Machine Learning Algorithms

Yvenn Amara-Ouali

2024

General Information on the Course

Course Plan

31/01 – Feature Engineering, Linear models & SVM
07/02 – Random Forests
14/02 – Gradient Boosting
28/02 – Unsupervised Learning
06/03 – Project Session 1
13/03 – Project Session 2

Assignment – to send by March 20th EOD

A report of maximum 10 pages detailing a benchmark of Machine Learning Algorithms that you have applied to a data science problem

Reading List

- Mahesh, Batta. **Machine learning algorithms-a review.** International Journal of Science and Research (IJSR), 2020
- **An Introduction to Statistical Learning** by Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani

Contact

yvenn.amara-ouali@universite-paris-saclay.fr

What is Machine Learning?

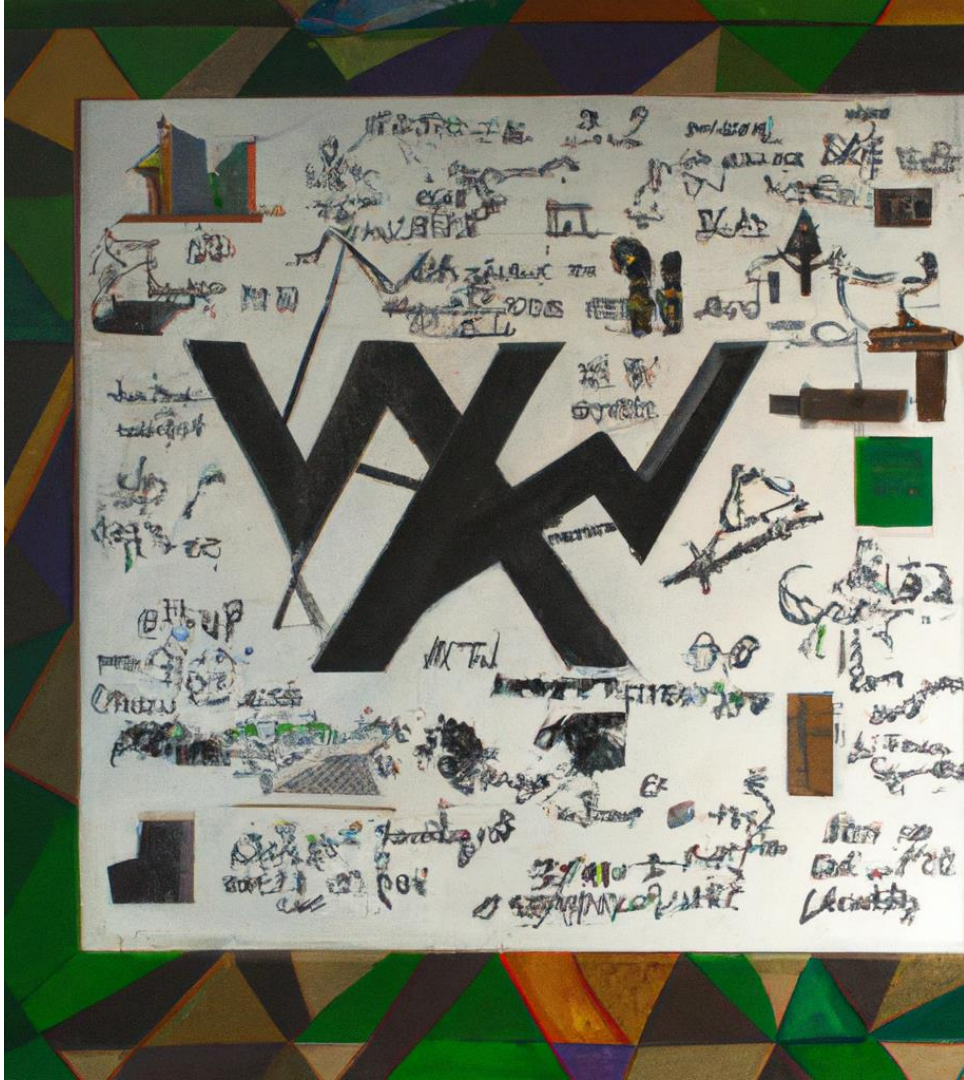
Historical Definition

Machine Learning is the field of study that gives **computers** the ability to **learn without being explicitly programmed**

Arthur Samuel (1959)

Feature Engineering

A crucial task



What is Feature Engineering?

