- Optimise cost at every split: <u>greedily.</u>

- Continue until a stopping criterion is met:

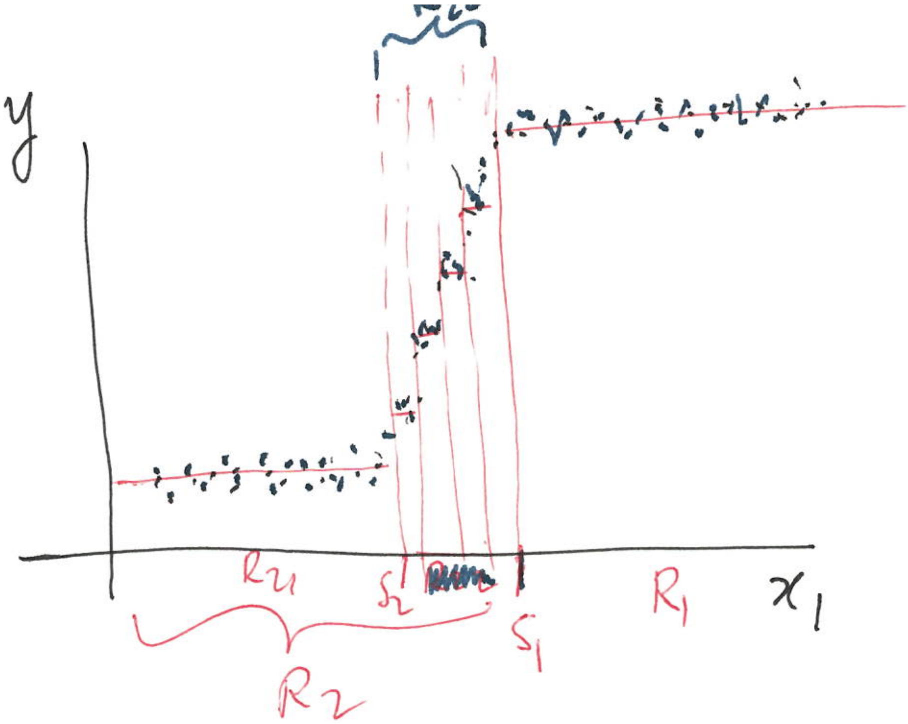
e.g.

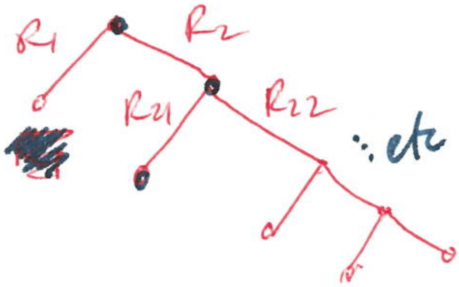
  - Plateau in optimisation
  - small # of points in regions
  - using all predictors.

$$\boxed{\begin{array}{l} \text{Given } \vec{x}^{in} \\ \quad \text{Find } R_{\vec{\alpha}} \;/\; \vec{x}^{in} \in R_{\vec{\alpha}} \\ \qquad \hat{y}^{DT} = \bar{y}_{R_{\vec{\alpha}}} \end{array}}$$

Mean of the region where $\vec{x}^{in}$ falls.

(TBC...)



