Statistical Methods in Artificial Intelligence
CSE471 - Monsoon 2015 : Lecture 22

Avinash Sharma

CVIT, IIIT Hyderabad

Lecture Plan

- Revision from Previous Lecture
- Practical Aspects of Spectral Clustering
- Relation to Kernel Kmeans
- Decision Trees
- Classification and Regression Trees (CART)
- C4.5 and ID3 versions of Decision Trees
- Graphical Models (Next Class)

Un-normalized Spectral Clustering

- Input: Laplacian matrix ($\mathbf{L}$) and the number (k) of clusters to compute.
- Output: Cluster $C_1, \dots, C_k$

1. Compute eigen-decomposition of $\mathbf{L}$ matrix : $\mathbf{L} = \mathbf{U\Lambda U}^{\boldsymbol{T}}$
   where, $\mathbf{U} = [\boldsymbol{u}_1, \dots, \boldsymbol{u}_n]$ and $\boldsymbol{\Lambda} = \mathrm{diag}(\,[\lambda_1, \dots, \lambda_\mathrm{n}]\,)$

2. Define k-dimensional graph embedding using the eigenvectors associated with k smallest eigenvalues: $\mathbf{Y} = [\boldsymbol{u}_1, \dots, \boldsymbol{u}_{\boldsymbol{k}}]^T$

3. Cluster the columns y_i for $i = 1, \dots, n$ into k clusters using the K-means clustering algorithm.

Related but not discussed here

- Random walk Laplacian
  - Transition Matrix
  - Commute Time Distance
  - Page Rank Algorithm (Google Search)

- Manifold Learning
  - Dimensionality Reduction Techniques
  - Non-linear Kernel Extensions
  - Relationship to Geometric Laplacian

- Spectral Clustering Variants
  - Constraint Spectral Clustering

- Nodal Domain and Sets of Eigenvectors

Practical Aspects

- What is the best graph construction ?
  - Choice of graph induction
  - Choice of weights (similarity function)
- Choice of Laplacian
  - Un-normalized (Combinatorial)
  - Normalized
- Eigen-decomposition
  - Scalability
  - Accuracy
- How to decide k ?
  - Eigengap analysis
  - Choose different k for embedding size and k-means

Relation to Kernel Kmeans

- Minimize $J = \sum_{i=1}^{N} \sum_{j=1}^{K} a_{ij} \left\| \varphi(\mathbf{x}_i) - \widetilde{\boldsymbol{\mu}_j} \right\|^2$

  such that $a_{ij} \epsilon \{0,1\}$ and $\sum_{j=1}^{K} a_{ij} = 1$ $\widetilde{\boldsymbol{\mu}_j} = \frac{\sum_{i=1}^{N} a_{ij} \varphi(\mathbf{x}_i)}{\sum_{i=1}^{N} a_{ij}}$

- We can rewrite the criterion function as:

  Minimize $J \approx$ Maximize trace(AGA^T)

  where, G is an $N \times N$ kernel matrix and A is the optimal normalized cluster membership matrix

- Both Spectral Clustering & Kernel Kmeans apply kmeans clustering on the eigenvectors of Kernel matrix G.

Please refer to notes uploaded in the resource folder for detail derivation of criterion function.

