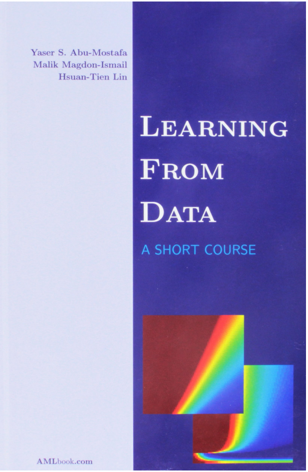
MAC 0459 / 5865

Data Science and Engineering

R. Hirata Jr. (hirata@ime.usp.br)

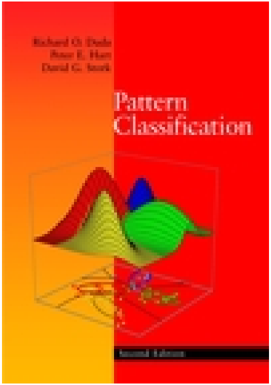
Class 19 (2020)

Bibliography



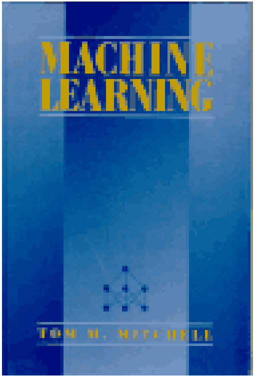
Learning From Data
by Yaser S. Abu-Mostafa, Malik Magdon-Ismail and Hsuan-Tien Lin

Bibliography



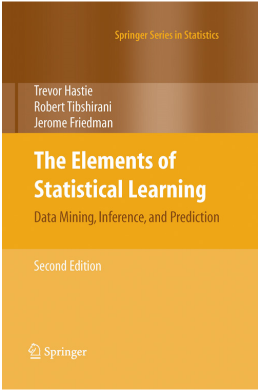
Pattern Classification
by Richard O. Duda, Peter E. Hart and David G. Stork
2nd ed., 2000

Bibliography



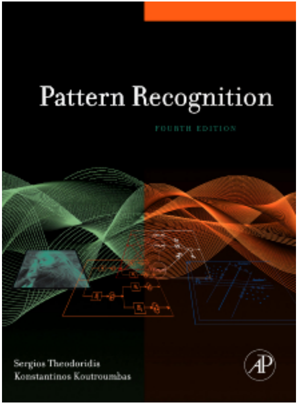
Machine Learning, Tom Mitchell, McGraw Hill, 1997

Bibliography



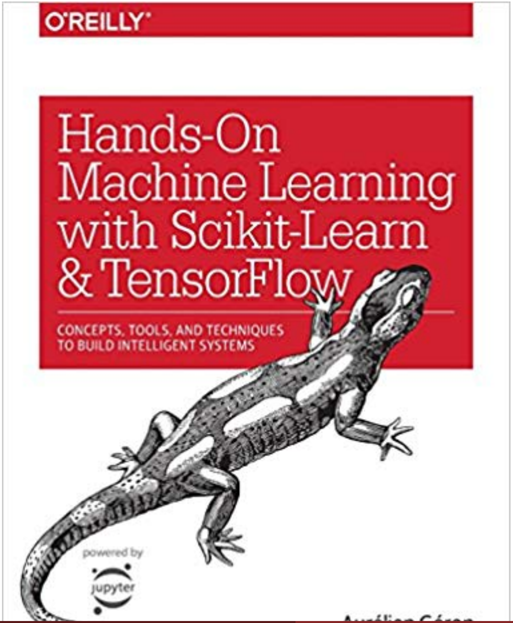
The Elements of Statistical Learning: Data Mining, Inference, and Prediction
Trevor Hastie, Robert Tibshirani, Jerome Friedman
Second Edition, 2009

Bibliography



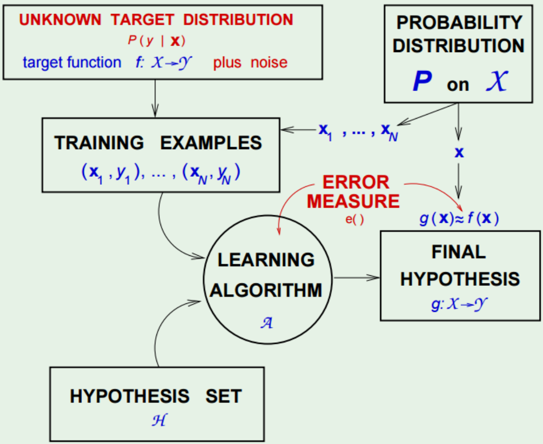
Pattern Recognition
Sergios Theodoridis, Konstantinos Koutroumbas
4th Edition, Academic Press, 2008

Bibliography



Stating the problem

The learning diagram - including noisy target



Decision Trees

(Decision Trees / Classification trees)

Example: Akinator

http://en.akinator.com/

- **Let S be a set of objects in $\mathbb{R}^d$ and $i \in \{1, 2, \ldots, d\}$.**
  Consider, for example, the subsets:

$$S_1 = \{x \in S : x_i \leq 0\}$$

$$S_2 = \{x \in S : x_i > 0\}$$

The question "$x_i \leq 0$?" admits of two answers, Yes or No, and partitions the set S in two subsets.
It partitions the space of features in two subspaces.

