Machine Learning and Pattern Recognition Practice Session II

Martin Palazzo

Universite de Technologie de Troyes
Universidad Tecnologica Nacional Buenos Aires
Biomedicine Research Institute of Buenos Aires - Max Planck Partner

martin.palazzo@utt.fr

October 26, 2020

Overview

Disclaimer

The following document has been created as supporting and guiding material during the practical lessons during the Pattern Recognition course of the Master OSS. The official bibliography and theory materials have been distributed previously by the organization committee of the Master OSS.

Data Normalization

Data pre-processing

In many cases the features of a data set does not belong to the same domain. imagine the features of a car like its weight, the number of kilometers recorded and the mili liters of it liquid break system. The features are measured in Kilometers, Kilograms and mL. This means that the scale of the original features can be really different. Learning algorithm can be sensitive to this situation. For this reason we can make a transformation of each feature along the samples known as auto-scaling.

Feature standard scaling

$$z_{ij} = \frac{(x_{ij} - \mu_j)}{\sigma_j} \quad \forall j$$

Confusion Matrix

Confusion Matrix

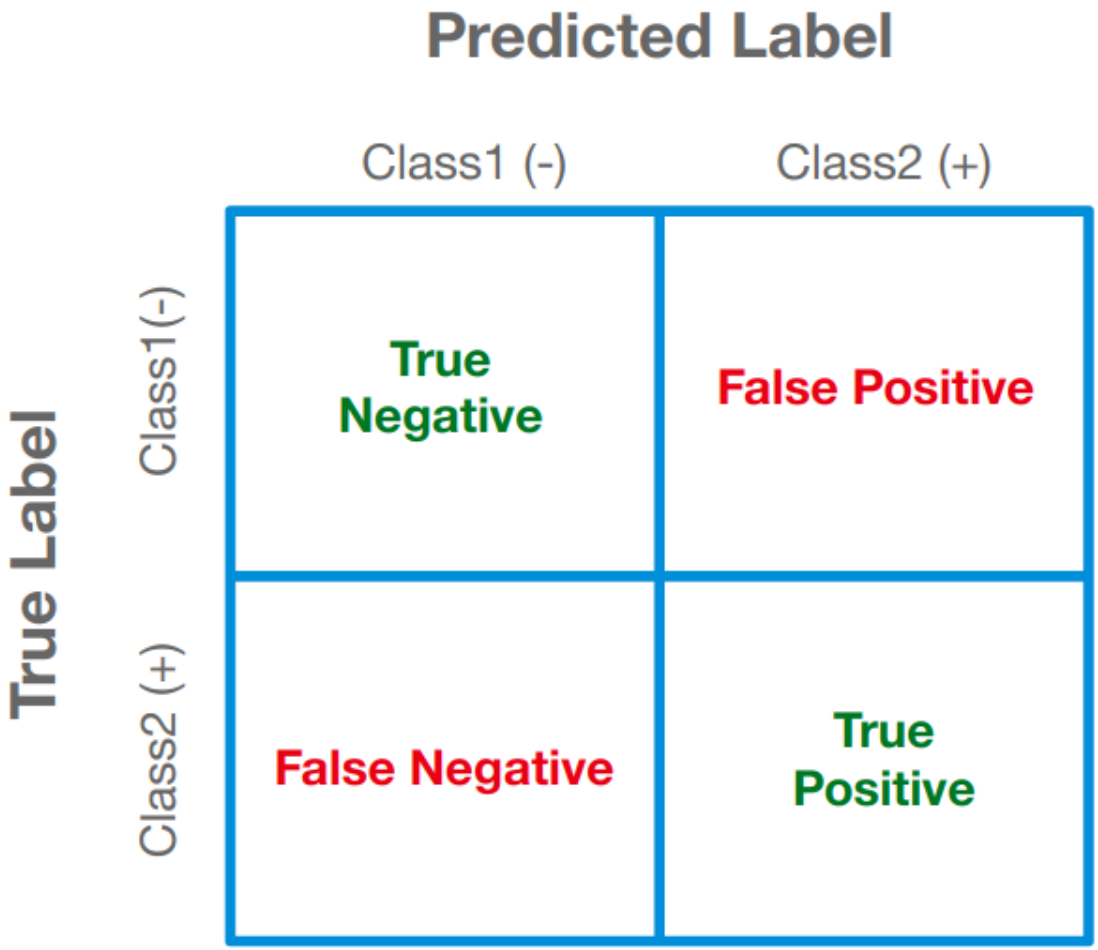


Figure: Confusion Matrix

Confusion Matrix

From the confusion matrix it is possible to obtain the following classification performance metrics

$$\text{TP} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

$$\text{Sensitivity (Recall)} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

Area under the ROC curve

The Area under the ROC curve

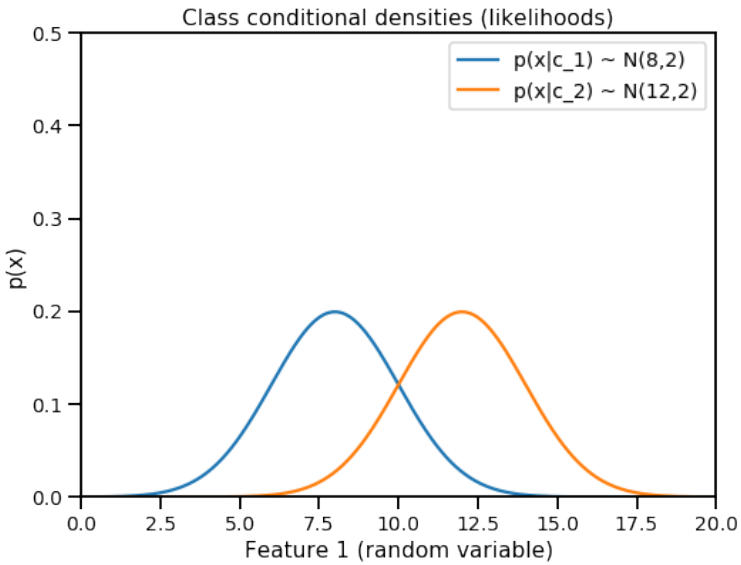
ROC curve

A Receiver Operating Characteristic curve (ROC) curve displays how well a model can classify binary outcomes. An ROC curve is made by plotting a false positive (FP) rate against a true positive (TP) rate for each possible cutoff value.

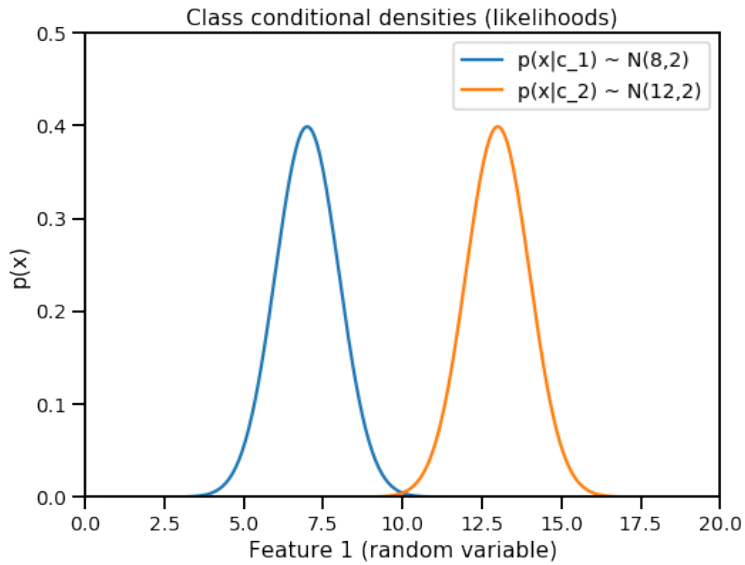
If we take the case of medical diagnosis, the ROC curve shows how well a model assigns correctly a healthy status to healthy people and a positive diagnosis to patients with disease. Considering the positive value to the Diagnosis status, by using different thresholds between the two classes it is analyzed:

- the fraction of positive diagnosis on patients with disease (TP rate).
- and the fraction of healthy patients that were incorrectly classified as positive diagnosis (FP rate).