Decision Trees

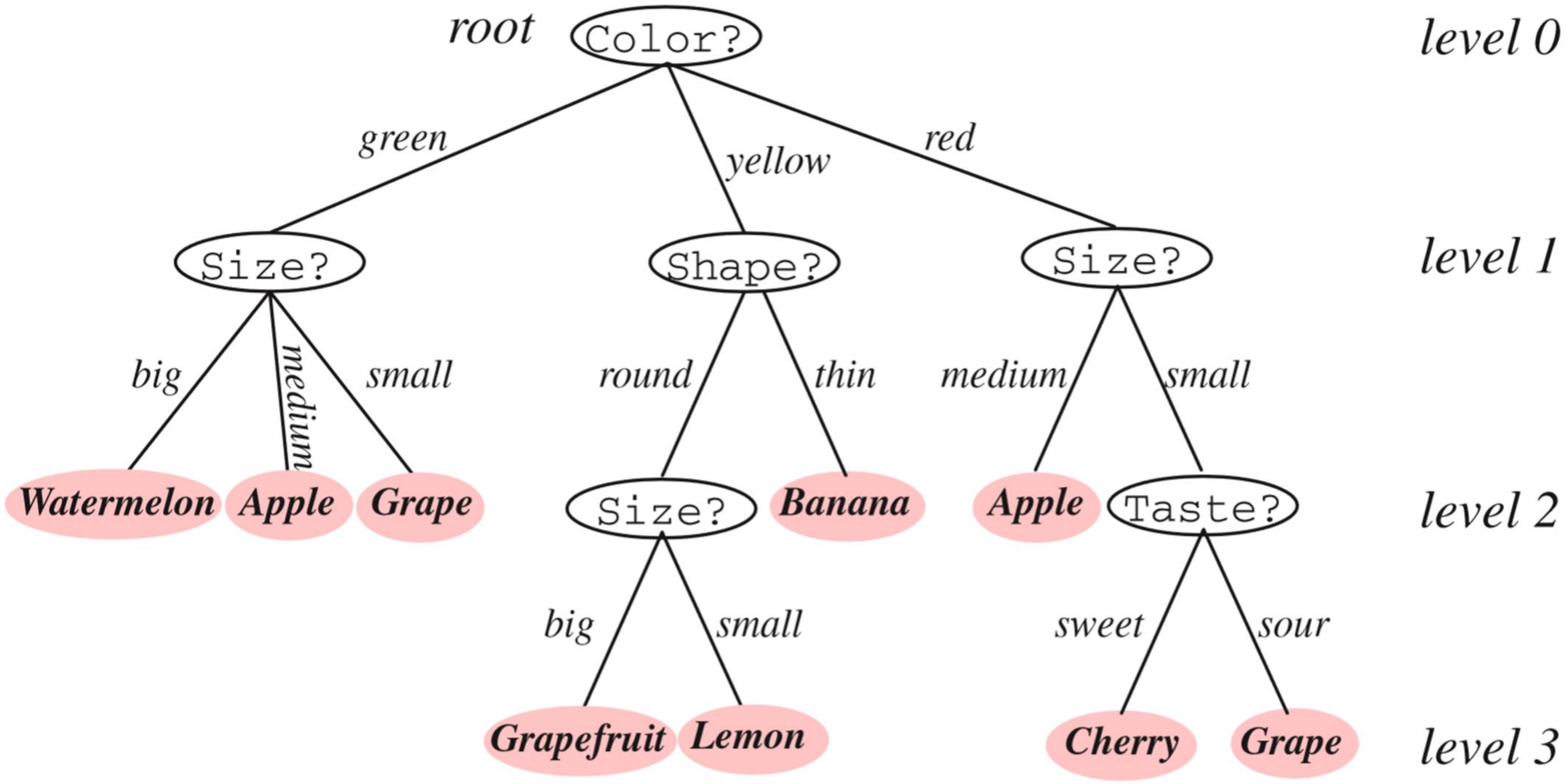
- An *inductive learning* task
  - Use particular facts to make more generalized conclusions

- A predictive model based on a branching series of Boolean tests
  - These smaller Boolean tests are less complex than a one-stage classifier

- Decision trees with discrete and finite values of target variable are called **classification trees**.

