ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Introduction to Machine Learning Methods

and CARET package in R

CS-411 : Digital Education & Learning Analytics

Mina Shirvani Boroujeni
Nov. 01 2016

What is machine learning?

- A group of algorithms that can automate analytical model building.
- Parameters of the model are learnt from the data.
- Applicable in situations where it is very challenging to define rules by hand:
  - Face detection
  - Speech recognition
  - Stock prediction
  - ...

Supervised vs. Unsupervised learning

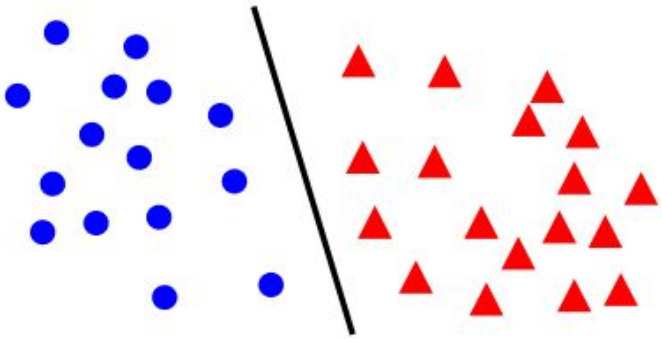
- **Supervised learning**:
  - Find relation between independent variable(s) X and the dependant variable Y
  - The value of dependant variable Y is known for the training data (labeled data)
  - Classification and Regression

- **Unsupervised learning**
  - Extract patterns/classes in the data without having the dependent variable (not-labeled data)
  - Clustering

Supervised learning: problem type

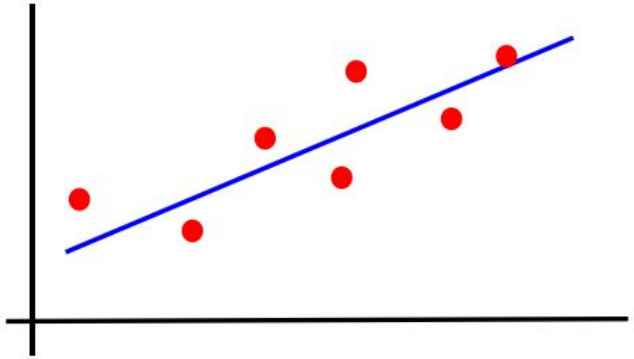
Classification

- Dependent variable is categorical
- **example**: estimate the price category of a car: *cheap* or *expensive*



Regression

- Dependent variable is continuous (or ordinal)
- **example**: estimate the price of a car



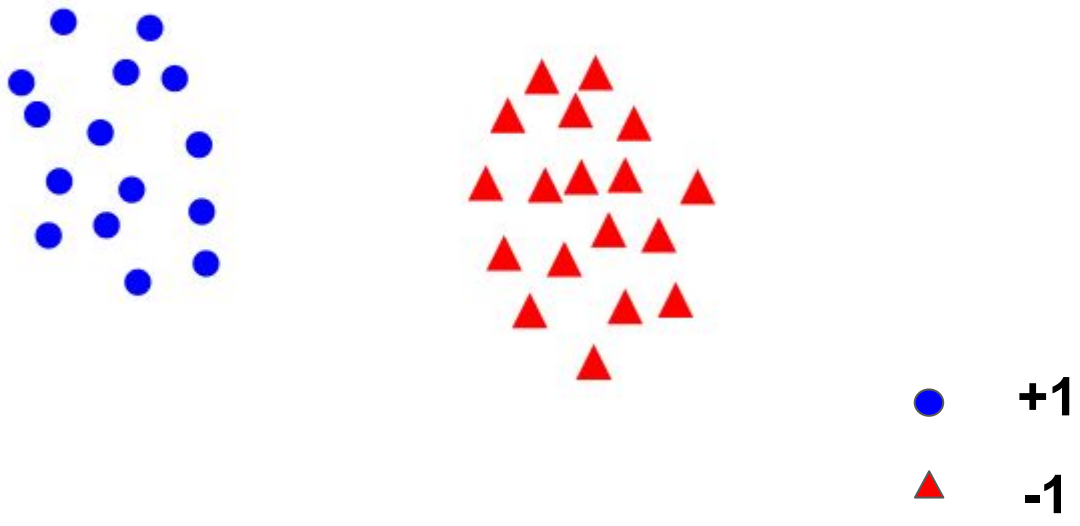
Supervised learning: methods

- Support vector machine (SVM)
- Neural network
- Decision tree
- Random forest
- ....