The Area under the ROC curve



(a) Overlapped



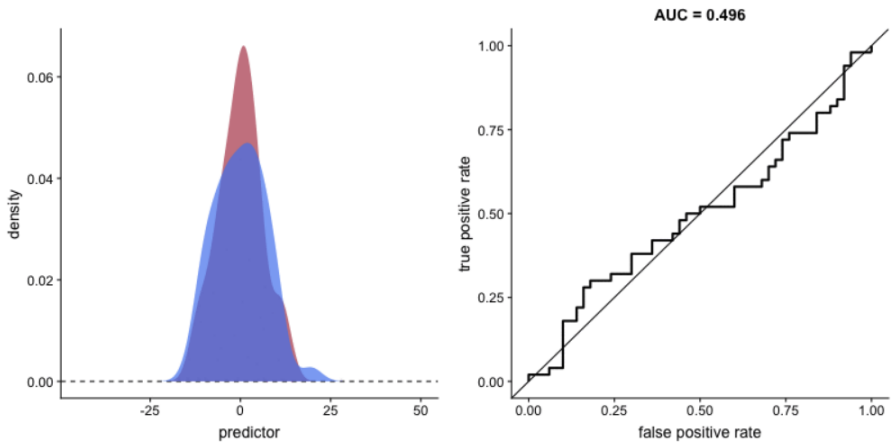
(b) Well separated

Figure: Class distributions overlapped and separated.

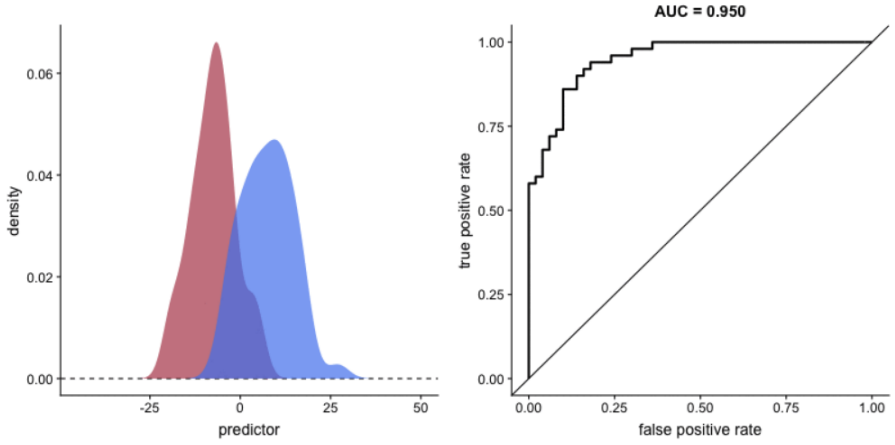
Question

If we build a classifier using the bayes rule, which case will present a better area under the curve (AUC) ROC?

The Area under the ROC curve



(a) Overlapped



(b) Well separated

Remark

The AUC ROC depends on the density functions of each class and the classifier used.

The Area under the ROC curve

Exercise 01

Generate 100 univariate samples of class 0 C_0 with $mu_0 = 6$, $\sigma_0 = 2$. Then generate 100 univariate samples of class 1 C_1 with $mu_0 = 13$, $\sigma = 2$. Generate 10 different thresholds (different priors) and compute the AUC ROC.

Exercise 02

Generate 100 univariate samples of class 2 C_2 with $mu_2 = 8$, $\sigma_2 = 2$. Then generate 100 univariate samples of class 3 C_3 with $mu_3 = 12$, $\sigma_3 = 2$. Generate 10 different thresholds (different priors) and compute the AUC ROC.