Definition

- Feature Engineering is the process of **creating**, **transforming**, and **selecting** features from raw data for use in machine learning algorithms

Goal

- Improve performance of machine learning models by creating informative, relevant, and representative features.

Process

- Apply various techniques to extract meaningful information from the data
- Represent data in a format that can be easily understood and utilized by machine learning models

Outcome

- Improved performance of machine learning models by creating features that are more informative, relevant, and representative of the problem space

Why is it important?

The various tasks of a machine learning project

1. Data Collection
2. Data Cleaning
3. Feature Engineering
4. Model Fitting
5. Hyperparameter search
6. Model Validation

Which one of them (should) takes the most time ?

Why is it important?

The various tasks of a machine learning project

1. Data Collection
2. Data Cleaning
3. **Feature Engineering**
4. Model Fitting
5. Hyperparameter search
6. Model Validation

Which one of them (should) takes the most time ?

Why is it important?

The various tasks of a machine learning project

1. Data Collection
2. Data Cleaning
3. **Feature Engineering**
4. Model Fitting
5. Hyperparameter search
6. Model Validation

Feature engineering is one your strongest weapon regardless of the ML algorithm you use

Identifying Relevant Features

Exhaustive search

- Evaluates all possible combinations of input features
- Finds the input feature subset that gives the best accuracy for a selected model
- Computationally very expensive with larger number of input features

Forward selection

- Begins with a null feature set
- Adds one input feature at a time
- Evaluates accuracy of the model
- Continues until a certain accuracy is reached or a predefined number of features is reached

Backward selection

- Starts with all the features
- Removes one feature at a time
- Evaluates accuracy of the model
- Retains feature set that yields the best accuracy

Filter Methods

Approach

- Ranks features based on univariate metrics
- Selects the highest-ranking features

Metrics

- Variance ⇒ removes constant and quasi-constant features
- Chi-square ⇒ used for classification, measures independence of two variables.
- Correlation coefficients ⇒ removes duplicate features
- Information gain/mutual information ⇒ measures dependency of independent variable in predicting target variable

Remarks

- Filter methods are model agnostic
- Entirely rely on features in the data set
- Computationally fast
- Statistical approach

Scaling and Normalisation

Scaling

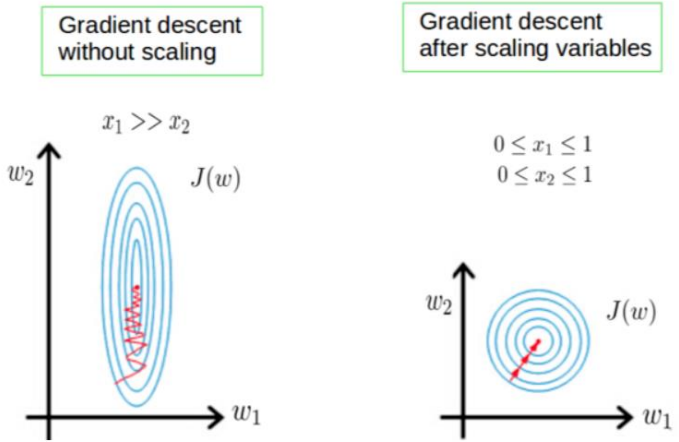
- Transforms features so that they have similar ranges or scales
- Examples - Min-Max Scaling, Standard Scaling, Robust Scaling

Normalisation

- Transforms features to have a 0 mean and a standard deviation of 1 $\rightarrow X' = \frac{X-\mu}{\sigma}$

Remarks

- Scaling should be done on the training set and the same scaling should be applied to the test set
- Decision trees and random forests, are not sensitive to the scale of features and do not require scaling or normalization



Encoding categorical variables (1/2)

One-hot encoding

- Creates a binary column for each unique category
- Assigns a value of 1 or 0 to indicate the presence or absence of the category
- Suitable for categorical variables with low cardinality (few unique categories)

Label encoding

- Assigns an integer value to each unique category
- Preserves the ordinal relationship between categories, if it exists
- May lead to incorrect results if the encoding creates an artificial order between categories

Numeric encoding

- Maps categories to numbers based on a specified criterion, such as frequency or average target value
- Often used for ordinal variables (categories with an inherent order).