- Decision trees where the target variable can take continuous values (typically real numbers) are called **regression trees.**

Decision Trees



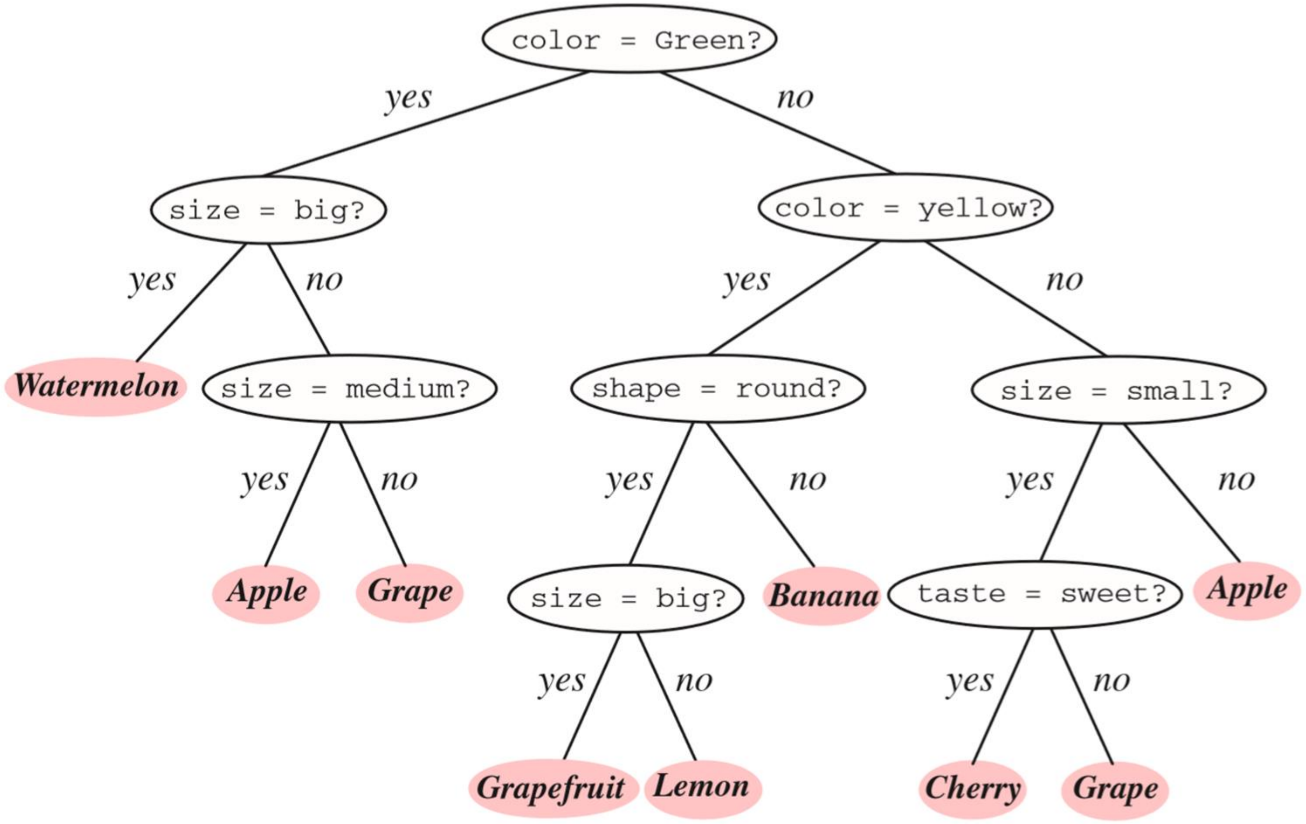
Decision Trees

- Decision tree acts as a tree structured classifier
  - Decision node: specifies a test on a single attribute
  - Leaf node: indicates the value of the target variable
  - Arc/edge: split of one attribute

- Decision trees are constructed in recursive manner by performing a split at every non-leaf node.
- An instances is classified by starting at the root of the decision tree and moving through it until a leaf node.

