Statistical Machine Learning from Data

Decision Trees

Samy Bengio

IDIAP Research Institute, Martigny, Switzerland, and
Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland
bengio@idiap.ch
http://www.idiap.ch/~bengio

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

January 25, 2006

1. Introduction
2. Training a Decision Tree
3. Controlling the Capacity of a Decision Tree

1. Introduction
2. Training a Decision Tree
3. Controlling the Capacity of a Decision Tree

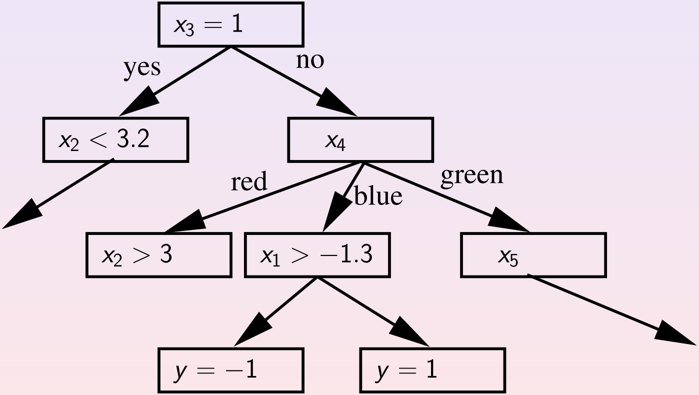
Rule Based Decisions

- Let D_n be a training set of n pairs $(\mathbf{x}^i, y^i)$
- Let y be a class (say -1 or 1)
- Let $\mathbf{x}$ be a vector of d attributes denoted $\{x_j | 1 \leq j \leq d\}$.
- A decision rule could be

$$\textit{if } x_3 \textit{ is true AND } x_5 \leq 3.2 \textit{ then } y = 1$$

- How to construct such a decision?
- This is the family of decision trees.

A Decision Tree



1 Introduction

2 Training a Decision Tree

3 Controlling the Capacity of a Decision Tree

The Most Informative Feature

- At each given node i of the tree, we have a subset D_i of training examples.
- We would like to select a feature j such that we divide D_i wisely according to the task.
- The objective: after the segmentation of D_i, all examples in a node should ideally be of the same class: this is called purity.
- There are several heuristics to lean towards purity.
- Let us first look at the case for discrete attributes.
- The choice made by most decision tree algorithms:

maximize the information gain

Information Gain

maximize the information gain

$$IG_j = H(Y) - H(Y|X_j)$$

where $H(Y)$ is the entropy of random variable Y
(the average number of bits needed to transmit Y)

$$H(Y) = -\sum_k P(Y=k) \log P(Y=k)$$

and $H(Y|X)$ is the conditional entropy...

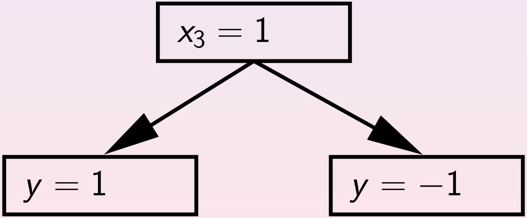
$$H(Y|X_j) = -\sum_m P(X_j = m) \sum_k P(Y=k|X_j=m) \log P(Y=k|X_j=m)$$

We thus select

$$j^* = \arg\max_j IG_j$$

Decision Stumps

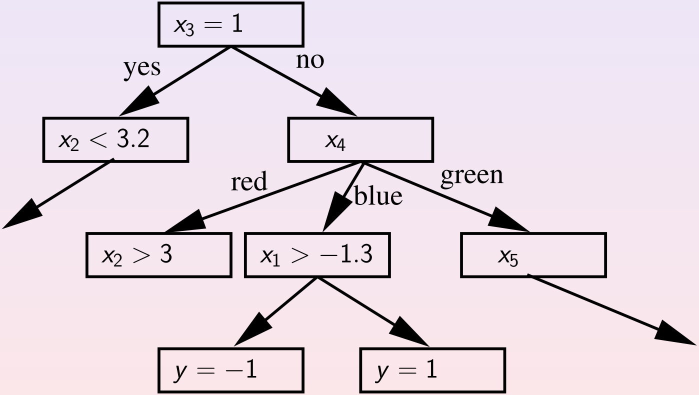
- A Stump is simply a one-level decision tree
- Hence, you select THE feature which maximizes the information gain in your whole data set



Recursion Step

- Take the original dataset
- Partition it according to the value of the attribute we split on
- For each partition, create a decision stump
- And do that recursively... until there is no more example to split
- You obtain a decision tree!

A Decision Tree



Continuous Input Variables

- What to do with non-discrete input variables?
- Instead of partitionning according to $X_j = k$
- Find the value of $X_j = t$ such that

$$H(Y|X_{j,t}) = \max_t H(Y|X_j < t)P(X_j < t) + H(Y|X_j \geq t)P(X_j \geq t)$$

- Compute the information gain as usual:

$$IG_j = H(Y) - H(Y|X_{j,t})$$

- if IG_j is the maximum among features, Partition X_j according to $X_j < t$ or not.

1 Introduction

2 Training a Decision Tree

3 Controlling the Capacity of a Decision Tree

Controlling the Capacity

- If you build a complete tree, it will learn by heart all your training set
- Hence, capacity is virtually infinite
- We need to create trees with controlled capacity
- There are various ways to do this:
  - Don't build the complete tree but stop before
  - Build the complete tree, and then prune it

... according to some validation set or prior information...