Support Vector Machine (SVM)

SVM classifier

In the simplest version, SVM aims to find the optimal (hyper)plane separating the two classes.

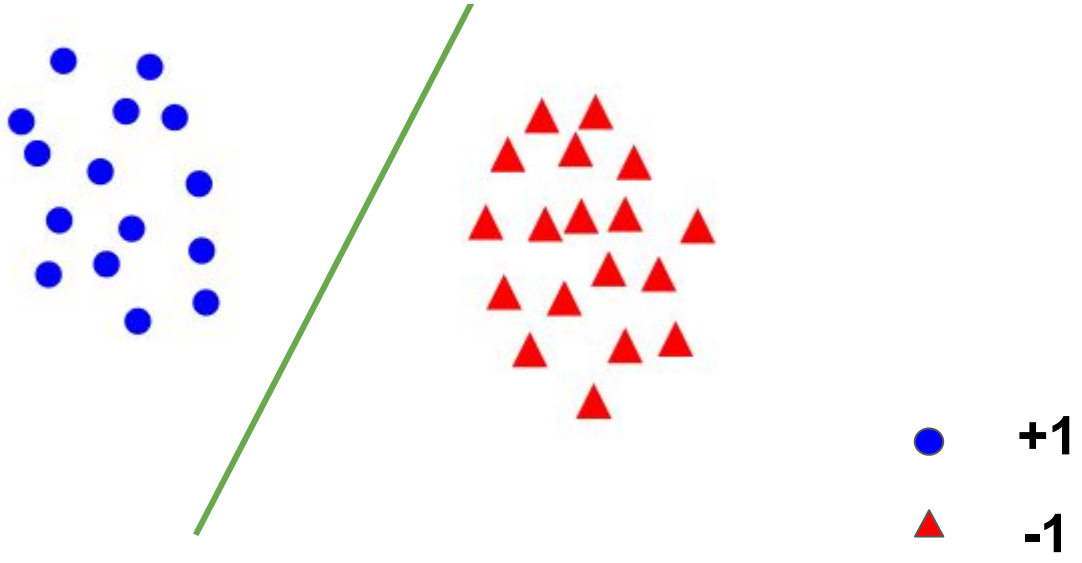


SVM classifier

In the simplest version, SVM aims to find the **optimal** (hyper)plane separating the two classes.

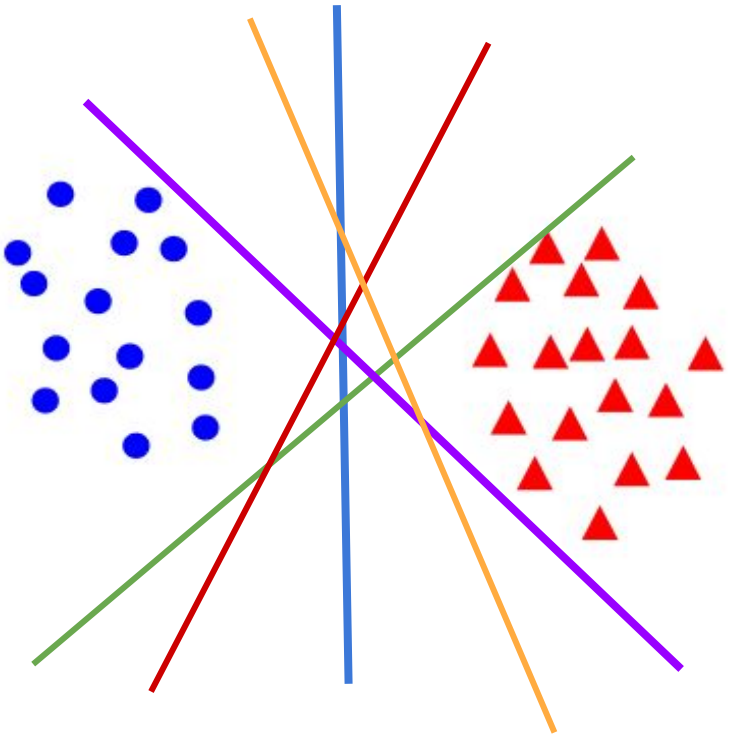
$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$$

$$f(\mathbf{x}_i) \begin{cases} \geq 0 & y_i = +1 \\ < 0 & y_i = -1 \end{cases}$$