Binary encoding

- Encodes categorical variables as binary bits
- Can handle high cardinality (many unique categories) and preserve sparsity

Encoding categorical variables (2/2)

Count encoding
- Replaces a categorical value with its frequency in the dataset
- Can handle high cardinality

Hashing encoding
- Maps categories to numbers using a hash function
- Useful for large datasets with many unique categories that cannot be one-hot encoded

Embedding encoding
- Maps categories to dense vectors that are learned during training
- Useful for high cardinality variables where it is hard to determine an appropriate encoding
- Typically used in deep learning models

Creating interaction features

Purpose

- To capture non-linear relationships between features
- To improve model performance by adding information that is not present in individual features

Examples

- Product features ⇒ the product of two or more features
- Polynomial features ⇒ a combination of powers and cross-terms of features

Remarks

- Interaction features can be created manually or through automated feature engineering techniques
- It increases the number of features, which can slow down model training and increase overfitting risks

Other feature creation

(Exponential) Smoothing

- Determine a smoothing parameter (α) that controls the weight given to past observations

$$\hat{x}_t = \alpha x_t + (1 - \alpha)\hat{x}_{t-1}$$

Aggregation

- Aggregate data by taking the mean, median, sum, count, etc. of a set of values over a specified time window or across categories

Binning

- Divide continuous data into discrete bins and encode the bins using one-hot encoding

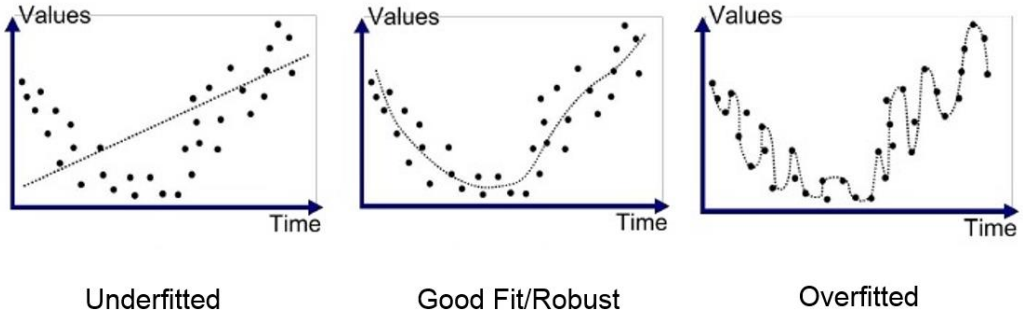
Derived Features

- Compute new features from existing features through mathematical operations such as logarithms, exponentials, etc.