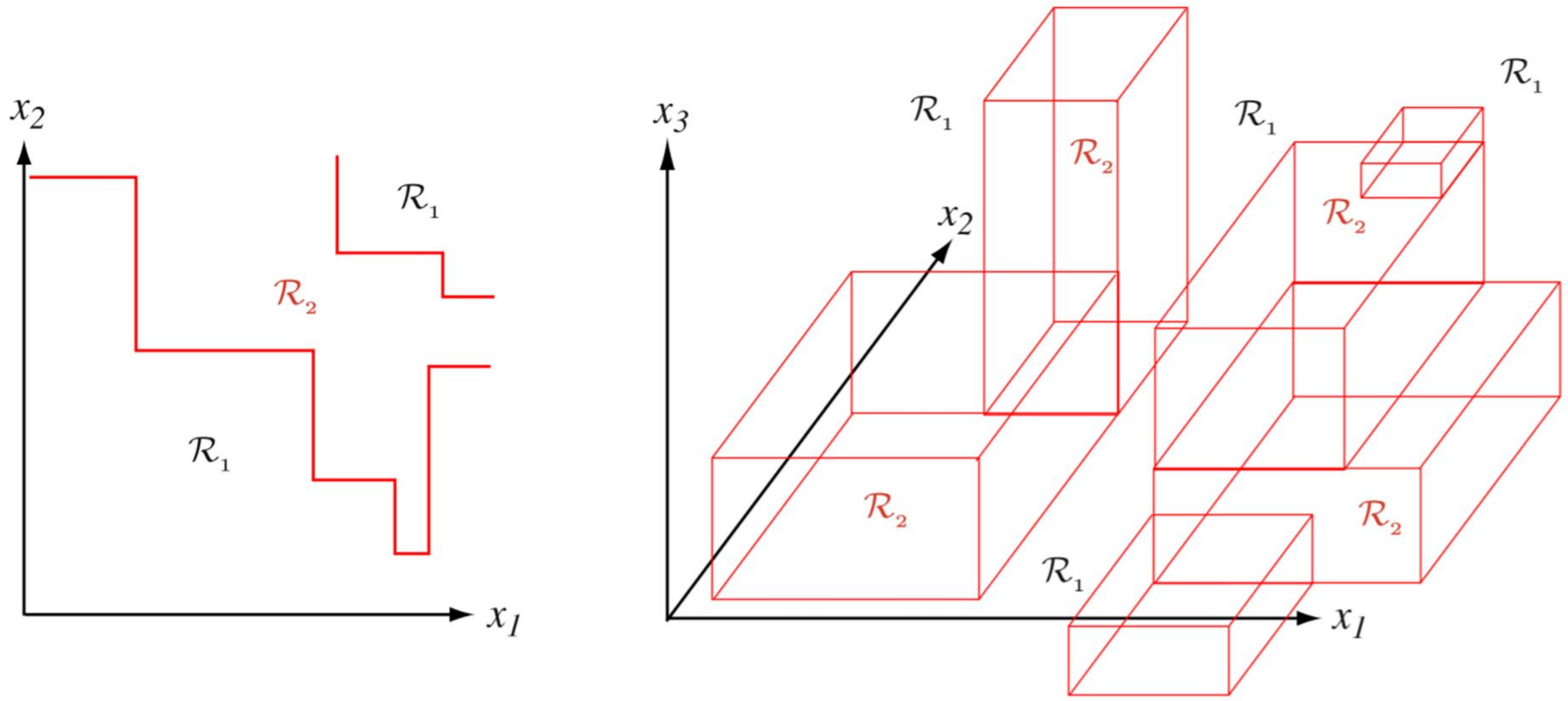
Classification and Regression Trees (CART)

- Optimal *Splits* (*Branching Factor)* at each node?
  - Every decision tree has a functionally equivalent binary tree.
  - Split is designer's choice.



Classification and Regression Trees (CART)

- A decision tree can learn arbitrary decision boundaries.