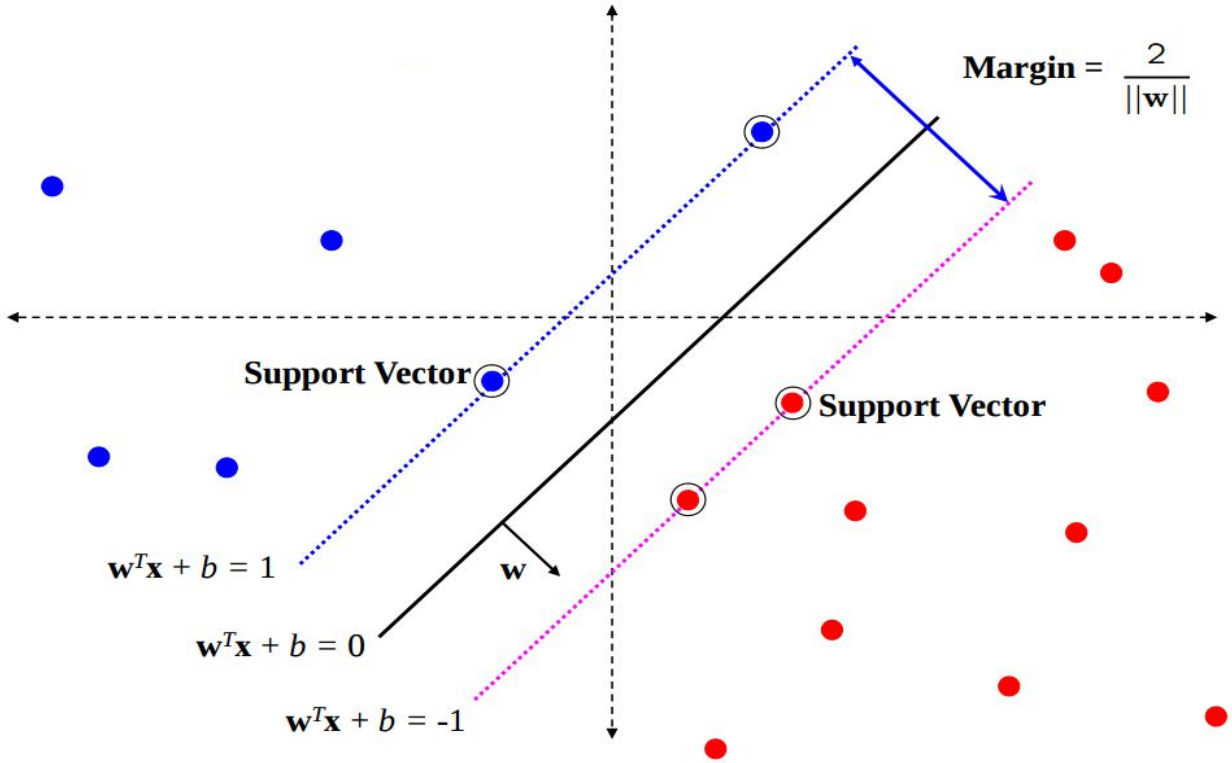
What is the optimal separator?

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$$



Basic idea in SVM: find the optimal separating hyperplane between the two classes by **maximizing the margin** between the classes' closest points (known as support vectors)