Pitfall of feature engineering

Overfitting

- Risk of interpolation of training data

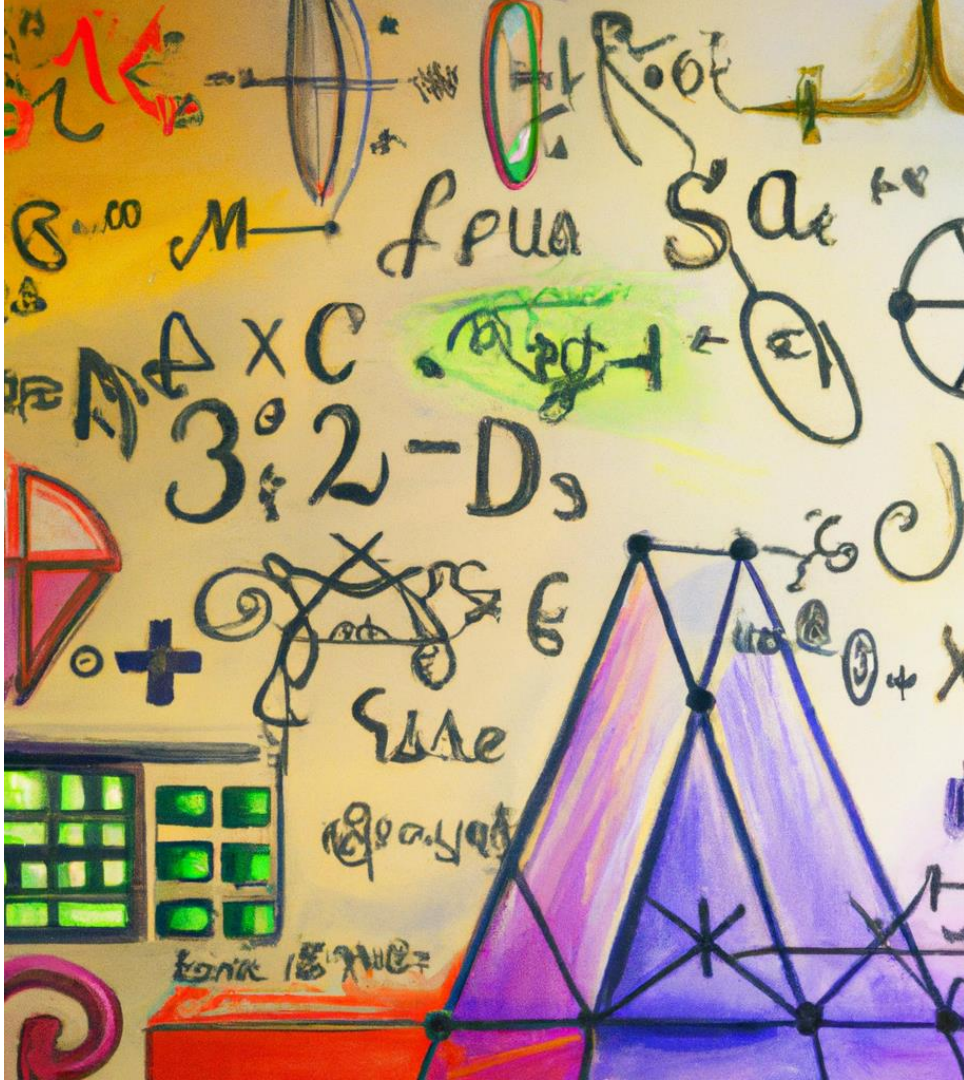


Data Leakage

- Data leakage refers to introduction of information into training data that is not available at prediction
- Occurs in feature engineering when a feature is created using target variable or test set information
- It leads to an artificial inflation of model performance

Additive Models

From Linear Regression
to GAMs



A variety of model classes

Linear Models (LM)

- Ordinary least-square regression

$$y_i = \sum_{j=1}^{d} x_i^j \beta^j + \epsilon_i = \boldsymbol{X}_i^T \boldsymbol{\beta} + \epsilon_i$$

Generalised Linear Models (GLM)

- Addition of a link function g

$$g(y_i) = \sum_{j=1}^{d} x_i^j \beta^j + \epsilon_i = \boldsymbol{X}_i^T \boldsymbol{\beta} + \epsilon_i$$

Generalised Additive Models (GAM)

- Addition of non-linear effects

$$g(y_i) = \boldsymbol{X}_i^T \boldsymbol{\beta} + \sum_{k=1}^{K} f_k(x_i^k) + \epsilon_i$$

Linear Models

Regression

- Linear regression assumes a linear relationship between the dependent variable and independent variables, which may not always hold true in real-world scenarios.
- It is simple to implement and can be used for both simple and multiple regression problems.
- Linear regression is sensitive to outliers and its performance may degrade in the presence of correlated independent variables (multicollinearity).

Estimation

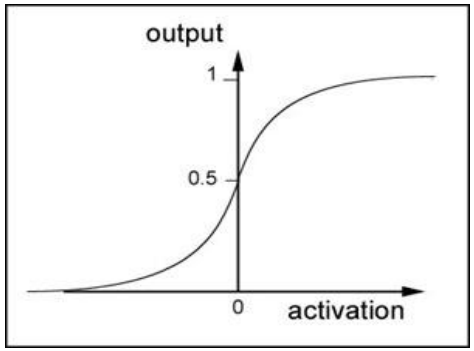
- Maximum likelihood ↔ Least-square

$$\widehat{\boldsymbol{Y}} = \boldsymbol{X}^T\boldsymbol{\beta}$$
$$\boldsymbol{X}\widehat{\boldsymbol{Y}} = \boldsymbol{X}\boldsymbol{X}^T\boldsymbol{\beta}$$
$$(\boldsymbol{X}\boldsymbol{X}^T)^{-1}\boldsymbol{X}\widehat{\boldsymbol{Y}} = \boldsymbol{\beta}$$