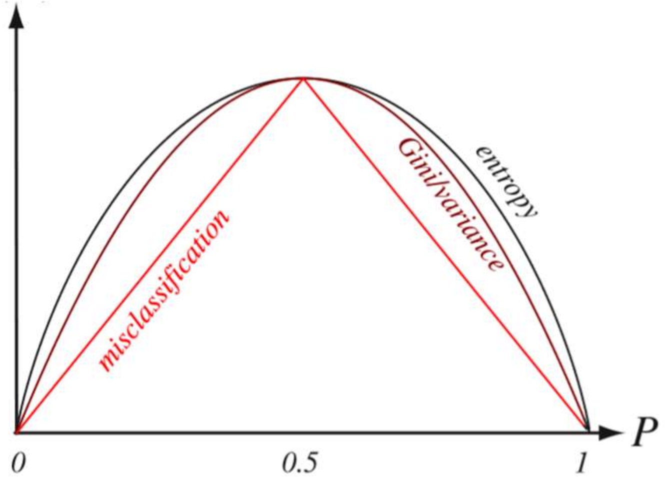


Classification and Regression Trees (CART)

- Which attribute should be tested at a node ?

or Which attribute should be used for splitting?

  - An attribute which helps to construct a simpler/compact tree with fewer nodes (supports Occam's razor).
  - An attribute which yields higher purity (or lower impurity) in descendent nodes.
  - Entropy Impurity : $-\sum_j P(\omega_j) \log_2 P(\omega_j),$
    - Entropy becomes 0 when all the patterns at node is having same class label.
    - Entropy becomes 1 when all the patterns at nodes are equi-probable from either class.
  - Gini Impurity: $\sum_{i \neq j} P(\omega_i)P(\omega_j) = 1 - \sum_j P^2(\omega_j).$
  - Misclassification Impurity: $1 - \max_j P(\omega_j),$



Classification and Regression Trees (CART)

- Which attribute should be tested at a node ?

or Which attribute should be used for splitting?

  - Choose an attribute splitting which maximizes the decrease in impurity as much as possible (or maximize the information gain)

$$\Delta i(N) = i(N) - P_L i(N_L) - (1 - P_L) i(N_R),$$

  - This is an local optimization done at each non-leaf node with greedy method and does not insure a final solution which minimizes overall decrease in impurity.
  - In case of multiple classes, twoing criterion can be used to define purity as 1 for a subset of classes instead of single class.
  - There is some tradeoff in the classification performance due to choice of impurity measure but the overall empirical performance is less affected by this choice.

Classification and Regression Trees (CART)

- Which attribute should be tested at a node ?

or Which attribute should be used for splitting?

  - Multi-way split to decrease the impurity favors large *B*.

$$\Delta i(s) = i(N) - \sum_{k=1}^{B} P_k i(N_k),$$

  - Another option is to measure the **gain ratio impurity**

$$\Delta i_B(s) = \frac{\Delta i(s)}{-\sum_{k=1}^{B} P_k \log_2 P_k}.$$

Classification and Regression Trees (CART)

- When to stop splitting ?
  - If the information gain is less than some threshold then we can stop.
  - This strategy yields an unbalanced tree with leaf nodes at different levels
  - Finding a best threshold is difficult.
  - Another option is to stop when a leaf node has some fix % of overall data points/patterns.
  - We can minimize the global criterion, which penalizes the large graphs (size) and overall impurity at all leaf nodes :

$$\alpha \cdot size + \sum_{leaf\ nodes} i(N),$$

  - In case of entropy impurity measure, this is similar to finding a minimum description length for every leaf node.
  - Hypothesis Testing: how far is a candidate split from a random split using the statistical measures like ***chi-squared statistics*** (provides confidence level)

$$\chi^2 = \sum_{i=1}^{2} \frac{(n_{iL} - n_{ie})^2}{n_{ie}},$$

Classification and Regression Trees (CART)

- Pruning in Decision Trees
  - It is an inverse to splitting operation.
  - An early stop to splitting is called Horizon Effect.
  - A better option is to create an overfitting tree and then start merging leaf nodes or collapsing subtrees to a single leaf nodes as long as it does not affect the global information gain.
  - This option is more computationally expensive option though.
  - A rule based simplification can also be seen as pruning but it doesn’t necessarily suggest a generalization of the tree.
  - Pruning provides improved interpretability.