Decision Trees

- **Each of these subsets can be recursively partitioned adding other questions.**

  Depending on the question, the space can be partitioned in more than two subspaces.

- **The sequence of questions and possible answers can be organized in a tree structure.**

  Questions are associated to the nodes of the tree and the number of possible answers define the number of branches of a node.

- **This idea is used by decision tree classifiers.**

Decision Trees Classifier

- The induction (training) fase starts by associating a set S of items (the whole space) to the root node.

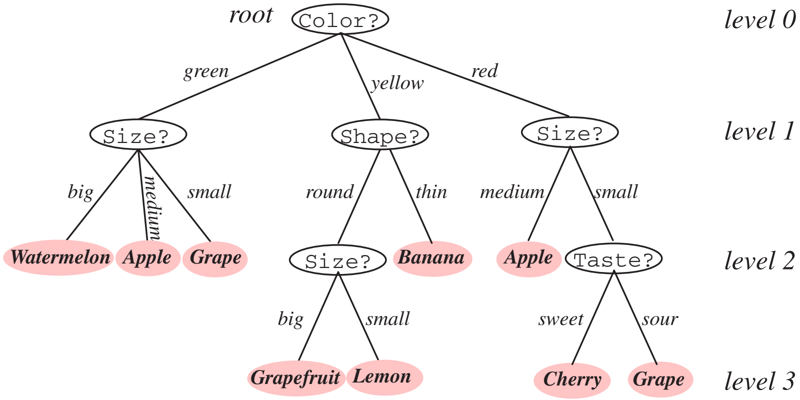
  The sets associated to the nodes are recursively partitioned if they are not pure (a node is pure if there is only one class of items in it). This partition is done using satisfactory questions.

  Leaf nodes correspond to pure subspaces.

- Given a decision tree, to classify a new item, one has to traverse the tree from the root to a leaf node, always following the branch corresponding to the correct answer to the question of the visited node.

Example of a decision tree

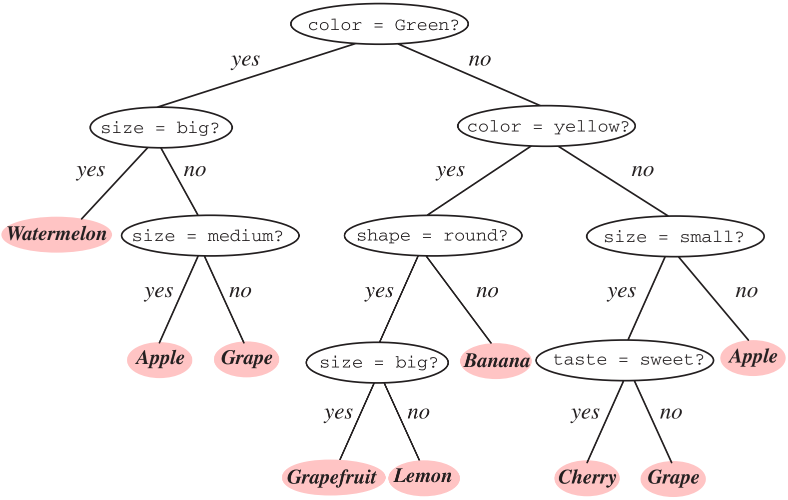
Classification of fruit based on color, size, form and taste



Source: Duda et al

Decision trees

Any decision tree can be expressed by a binary tree.



Inducing a decision tree

Growing a decision tree

- How do we choose a question to be associated to a certain node?
- When do we stop splitting a subspace? What is a pure subspace?

Most algorithms consider binary decision trees (because of is expressivity and simplicity in the training)

Decision trees

Possible questions

- **Categoric features**: Just consider $x_i = v$?, where v is a possible value assumed by x_i.

  This question has to have a binary answer

- **Numeric features**: sort the values of x_i and, for each two values v_k e v_{k+1} of x_i such that v_k and v_{k+1} are from different classes, consider the question $x_i \leq \frac{v_k + v_{k+1}}{2}$?

  (the number of possible questions can be HUGE ...)

- We can consider a **linear combination of some features**. In this case, the hyperplane is not neccessarily orthogonal to any feature.

Decision trees: how to decide which question to use

- **Choose simple questions that can induce simple and compact trees**
- **Choose questions such that their answers can generate subspaces that has less mixture of classes in each subspace.**
- **To do that, one defines purity measures, that are computed for each node**
- **To split a set of examples, the algorithm computes an impurity measure for each possible division and choose the one that minimizes the lower impurity**

Decision trees: minimizing the impurity

- Let N be a node of the tree. The **impurity index** of the subset associated to this node is denoted by $i(N)$

- Let N_L and N_R be the **child nodes** that would be created if we split N.

