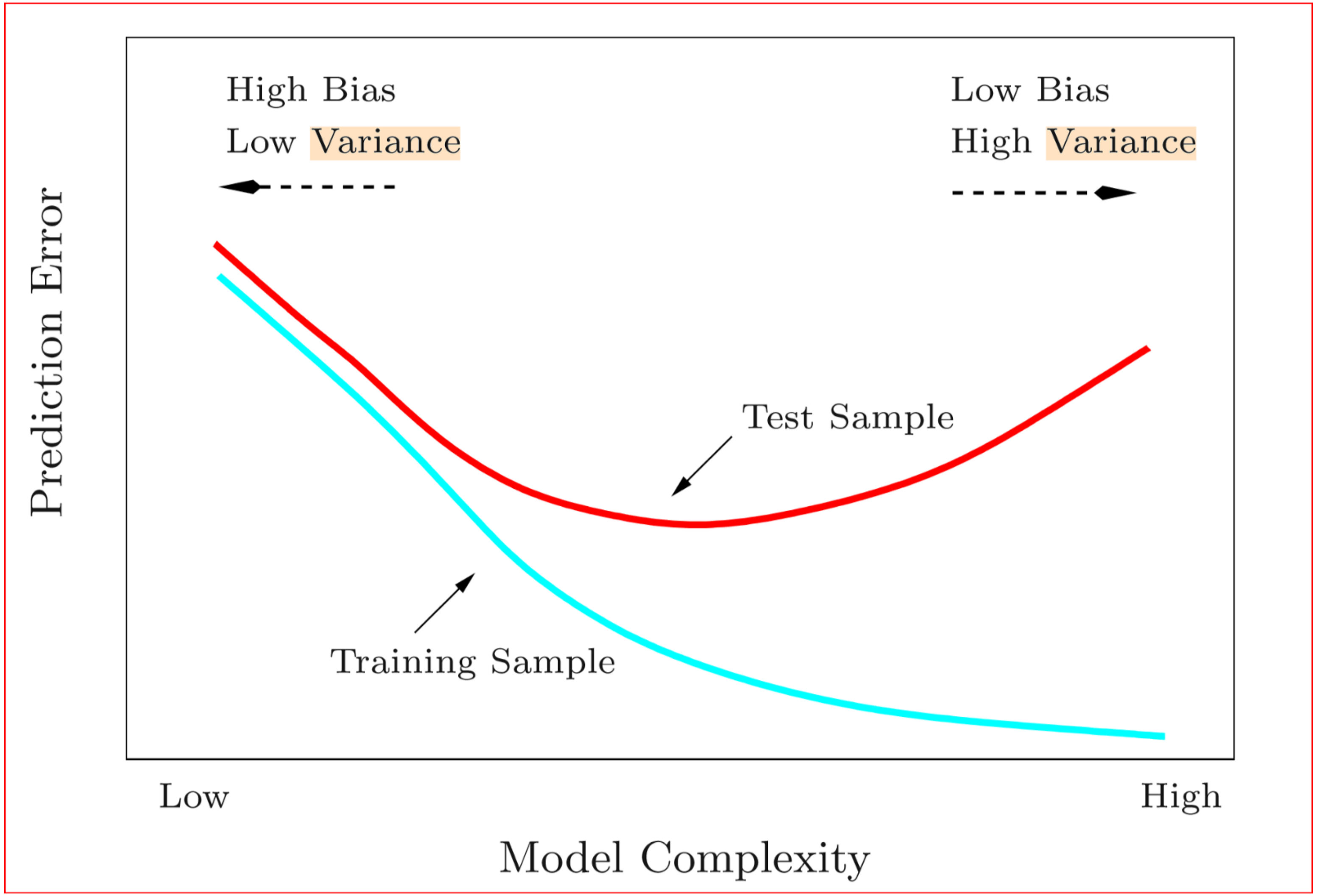
Tons of clustering techniques



Summary: clustering

- Clustering has a long history and still is in active research
  - There are a huge number of clustering algorithms, among them: Density based algorithm, Sub-space clustering, Scale-up methods, Neural networks based methods, Fuzzy clustering, Co-clustering …
  - More are still coming every year
- Clustering is hard to evaluate, but very useful in practice
- Clustering is highly application dependent (and to some extent subjective)
- Competitive learning in neuronal networks performs clustering analysis of the input data

Ensemble learning: Fighting the Bias/Variance Tradeoff



Ensemble methods

- Combine different models together to

- Minimize variance
  - Bagging
  - Random Forests

- Minimize bias
  - Functional Gradient Descent
  - Boosting
  - Ensemble Selection