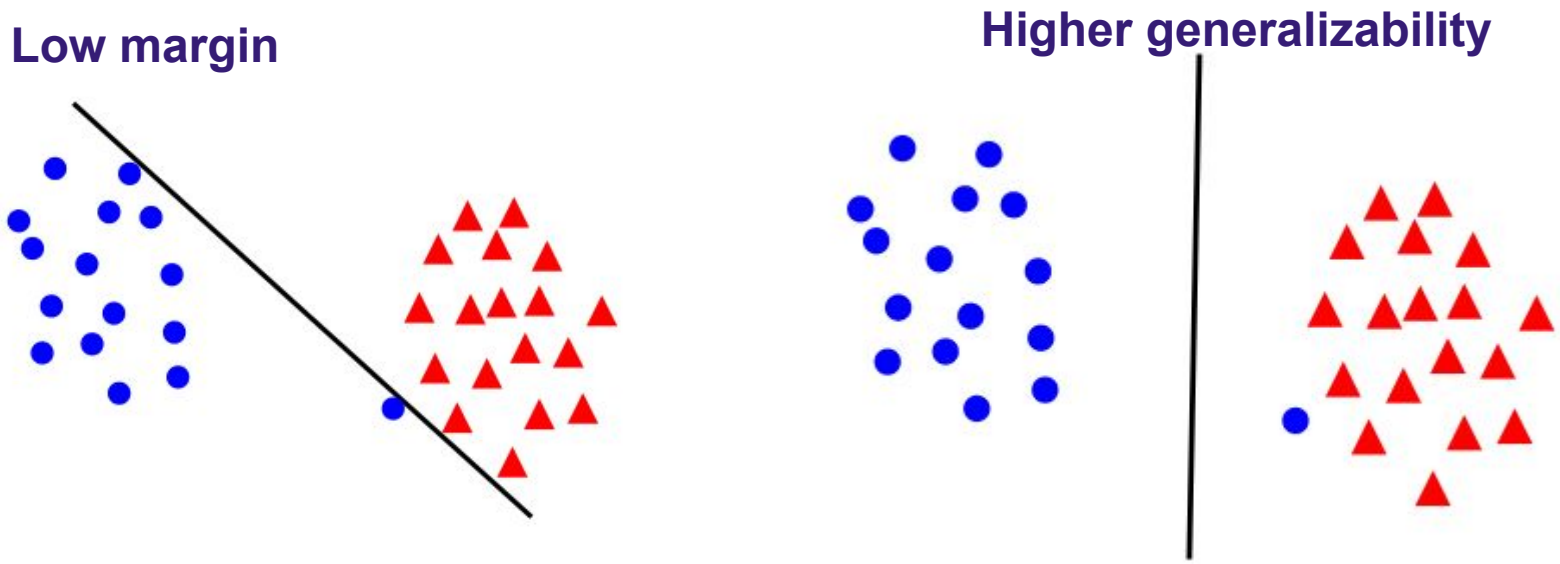
Margin maximization



$$\min_{\mathbf{w}} ||\mathbf{w}||^2 \quad \text{subject to } y_i \left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq 1 \text{ for } i = 1 \ldots N$$

For details refer to: http://www.robots.ox.ac.uk/~az/lectures/ml/lect2.pdf
http://www.tristanfletcher.co.uk/SVM%20Explained.pdf

SVM Regularization: Cost parameter (C)

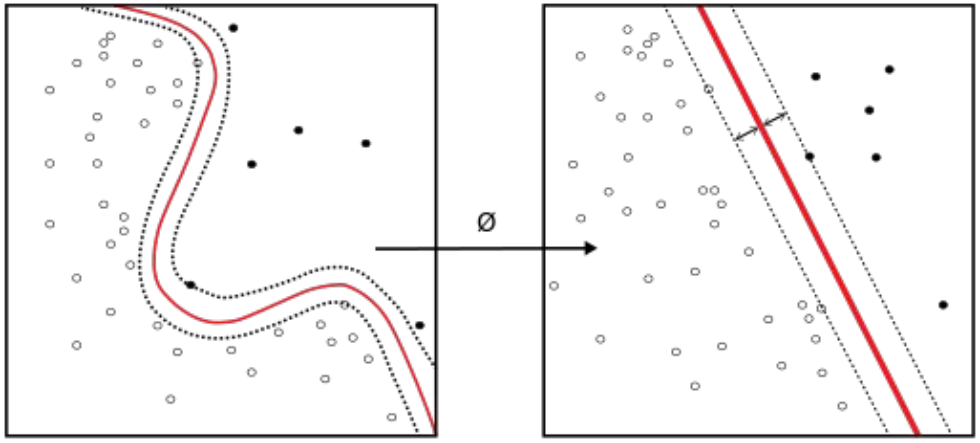


C parameter: Controls the trade off between the margin and the number of mistakes on the training data:

- **small C** → loose constraints → large margin
- **large C** → hard constraints → narrow margin

Non-linear classification

Kernel trick: apply a transformation on the data points to map them to a new space where they are linearly separable



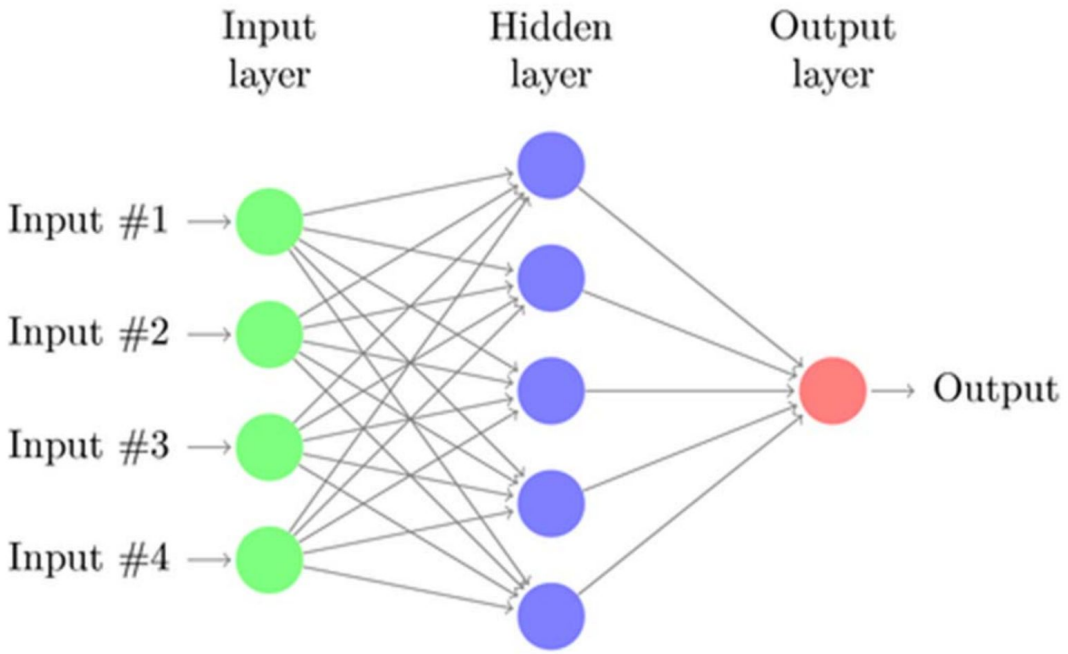
Gaussian kernels (**R**adial **B**asis **F**unction) are widely used in different applications

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\sigma^2}\right)$$

Neural Networks (NN)

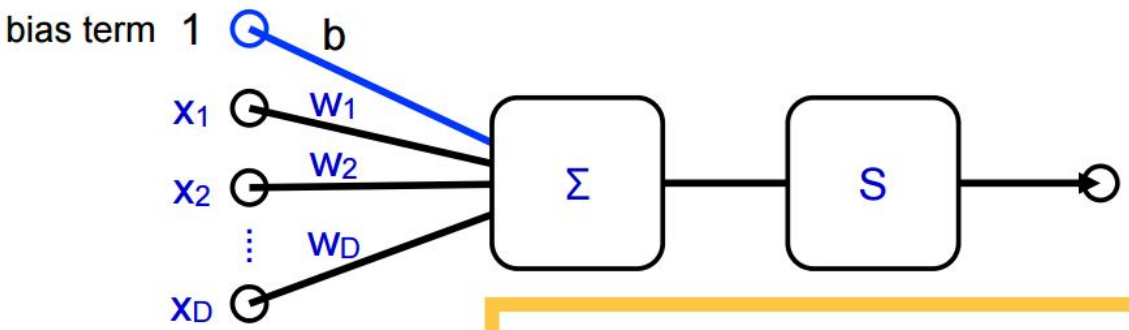
Neural networks

- Each input is a feature
- Weights of the connections are determined during the training phase