Train, Validation and Test sets

Train, Validation and Test sets

Given a data set $S = (x_1, y_1)...(x_n, y_n)$ we never will fit a classifier using the full set.

Train set

The set of samples $X_{tr} \in S$ used to build the model (classifier, regression function).

Validation set

The set of samples $X_{val} \in S$ used to measure the prediction performance of the classifier. Once it is validated, train and validation sets are merged to define a final model.

Test set

The independent set of samples $X_{te} \in S$ used to measure the prediction performance of the final classifier. These samples are never used during the training and fit process.

Cross Validation

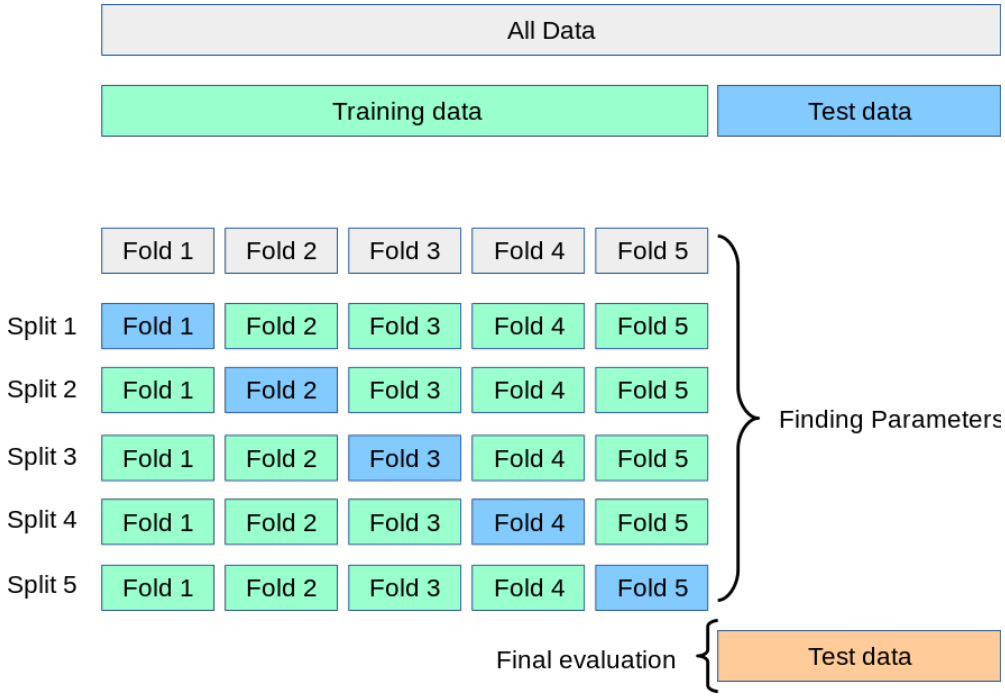


Figure: Cross validation process

5 fold cross validation

For cross validation first we can split the full dataset in 80% for train and 20% for test. Then make CV on training set.

Train, Validation and Test sets

Exercise 03

Given the breast cancer wisconsin dataset divide the data into train and test sets. Then using the training set train a KNN model by 5 fold cross validation. Compute the accuracy and empirical error of each fold. Select the best parameter and re-train using the full training set. Finally classify the rest of the test labels using the trained model.

Cost functions

Cost functions

We can measure the performance of our model by computing the Empirical Error defined as