Linear Models

Classification

- Logistic Regression* is a statistical method for binary classification problems
- A linear combination of the features is transformed into a binary output using a logistic or sigmoid function
- The name "logistic regression" comes from the use of the logistic function to transform the output.



$$\hat{y} = P(y = 1|\boldsymbol{X}) = \sigma(\boldsymbol{W}^T\boldsymbol{X}) = \frac{1}{1 + e^{-\boldsymbol{W}^T\boldsymbol{X}}}$$

$$\log(\text{odds}) = \log\left(\frac{P(y = 1|\boldsymbol{X})}{P(y = 0|\boldsymbol{X})}\right) = \boldsymbol{W}^T\boldsymbol{X} \text{ is linear!}$$

*although called ‘regression’ it is a ‘classification’ algorithm!

Linear Models

Classification

- Logistic Regression*

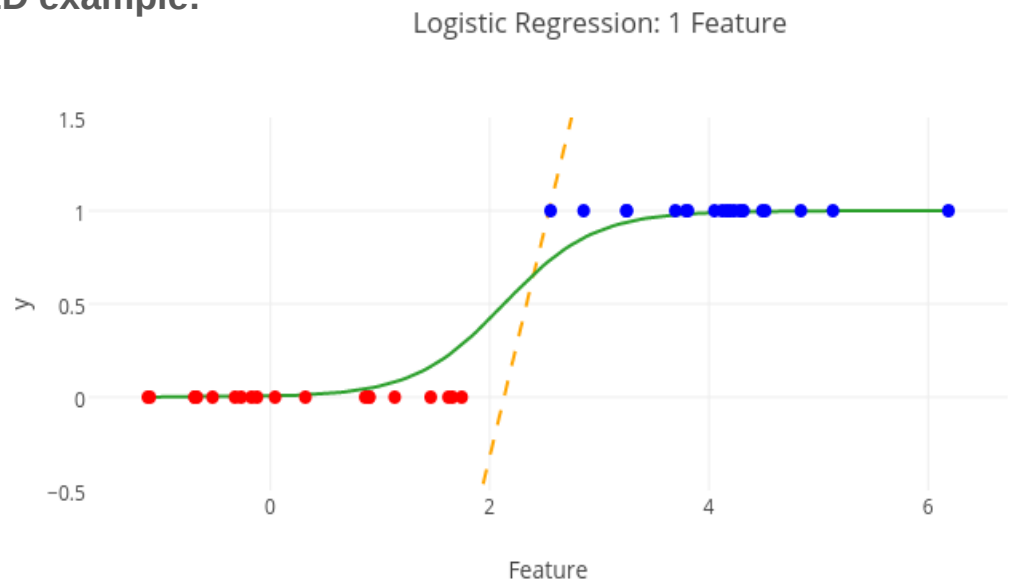
$\sigma(W^T X)$

$W^T X$

Remark

- Decision boundaries are still 'linear'
- The "contour lines" $(y(x) = cst)$ are non-linear, parametrizing the probability of the event being y=0 or y=1 as 'distance' from the boundary....

1D example:



*although called 'regression' it is a 'classification' algorithm!

Generalised Linear Models

Definition

- GLMs are a generalization of linear regression to allow for non-normal distributions of the response variable.

Example

- Poisson regression

Remark

- GLMs provide a more robust solution to linear regression in cases where the response variable has non-normal distribution, or when the relationship between the predictor variables and the response variable is not linear.