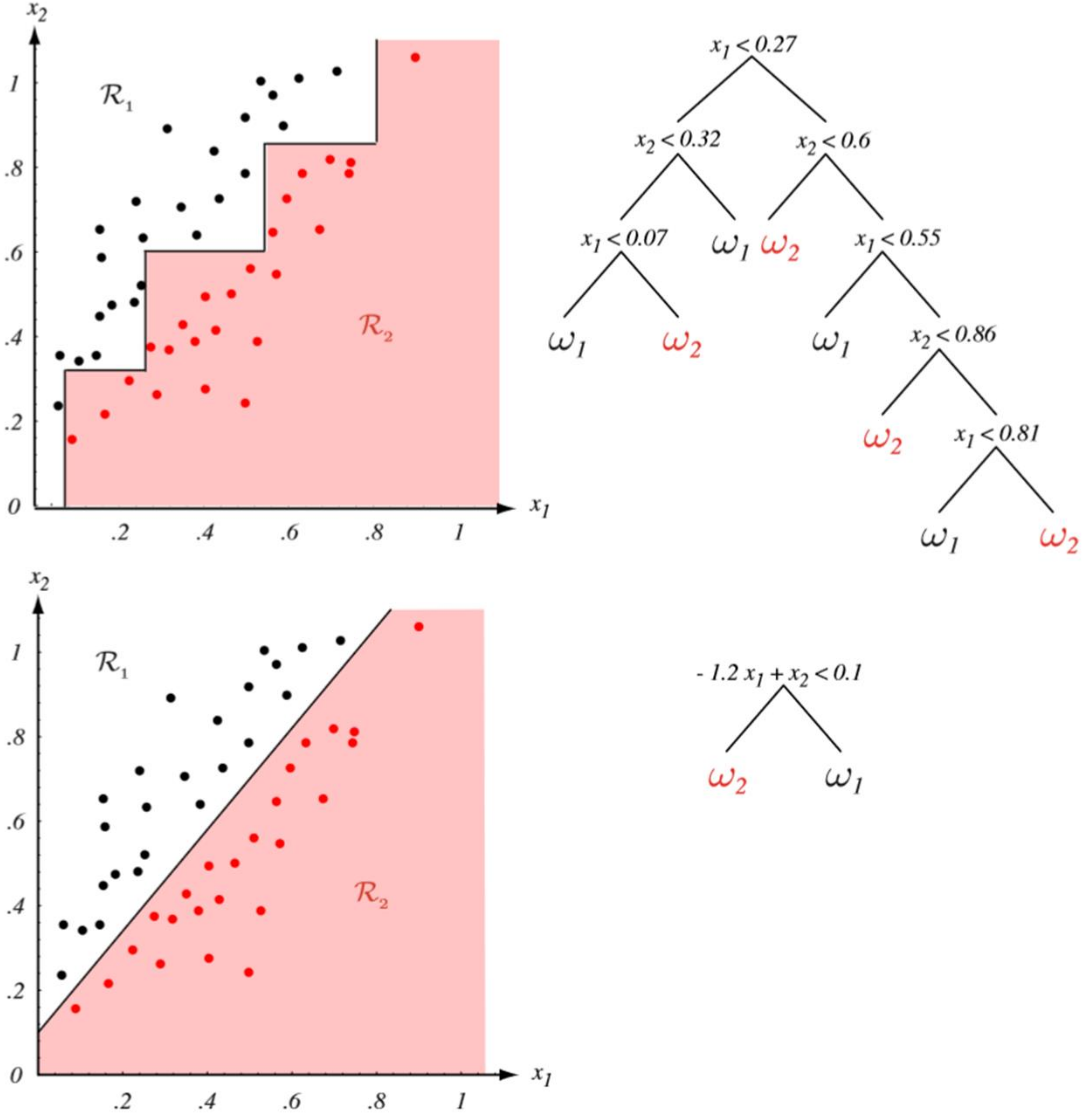
Classification and Regression Trees (CART)

- How to assign a class label to an impure Leaf node?
  - A simple majority voting strategy is sufficient.

- Computational Complexity of tree creation is $O(dn\ (\log n)^2)$ and classification is $O(\log n)$, i.e., the levels in the tree.