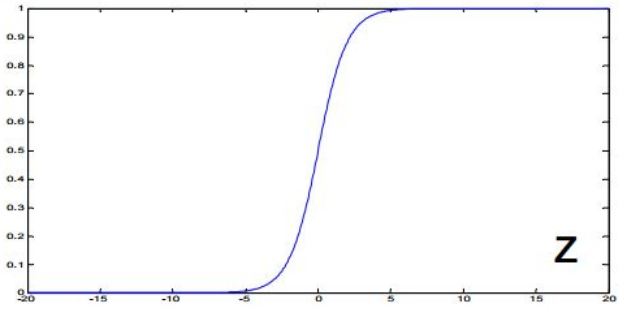
Single neuron - perceptron

[Rosenblatt 57]



$$f(\mathbf{x};\mathbf{w}) = S\left(\langle \mathbf{w}, \mathbf{x} \rangle + b\right) = \frac{1}{1 + e^{-w_1 x_1 - \cdots - w_D x_D - b}}$$

$$S(z) = \frac{1}{1 + e^{-z}}$$



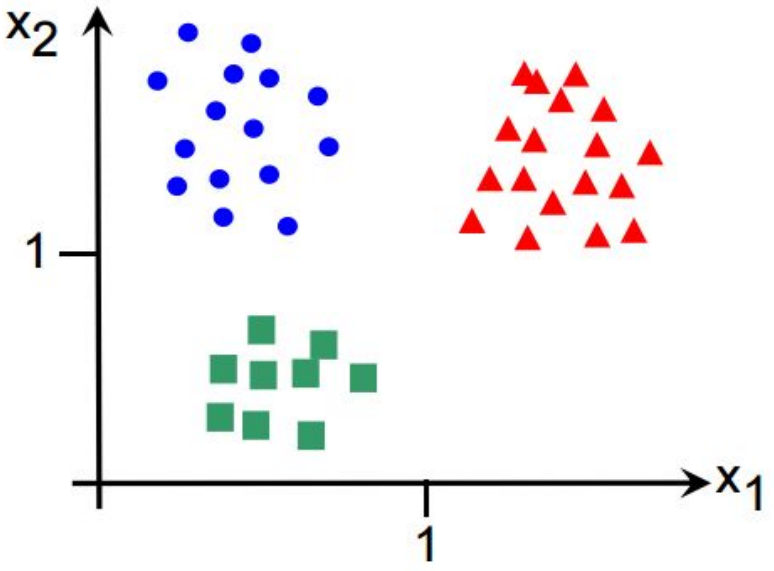
Perceptron steps:

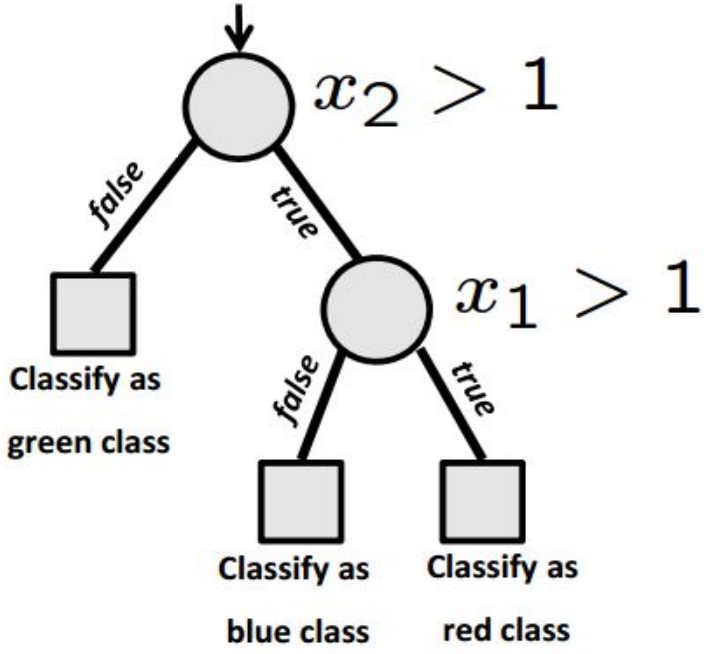
1. Map a vector **x** to a scalar score by an affine projection (**w**, b)
2. Transform the score monotonically but non-linearly by the sigmoid *S*()

http://www.robots.ox.ac.uk/~az/lectures/ml/lect4.pdf

Decision Trees

Decision tree





Decision tree splits the nodes on all available variables and then selects the split which results in most homogeneous sub-nodes.

