NON LINEAR LEARNING ALGORITHMS

* **Support Vector Machines (SVM)**

SVM are based kernel methods require only a user-specified kernel function 𝐾(𝑥𝑖, 𝑥𝑗), i.e., a **similarity function** over pairs of data points (𝑥𝑖, 𝑥𝑗) into kernel (dual) space on which learning algorithms operate linearly, i.e. every operation on points is a linear combination of 𝐾(𝑥𝑖, 𝑥𝑗) .

Outline of the SVM algorithm:

1. Map points 𝑥 into kernel space using a kernel function: 𝑥 → 𝐾(𝑥, .).

2. Learning algorithms operate linearly by dot product into high-kernel space 𝐾(., 𝑥𝑖)\*𝐾(., 𝑥𝑗).

i) Using the kernel trick (Mercer’s Theorem) replace dot product in hgh dimensional space by a simpler operation such that 𝐾(., 𝑥𝑖) ・ 𝐾(., 𝑥𝑗) = 𝐾(𝑥𝑖, 𝑥𝑗). Thus we only need to compute a similarity measure for each pairs of point and store in a 𝑁 × 𝑁 Gram matrix.

ii) Finally, The learning process consist of estimating the 𝛼𝑖 of the decision function that maximises the hinge loss (of 𝑓(𝑥)) plus some penalty when applied on all training points.

**𝑓(𝑥) = sign (∑𝛼𝑖 𝑦𝑖 𝐾(𝑥𝑖, 𝑥))**

3. Predict a new point 𝑥 using the decision function.

* **Random Forest**

A random forest is a meta estimator that fits a number of **decision tree learners** on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting.

* **Decision Tree Learner**

A tree can be “learned” by splitting the training dataset into subsets based on an features value test.

Each internal node represents a “test” on an feature resulting on the split of the current sample. At each step the algorithm selects the feature and a cutoff value that maximises a given metric. Different metrics exist for regression tree (target is continuous) or classification tree (the target is qualitative).

This process is repeated on each derived subset in a recursive manner called **recursive partitioning**. The recursion is completed when the subset at a node has all the same value of the target variable, or when splitting no longer adds value to the predictions. This general principle is implemented by many recursive partitioning tree algorithms.

Decision trees are simple to understand and interpret however they tend to overfit the data. However decision trees tend to overfit the training set. Leo Breiman propose random forest to deal with this issue.



**Classification tree**