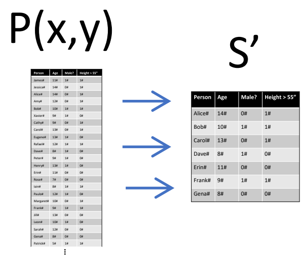
Ensemble methods

- Combine different models together to

- Minimize variance
  - Bagging
  - Random Forests

- Minimize bias
  - Functional Gradient Descent
  - Boosting
  - Ensemble Selection

Bagging



- **Goal:** reduce variance

sampled independently

- **Ideal setting:** many training sets S'
  - Train model using each S'
  - Average predictions

Variance reduces linearly
Bias unchanged

$$\boxed{\underbrace{E_S[(h(x|S) - y)^2]}_{\text{Expected Error}} = E_S[(Z-\check{z})^2] + \check{z}^2}$$

Variance **Bias**

$Z = h(x|S) - y$

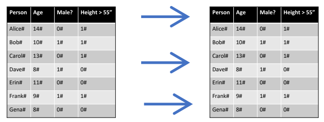
$\check{z} = E_S[Z]$

"Bagging Predictors" [Leo Breiman, 1994]

http://statistics.berkeley.edu/sites/default/files/tech-reports/421.pdf

Bagging

S S’



from S

- **Goal:** reduce variance
- **In practice:** resample S’ with replacement
  - Train model using each S’
  - Average predictions

Variance reduces sub-linearly
(Because S’ are correlated)
Bias often increases slightly

$$\boxed{E_S[(h(x|S) - y)^2] = E_S[(Z-\check{z})^2] + \check{z}^2}$$

Expected Error — Variance — Bias

$$Z = h(x|S) - y$$
$$\check{z} = E_S[Z]$$

Bagging = Bootstrap Aggregation

“Bagging Predictors” [Leo Breiman, 1994]
http://statistics.berkeley.edu/sites/default/files/tech-reports/421.pdf

Random Forests

- **Goal:** reduce variance
  - Bagging can only do so much
  - Resampling training data asymptotes

- **Random Forests:** sample data & features!
  - Sample S'
  - Train DT
    - At each node, sample features (sqrt)
  - Average predictions

Further de-correlates trees

"Random Forests – Random Features" [Leo Breiman, 1997]
http://oz.berkeley.edu/~breiman/random-forests.pdf

Ensemble methods

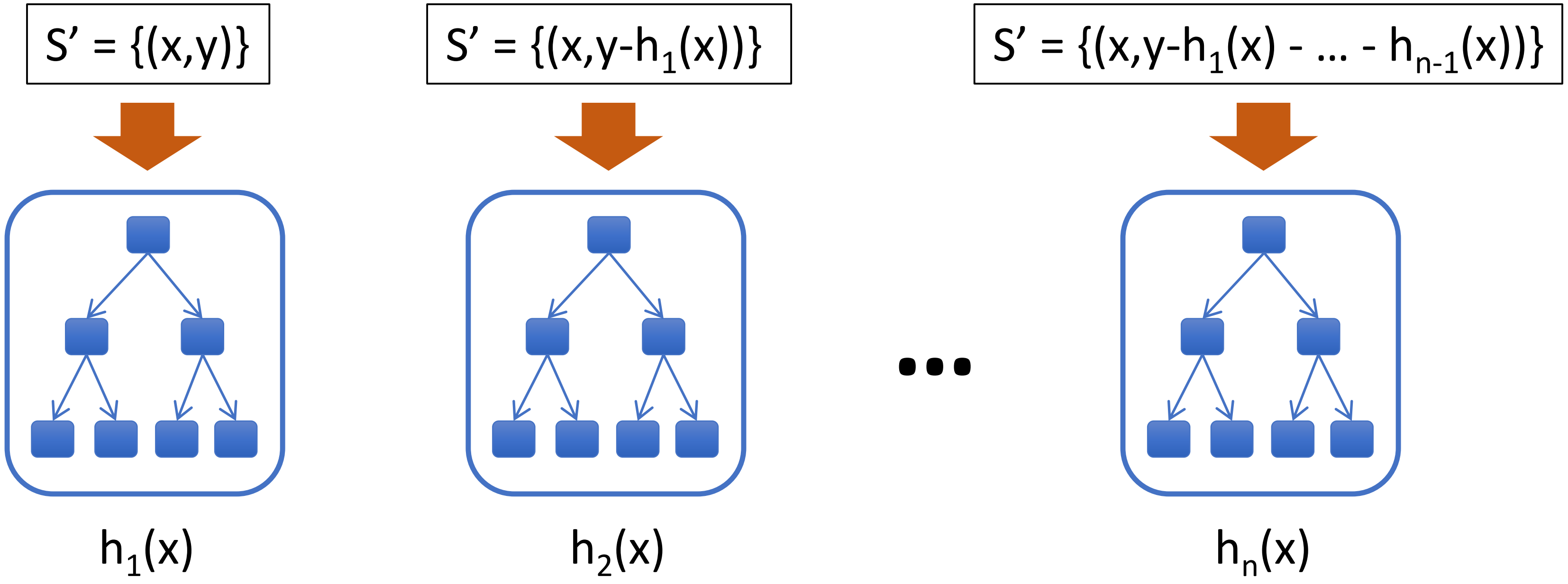
- Combine different models together to

- Minimize variance
  - Bagging
  - Random Forests

- Minimize bias
  - Functional Gradient Descent
  - Boosting
  - Ensemble Selection