http://www.robots.ox.ac.uk/~az/lectures/ml/lect4.pdf

Tree modeling parameters

- Minimum samples for a node split
- Minimum samples for a terminal node (leaf)
- Maximum depth of tree (vertical depth)
- Maximum number of terminal nodes
- Maximum features to consider for a best split (mtry)
    - As a thumb-rule, square root of the total number of features works well but we should check upto 30-40% of the total number of features.
    - Higher values might lead to overfitting

Ensemble Methods

Meta learners (ensemble methods)

Bagging

- Build models on *random subsets of the dataset* and combine the models through averaging (regression) or max vote (classification).

$$M = (M_1 + M_2 + M_3 + ... + M_n)/n$$

- **Example**: Random forests

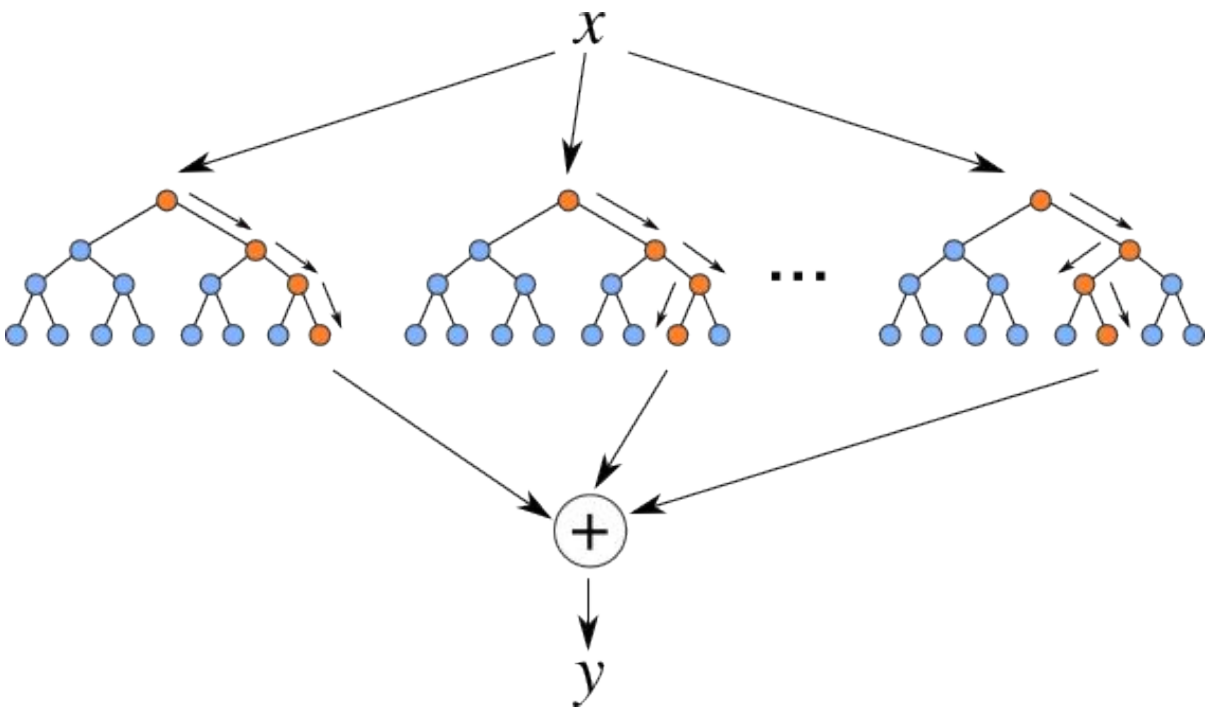
Boosting

- Build several small, simple, 'weak' models on the whole dataset and combine the models using weighted average approach.

$$M(x) = \beta_1 M_1(x) + \beta_2 M_2(x) + ... + \beta_n M_n(x)$$

- **Example**: gradient boosted model

Random forest



Training phase:

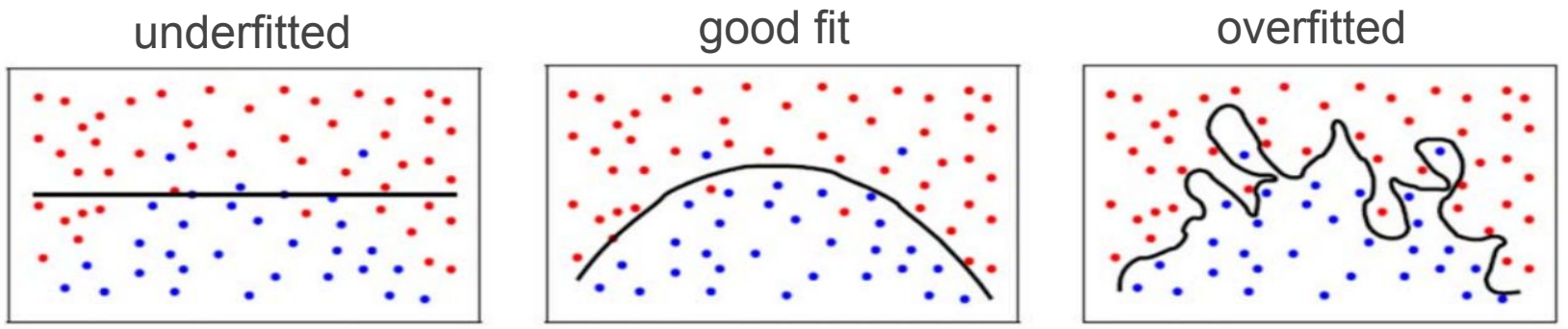
1. Sample, with replacement, *n* training examples
2. Train a decision or regression tree for each sample

Prediction phase:

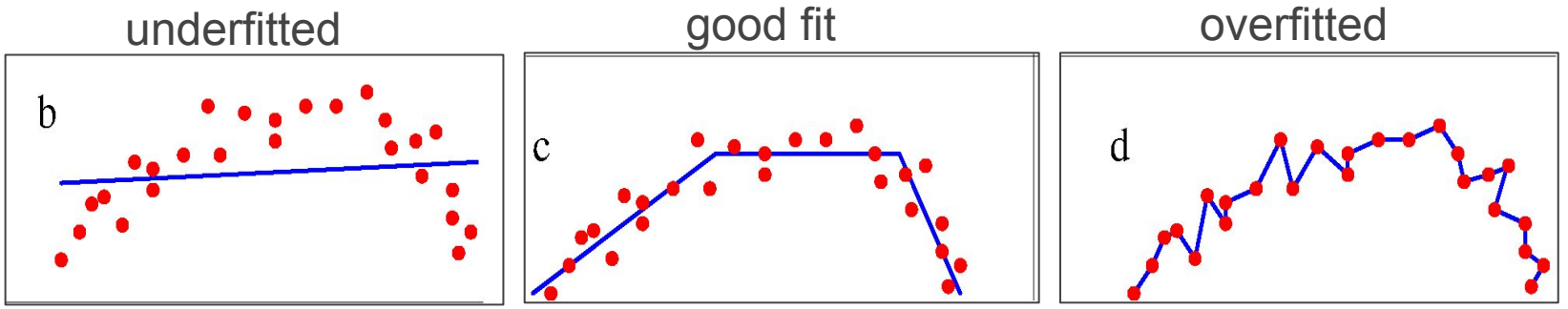
1. *Classification*: Return the majority vote
2. *Regression*: Return the average prediction

Trade off: goodness of fit and generalizability

- Classification



- Regression



Cross Validation

A model validation technique for assessing how well the model generalizes to new data.

- Split training data into **training** and **validation** set
- Build the models using training data
- Evaluate the model on validation data

Model Evaluation

Classification

- **Accuracy**
- **Kappa**

Regression:

- **RMSE**
- **R-squared**

Resources

Detailed documentation of CARET package:

- http://topepo.github.io/caret/index.html

Lists of available models:

- http://topepo.github.io/caret/modelList.html
- http://topepo.github.io/caret/bytag.html

A short introduction:

- https://cran.r-project.org/web/packages/caret/vignettes/caret.pdf