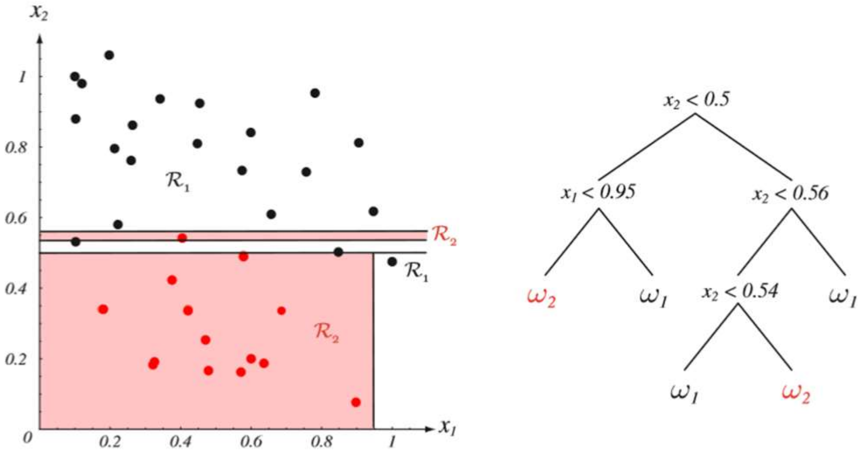
- Feature Choice: A PCA in feature space might be a better choice for attributes instead of considering original features as it can yield simpler trees.

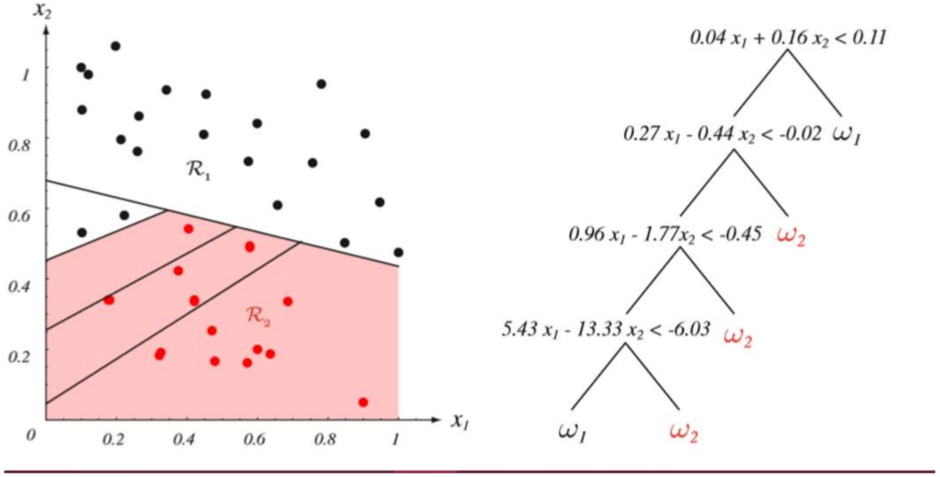
Classification and Regression Trees (CART)



Classification and Regression Trees (CART)

- Multivariate Decision Trees
  - If the natural split is not parallel to original axes or if the training and test data distributions are significantly different, the resulting decision tree might have poor generalization even after pruning.
  - Idea is to use a linear function of multiple attributes (e.g, LMS) instead of a single attribute.