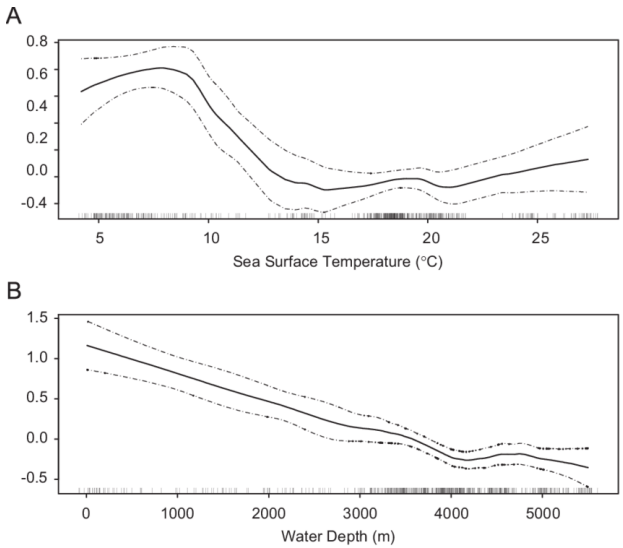
Generalised Additive Models

Definition

- Generalized Additive Models (GAMs) are a type of regression model that allow for non-linear relationships between the predictors and the response

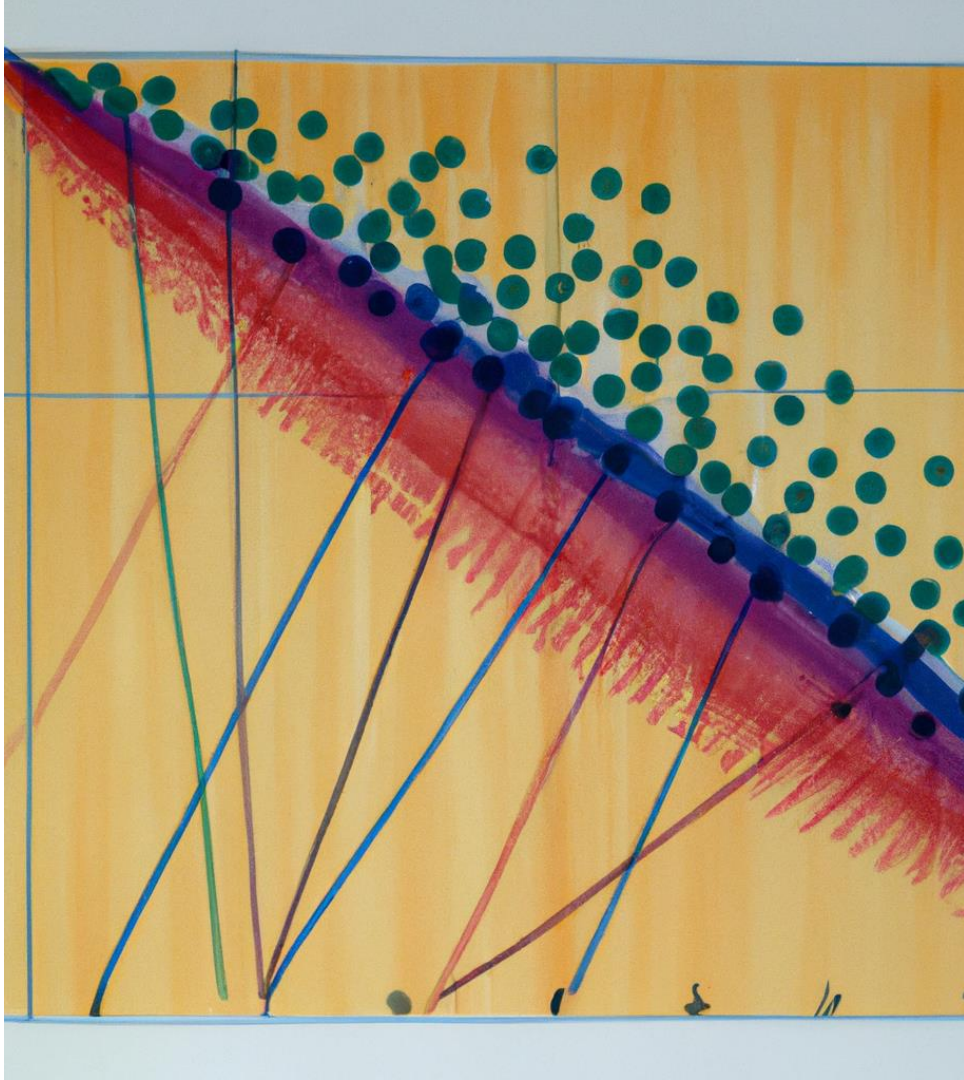
Remarks

- Unlike traditional linear regression, GAMs allow for non-linear effects of predictors by fitting smooth functions to the relationship between each predictor and the response
- These smooth functions can be estimated using a variety of methods, including splines and penalized regression