$$J_{emp} = \frac{1}{n} \sum_{i=1}^{n} Q(d(x_i, \theta), y_i)$$

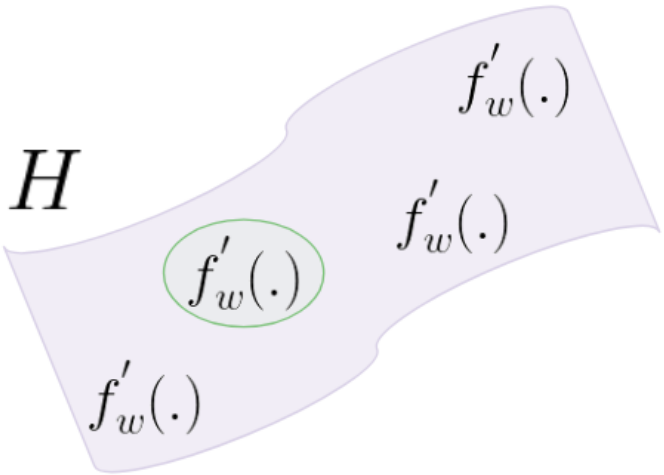
The error to minimize can be designed in many ways. There are two alternatives (among many)

$$Q(x, y) = (y - d(x, \theta))^2 \quad \text{The quadratic cost}$$

$$Q(x, y) = |y - d(x, \theta)| \quad \text{The absolute cost}$$

Classification with hyper-planes

Hypothesis space



$$H = \left\{ f_w^1(.), f_w^2(.), \ldots, f'_w(.), \ldots \right\}$$

Figure: Space of possible funtion defined by the parameters w

Complexity of the decision boundary

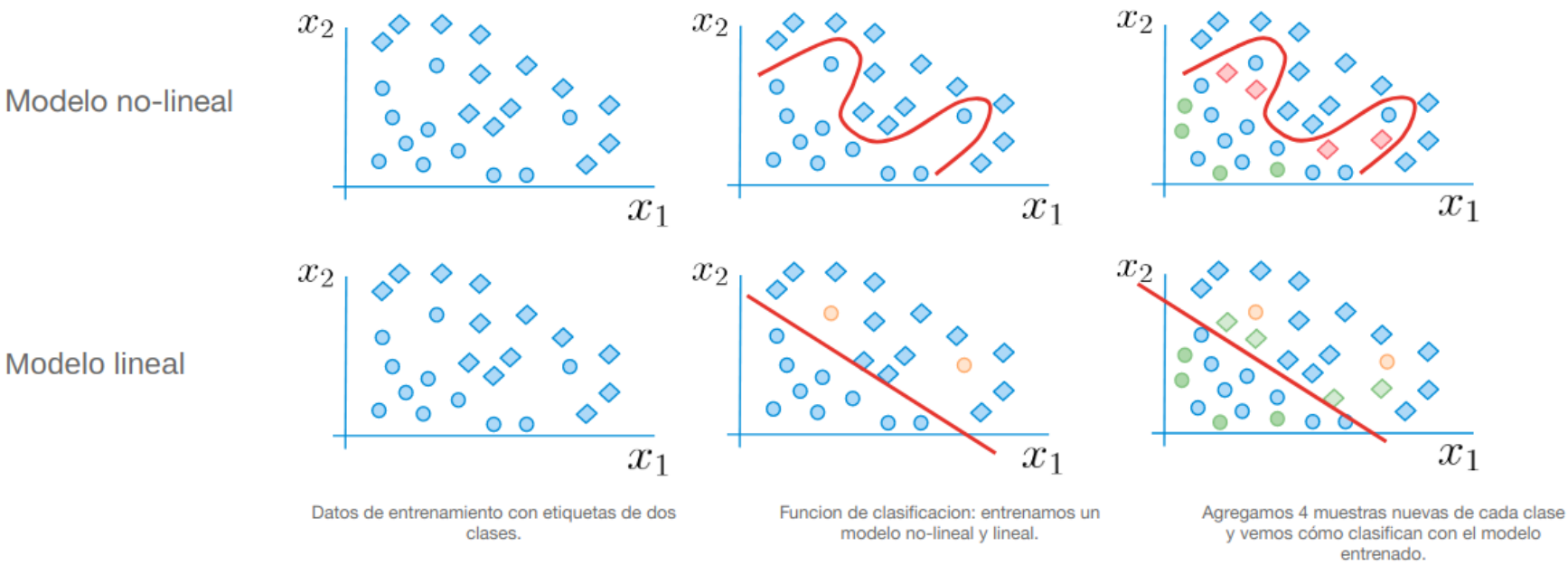


Figure: The complexity of the decision function may change the solution

Linear Classifier: hyperplanes

$$f(x) = b + w^T x = 0$$

$$D(x) = \text{sign}[w^T x + b]$$

There are many types of decision functions. The best-known family of functions is the linear ones. These functions are hyper-planes characterized by parameters w (vector w = [w1. . . .wd]) that will determine how the decision boundary is positioned in the hyper-space of dimension d. In binary classification, the decision function will assign a value of y = 1 or y = -1 depending on which side of the hyperplane the x samples are positioned.

Linear Classifier: toy exercise