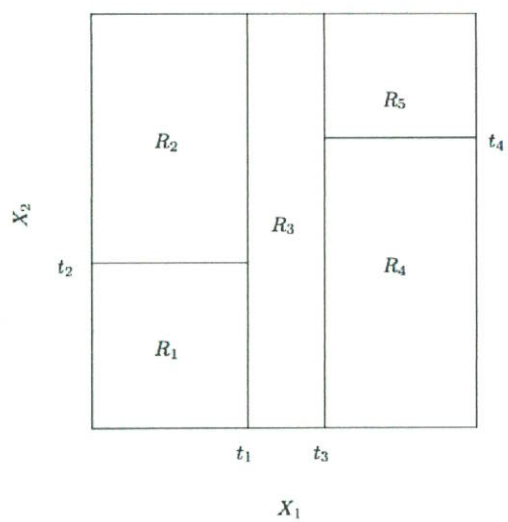

Example of how a DT would proceed with the case of one descriptor

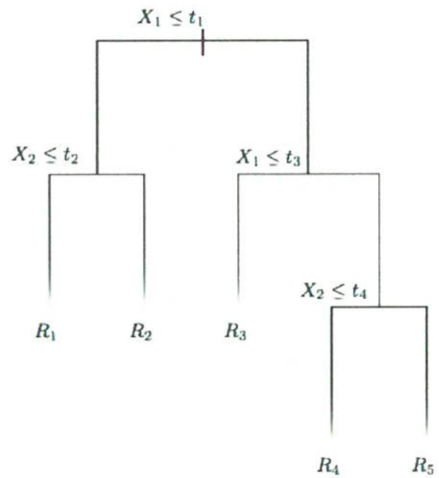
$$\left\{ \begin{array}{c} \vec{x} = x_1 \in \mathbb{R} \\ \text{and} \\ y \in \mathbb{R} \end{array} \right\}$$

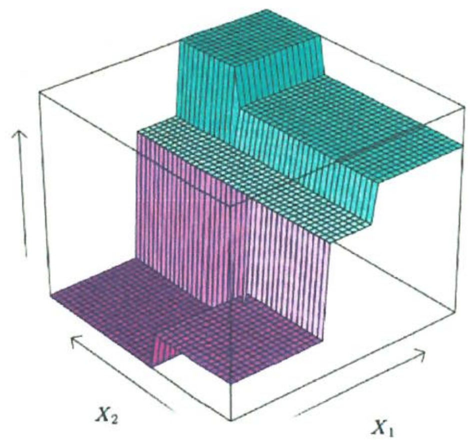
# Expressiveness of Decision Trees

We've seen that classification trees approximate boundaries in the feature space that separate classes.

Regression trees, on the other hand, define ***simple functions*** or step functions, functions that are defined on partitions of the feature space and are constant over each part.