Support Vector Machines

Regression and Classification



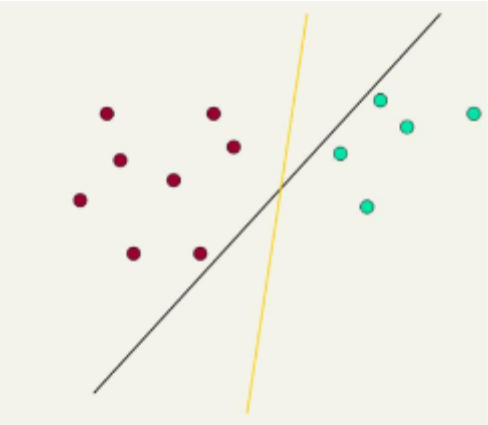
Which Hyperplane to pick ?

Problem definition

- Find an optimal hyperplane to separate red and green points

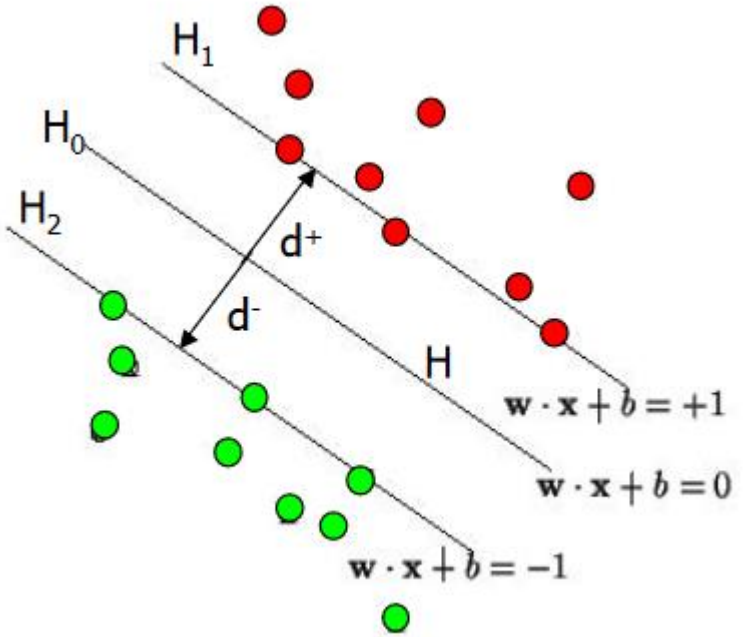
Support Vectors

- Support vectors are the elements of the training set that would change the position of the dividing hyperplane if removed from the training set
- The problem of finding the optimal hyper can be solved by optimization techniques
- We use Lagrange multipliers to get this problem into a form that can be solved analytically



Hyperplane definition

$$w.x_i + b \geq 1 \text{ when } y_i = 1$$
$$w.x_i + b \leq -1 \text{ when } y_i = -1$$



KKT conditions

Under our above assumptions, there must exist w^*, α^*, β^* so that w^* is the solution to the primal problem, α^*, β^* are the solution to the dual problem, and moreover $p^* = d^* = \mathcal{L}(w^*, \alpha^*, \beta^*)$. Moreover, w^*, α^* and β^* satisfy the **Karush-Kuhn-Tucker (KKT) conditions**, which are as follows:

$$\frac{\partial}{\partial w_i}\mathcal{L}(w^*, \alpha^*, \beta^*) = 0, \quad i = 1, \ldots, n \tag{3}$$

$$\frac{\partial}{\partial \beta_i}\mathcal{L}(w^*, \alpha^*, \beta^*) = 0, \quad i = 1, \ldots, l \tag{4}$$

$$\alpha_i^* g_i(w^*) = 0, \quad i = 1, \ldots, k \tag{5}$$

$$g_i(w^*) \leq 0, \quad i = 1, \ldots, k \tag{6}$$

$$\alpha^* \geq 0, \quad i = 1, \ldots, k \tag{7}$$

Moreover, if some w^*, α^*, β^* satisfy the KKT conditions, then it is also a solution to the primal and dual problems.

Non-linear SVMs