Gradient Boosting

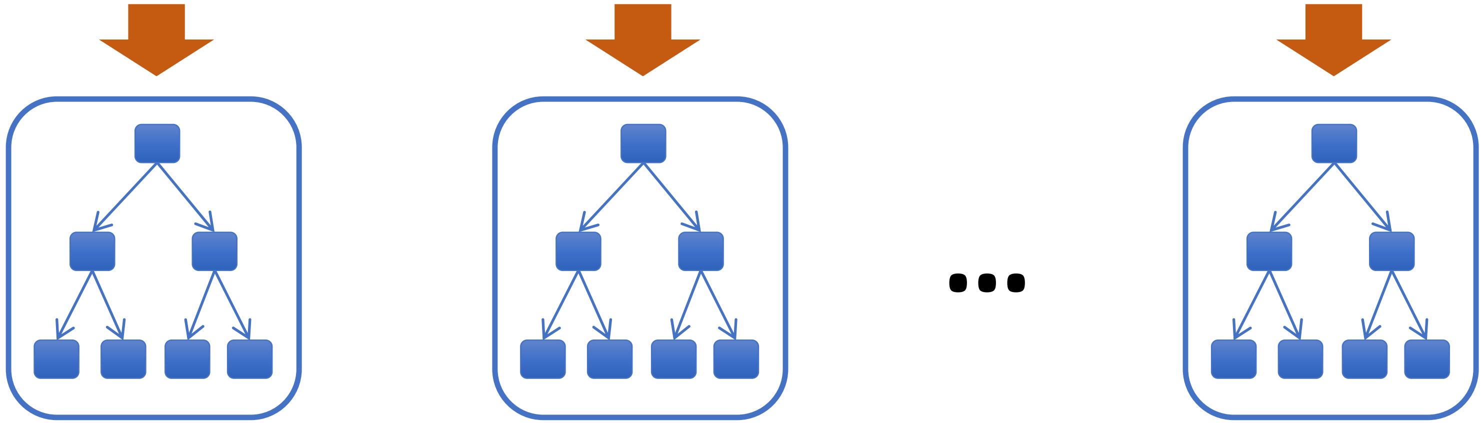
$$h(x) = h_1(x) + h_2(x) + \dots + h_n(x)$$



Boosting (AdaBoost)

$$h(x) = a_1h_1(x) + a_2h_2(x) + \ldots + a_3h_n(x)$$

$S' = \{(x,y,u_1)\}$ $S' = \{(x,y,u_2)\}$ $S' = \{(x,y,u_3))\}$



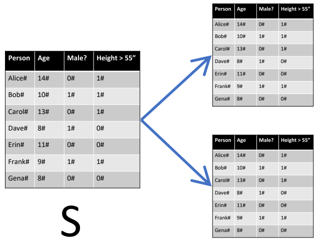
$h_1(x)$ $h_2(x)$ … $h_n(x)$

u – weighting on data points
a – weight of linear combination

https://www.cs.princeton.edu/~schapire/papers/explaining-adaboost.pdf

Stop when validation performance plateaus (will discuss later)

Ensemble Selection



S

Training S’

Validation V’

H = {2000 models trained using S’}

Maintain ensemble model as combination of H:

$h(x) = h_1(x) + h_2(x) + \dots + h_n(x) + h_{n+1}(x)$

Denote as h_{n+1}

Add model from H that maximizes performance on V’

Repeat

Models are trained on S’
Ensemble built to optimize V’

“Ensemble Selection from Libraries of Models”
Caruana, Niculescu-Mizil, Crew & Ksikes, ICML 2004

Summary

Method	Minimize Bias?	Minimize Variance?	Other Comments
Bagging	Complex model class. (Deep DTs)	Bootstrap aggregation (resampling training data)	Does not work for simple models.
Random Forests	Complex model class. (Deep DTs)	Bootstrap aggregation + bootstrapping features	Only for decision trees.
Gradient Boosting (AdaBoost)	Optimize training performance.	Simple model class. (Shallow DTs)	Determines which model to add at run-time.
Ensemble Selection	Optimize validation performance	Optimize validation performance	Pre-specified dictionary of models learned on training set.