$$x_{train} = \begin{bmatrix} 1 & 1 \\ 3 & 0.5 \\ 2 & 3 \\ 2.5 & 1.5 \\ 1.5 & 2 \\ 2.5 & 4 \\ 3.5 & 1.5 \\ 4 & 3 \end{bmatrix} \quad y_{train} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Given the following dataset compute manually the values of **w** and **b** that minimize the empirical error.

Perceptron

Perceptron classifier

The Perceptron algorithm is a linear classifier proposed in the 60s. Despite is not the best option to use nowadays, it represents the building block of neural networks and also the first successful linear discriminator in the modern history of pattern recognition.

$$y(x) = f(w^T x)$$

Perceptron is designed for binary classification and the idea is to output the following considering class 1 $C_1 = 1$ and class 2 $C_2 = -1$

$$f(x^T \phi(x)) = \begin{cases} +1 & \text{if } (w^T x) \geq 0 \\ -1 & \text{if } (w^T x) < 0 \end{cases}$$

So finally we want a function that $(w^T x)y > 0$

Perceptron classifier

Perceptron training

The perceptron model start with random weight initialization. The idea is to make iterations while computing the classification error. At each iteration if the error does not decreases, then the weights are updated by gradient descent.

Perceptron optimization

At each iteration, the error should decrease until convergence. This case can only occur if classes are completely separable. In case of not separable classes then the Perceptron optimization does not never converge to cero error.

