Let $S = ((x_1, y_1), ..., (x_n, y_n))$ be a labeled sample with x_i drawn from the D-dimensional feature space $\mathcal{X}$ and y_i from the label space $\mathcal{Y}$.

1 Decision trees

A decision partitions the feature space into different regions $(R_1, ..., R_{|T|})$, where $|T|$ is the number of leaves in the tree, and assigns a simple model (usually just a constant label) to each regions.

Each inner node in the tree is associated with one of the features $d \in \{1, ..., D\}$ and a threshold Θ. Starting with the root node, the training data S is recursively split into smaller and smaller binary partitions, depending on whether the feature x_d is larger or smaller than the threshold Θ. The final model of a given leaf are determined by the training points ending up in that leaf.

We grow the tree recursively by choosing d and Θ for first the root node, then its children, then their children and so forth, each choice of d and Θ maximizing the information gain on the training points $\tilde{S}$ being split at the given node:

$$G_{d,\Theta}(\tilde{S}) = Q(\tilde{S}) - \left(\frac{L_{d,\Theta}}{\tilde{S}} Q(L_{d,\Theta}) + \frac{R_{d,\Theta}}{\tilde{S}} Q(R_{d,\Theta}) \right) \tag{1}$$

That is, we have defined some impurity measure Q going from sets of training points to real numbers, and then we want to maximize how much more our splitted data is compared to our non-splitted data. Therefore, we substract the weigthed sum of the impurity of our two sets of splitted training points $L_{d,\Theta}$ and $R_{d,\Theta}$ with the impurity of our training points $\tilde{S}$ before the split at the given node.

We grow the tree, until each node is pure, or the number of training points ending up at the node is below a given threshold ϕ. After we have grown the tree this way, we then prune it to remove some of its complexity and avoid overfitting.

We can also avoid overfitting by using a random forest instead of a single decision tree. In random forests, we do not use pruning.

See the algorithm for the recursive growing of the tree in the lecture slides.

2 Regression trees

Let $\mathcal{X} = \mathbb{R}^D$ and $\mathcal{Y} = \mathbb{R}$.

Let the impurity measure Q of a set of training points S_η at some node η associated with some constant c_η be the squared loss

$$Q(S_\eta) = \frac{1}{|S_\eta|} \sum_{(x_n, y_n) \in S_\eta} (y_n - c_\eta)^2 \tag{2}$$

To maximize the information gain from a (d, Θ) split at the node, we have to choose the constants c_L and c_R and the split-parameters (d, Θ) such as to minimize

$$\sum_{(x_n, y_n) \in L_{d,\Theta}} (y_n - c_L)^2 + \sum_{(x_n, y_n) \in R_{d,\Theta}} (y_n - c_R)^2 \tag{3}$$

where $L_{d,\Theta}$ and $R_{d,\Theta}$ are the splitted sets resulting from the choice of d and Θ.

We can do this by setting

$$c_L = \frac{1}{|L_{d,\Theta}|} \sum_{(x_n, y_n) \in L_{d,\Theta}} y_n \tag{4}$$

and equivalently for c_R. That means we have to choose d and Θ to minimize

$$\sum_{(x_n, y_n) \in L_{d,\Theta}} (y_n - \frac{1}{|L_{d,\Theta}|} \sum_{(x_n, y_n) \in L_{d,\Theta}} y_n)^2 + \sum_{(x_n, y_n) \in R_{d,\Theta}} (y_n - \frac{1}{|R_{d,\Theta}|} \sum_{(x_n, y_n) \in R_{d,\Theta}} y_n)^2 \tag{5}$$

We can do this by sorting the training features x_d and then checking threshold values set to the midpoint between each of the consecutive pair of sorted values.

After we have grown the tree, we need to prune it to avoid overfitting, see the lecture slides.

The pruning involves a hyper parameter α, which needs to be set with the use of some kind of validation process.