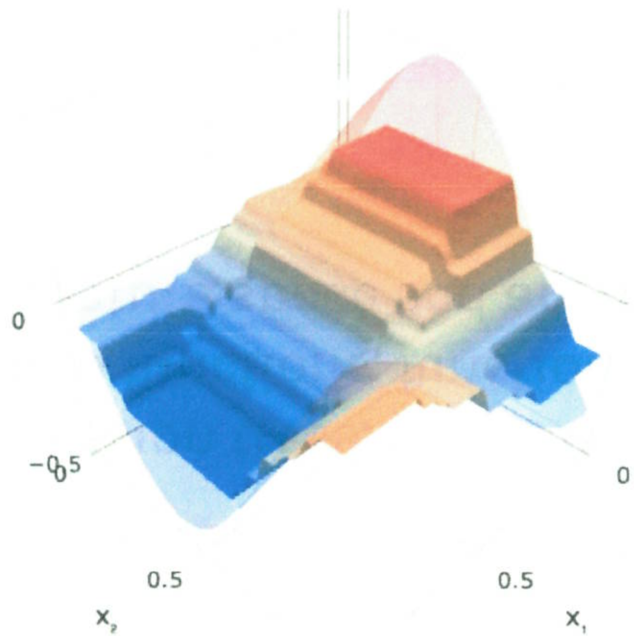


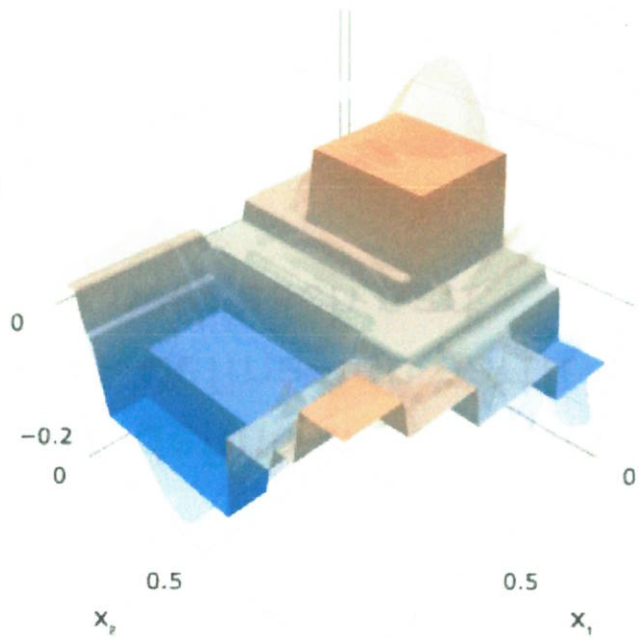



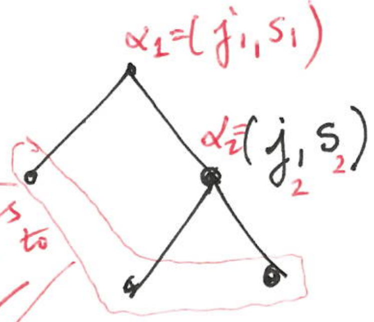
# Expressiveness of Decision Trees

For a fine enough partition of the feature space, these functions can approximate complex non-linear functions.



# Decision trees for classification:




leaves correspond to regions

Same as above:
Each split denoted by $(j,s)=\alpha_i$

For region $R_{\vec{\alpha}}$ :

$$\left[\pi(R_{\vec{\alpha}})\right]_q = \frac{\sum_{i=1}^{N} I\left(\vec{x}^{(i)} \in R_{\vec{\alpha}} \;\&\; y^{(i)} \in C_q\right)}{\sum_{i=1}^{N} I\left(\vec{x}^{(i)} \in R_{\vec{\alpha}}\right)}$$

$q = 1, \dots, Q$

$$\underset{Q\times 1}{\vec{\pi}(R_{\vec{\alpha}})} = \begin{bmatrix} \vdots \\ \pi(R_{\vec{\alpha}})_q \\ \vdots \end{bmatrix}$$

Probability vector of belonging to each class $q$ in $R_{\vec{\alpha}}$

Given $\vec{x}^{in}$ :

(1) Find $R_{\vec{\alpha}}$ s/ $\vec{x}^{in} \in R_{\vec{\alpha}}$

(2) Obtain $\vec{\pi}(R_{\vec{\alpha}})$

(3) $\hat{y} = \hat{f}^{DT}(\vec{x}^{in}) = \arg\max_{q} \vec{\pi}(R_{\vec{\alpha}}(\vec{x}^{in}))$

Important tweak:

What is the Cost function for the splits in the decision tree:

(1) ~~Contingency table: Minimise the error rate.~~ Not sensitive

(2) Gini index: or Cross-entropy — Cost function deals with how informative the split is:

Gini index of $\vec{\pi}(R_{\vec{\alpha}})$

$$GI[\vec{\pi}(R_{\vec{\alpha}})] = \sum_{q=1}^{Q} \pi(R_{\vec{\alpha}})_q \left(1 - (\pi(R_{\vec{\alpha}}))_q\right)$$

Cross entropy

$$CE[\vec{\pi}(R_{\vec{\alpha}})] = \sum_{q=1}^{Q} \pi(R_{\vec{\alpha}})_q \log(1 - \pi(R_{\vec{\alpha}})_q)$$

Two versions of information (or entropy)

Properties of GI:

$$i \in \{1, \dots, J\} \qquad \begin{bmatrix} p_1 \\ \vdots \\ p_J \end{bmatrix} = \vec{p}$$

$$GI(\vec{p}) = \sum_{i=1}^{J} p_i (1 - p_i) = 1 - \sum_{i=1}^{J} p_i^2$$

---

Lagrange multipliers

$$L = GI(\vec{p}) - \lambda \left( \sum_{i=1}^{J} p_i - 1 \right)$$

$$\begin{cases} \nabla_{\vec{p}} L = -2\vec{p} - \lambda \vec{1} \\ \dfrac{\partial L}{\partial \lambda} = \sum_{i=1}^{J} p_i - 1 \end{cases}$$

$$\vec{p}^* = \frac{-\lambda}{2} \vec{1} \qquad \underline{\underline{p_i^* = \frac{1}{J}}}$$

- The maximum GI is when all elements of a probability vector are <u>equal</u>

- If $\vec{p} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \Rightarrow GI(\vec{p}) = 0$

  It's the minimum.

Cost function is GI:
In every split, we minimize GI:
Find $(j, s)$ such that
$\min_{j,s} GI(\vec{\pi}(R_j))$