  Let P_L (P_R) be **the proportion of items** in N that would go to node N_L (N_R)

- The **purity reduction** (or **information gain**) of this splitting can be computed by

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

Decision trees: non binary splittings

If a node N, associated to a set S of items is splitted into subsets S_v (where v is an index), associated to the child nodes N_v. The **information gain** of this division is given by:

$$Gain(N) = i(N) - \sum_v \frac{|S_v|}{|S|} \, i(N_v)$$

However: the larger the number of subsets S_v, the larger the gain!

Decision trees: non binary splittings

If a node N, associated to a set S of items is splitted into subsets S_v (where v is an index), associated to the child nodes N_v. The **information gain** of this division is given by:

$$Gain(N) = i(N) - \sum_v \frac{|S_v|}{|S|}\, i(N_v)$$

However: the larger the number of subsets S_v, the larger the gain!

It is better to consider something as:

$$GainRatio(N) = \frac{Gain(N)}{SplitInfo(N)}$$

where

$$SplitInfo(N) = -\sum_v \frac{|S_v|}{|S|} \log \frac{|S_v|}{|S|}$$

Decision trees: measures of impurity

Entropy impurity:

$$i(N) = -\sum_{i=1}^{c} P(\omega_i|N) log_2 P(\omega_i|N)$$

Gini impurity:

$$i(N) = \sum_{i \neq j} P(\omega_i|N)P(\omega_j|N) = \frac{1}{2}\Big[1 - \sum_{j} P^2(\omega_j|N)\Big]$$

Misclassification impurity:

$$i(N) = 1 - max_j P(\omega_j|N)$$

Decision trees: measures of impurity

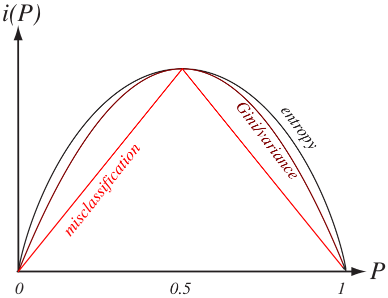


FIGURE 8.4. For the two-category case, the impurity functions peak at equal class frequencies and the variance and the Gini impurity functions are identical. The entropy, variance, Gini, and misclassification impurities (given by Eqs. 1–4, respectively) have been adjusted in scale and offset to facilitate comparison here; such scale and offset do not directly affect learning or classification. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

The impurity $i(N)$ is maximum when the elements of each class are equally presented in N

Decision trees: measures of impurity

The choice of questions that minimize the impurity can result in a division such that the same class may be splitted in two different subsets.

Twoing criteria

twoing: tries to choose a splitting such that the elements of the same class between the c classes do not fall in different subsets.

Algorithm: Let $C_1 = \{\omega_{i_1}, \ldots, \omega_{i_k}\}$ and $C_2 = C \setminus C_1$ where C denotes the set of all classes

Consider all possible dicotomies C_1/C_2

Compute one of the impurity measures considering that the classes are C_1 and C_2

Decision trees: when to stop splitting

- when the note is **pure**.
  - Problem 1: Usual to have leaves with only one sample.
  - Problem 2: *overfitting*.

Decision trees: when to stop splitting

- when the note is **pure**.
  - Problem 1: Usual to have leaves with only one sample.
  - Problem 2: *overfitting*.

- when the **impurity decreasing is not significant**
  Advantage of having a unique criteria for all nodes.
  Difficult to quantify when the decreasing is not significant.

Decision trees: when to stop splitting

- when the note is **pure**.
  - Problem 1: Usual to have leaves with only one sample.
  - Problem 2: *overfitting*.

- when the **impurity decreasing is not significant**
  Advantage of having a unique criteria for all nodes.
  Difficult to quantify when the decreasing is not significant.

- when the **number of samples** associated to a node is small (absolute value or percentage).
  Similarity with next neighbors (small volume for dense regions and large volumes to sparse regions).

Decision trees: when to stop splitting

- Use **cross validation** (uses part of the training data to evaluate the error; if the splitting results in a smaller validation error, split; otherwise, stop)
  Disadvantage: part of the sample has to be used to validation.

Decision trees: when to stop splitting

- Use **cross validation** (uses part of the training data to evaluate the error; if the splitting results in a smaller validation error, split; otherwise, stop)
  Disadvantage: part of the sample has to be used to validation.
- Use some **global criterium** (for instance, take in consideration all nodes).

Decision trees: when to stop splitting

- Use **cross validation** (uses part of the training data to evaluate the error; if the splitting results in a smaller validation error, split; otherwise, stop)
  Disadvantage: part of the sample has to be used to validation.

- Use some **global criterium** (for instance, take in consideration all nodes).

- etc