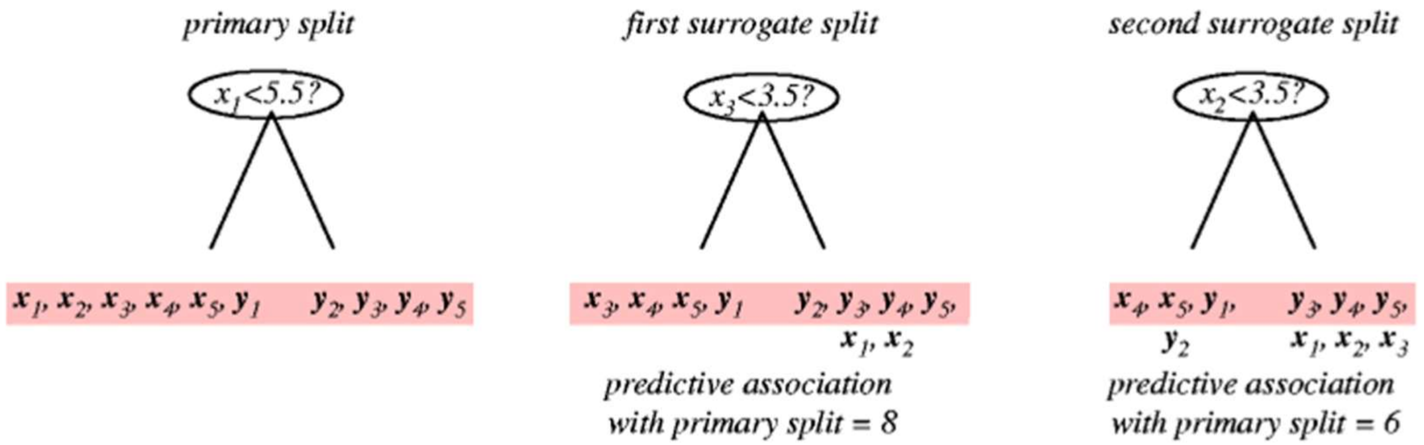

x_2
1
0.8
0.6
0.4
0.2
0
0.2
0.4
0.6
0.8
1
x_1
$\mathcal{R}_1$
$\mathcal{R}_2$
$0.04\, x_1 + 0.16\, x_2 < 0.11$
$0.27\, x_1 - 0.44\, x_2 < -0.02$
ω_1
$0.96\, x_1 - 1.77 x_2 < -0.45$
ω_2
$5.43\, x_1 - 13.33\, x_2 < -6.03$
ω_2
ω_1
ω_2

Classification and Regression Trees (CART)

- How to handle Class Imbalance in data ?
  - Use Class priors and decision costs in the impurity formulation.

$$i(N) = \sum_{ij} \lambda_{ij} P(\omega_i) P(\omega_j),$$

- How to handle Missing attributes ?
  - Use the idea of defining surrogate splits at every node. Each surrogate split maximize the predictive association with the primary split.


primary split
$x_1<5.5?$
$x_1, x_2, x_3, x_4, x_5, y_1$
y_2, y_3, y_4, y_5
first surrogate split
$x_3<3.5?$
x_3, x_4, x_5, y_1
$y_2, y_3, y_4, y_5, x_1, x_2$
predictive association with primary split = 8
second surrogate split
$x_2<3.5?$
x_4, x_5, y_1, y_2
$y_3, y_4, y_5, x_1, x_2, x_3$
predictive association with primary split = 6

Advantages of Decision Trees

- Simple to understand and interpret.
- Requires little data preparation.
- Able to handle both numerical and categorical data.
- Possible to validate a model using statistical tests. That makes it possible to account for the reliability of the model.
- Robust to variation in data.
- Performs well when trained with large datasets.

Limitations of Decision Trees

- The problem of learning an optimal decision tree is known to be NP-complete. Thus, greedy algorithm based solutiong only make locally-optimal decisions at each node. Such algorithms cannot guarantee to return the globally-optimal decision tree.

- Decision-tree learners can create over-complex trees that do not generalize well from the training data (overfitting).

- For data including categorical variables with different numbers of levels, information gain in decision trees is biased in favor of those attributes with more levels.

Variants of Decision Tree

- ID3
  - Trees with higher branching factor and one level per attribute.
  - Real values are binned and each node has branching factor equal to number of bins per attribute.
  - Stops when each leaf node is pure.
- C4.5
  - Similar to CART for real value attributes but similar to ID3 for categorical attributes
  - No surrogate splits used for missing attributes. Instead all probable leaf nodes are returned and decision is made based on probability of each class label.

- Extended classifiers using Decision Trees
  - Bagging
  - Random Forest
  - Rotation Forest