Perceptron classifier

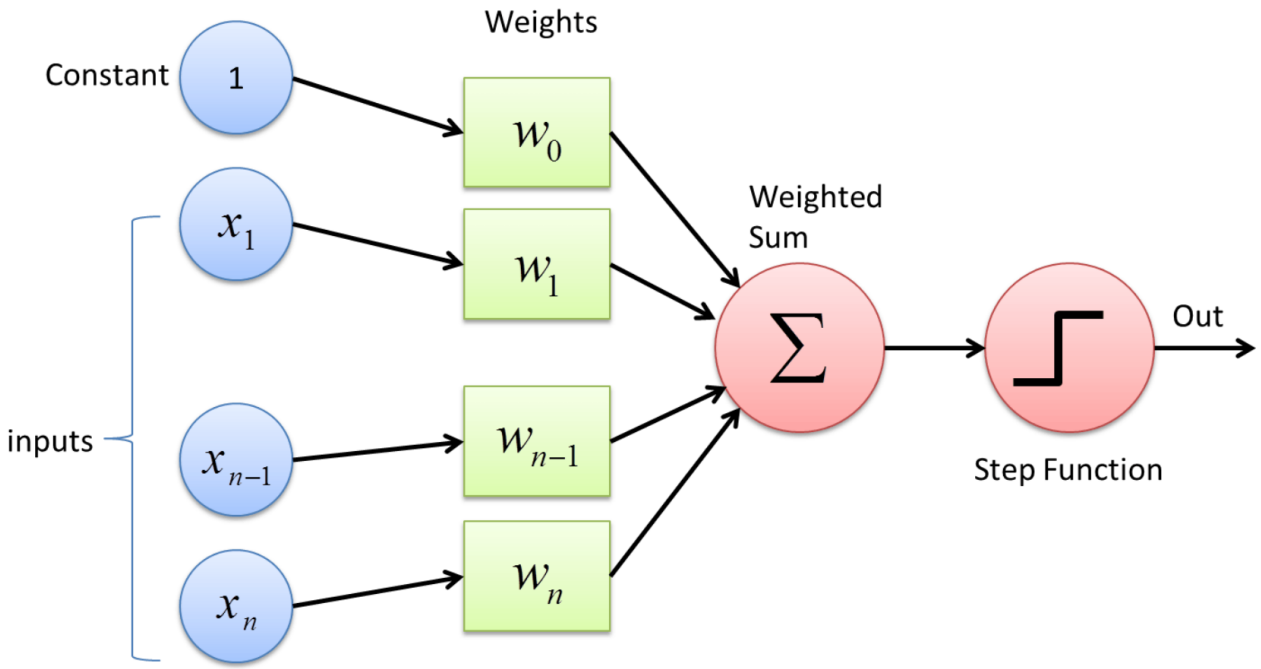


Figure: Perceptron architecture

Some limitations of the Perceptron lies in it can not converge when classes are not separable. In addition, for separable classes many solutions can occur. It has not any margin optimization!!

Perceptron classifier: toy exercise

How it started

Psychological Review
Vol. 65, No. 6, 1958

THE PERCEPTRON: A PROBABILISTIC MODEL FOR INFORMATION STORAGE AND ORGANIZATION IN THE BRAIN[1]

F. ROSENBLATT

Cornell Aeronautical Laboratory

How it is going

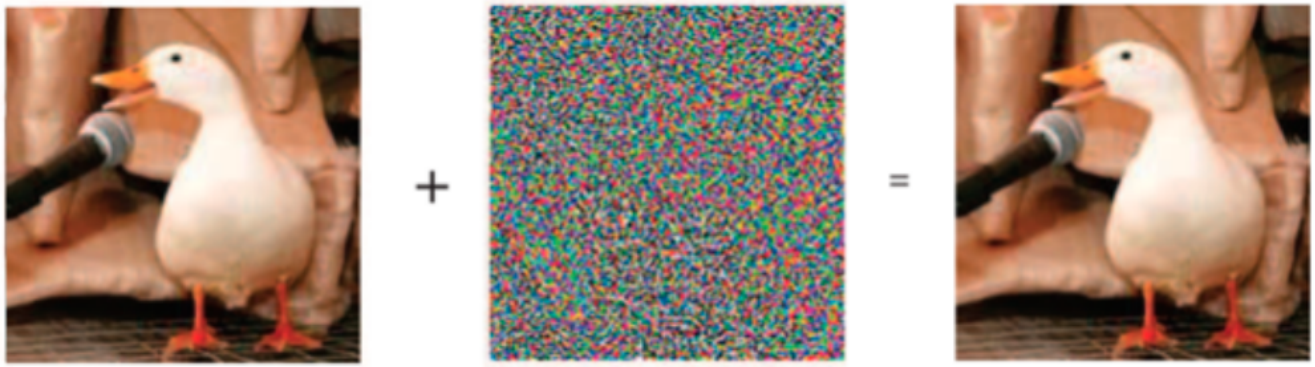


Figure: Perceptron dream

Perceptron classifier: toy exercise

Exercise 04

Design a two dimensional dataset where the perceptron will never find a solution and classes are completely separable.

Perceptron classifier

Exercise 05

Given the Wisconsin Breast dataset divide the data into train and test sets (equal to exercise 07). Then using the training set train a Perceptron model by 5 fold cross validation. Compute the accuracy and empirical error of each fold. Select the best parameter and re-train using the full training set. Finally classify the rest of the test labels using the trained model. Compare the results with the KNN.

Logistic Regression

Logistic Regression Classifier

The logistic regression classifier is a linear model embedded with a sigmoid function

$$p(y \mid x) = \sigma(w^\top x)$$

where the sigmoid function is

$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

Sigmoid Function

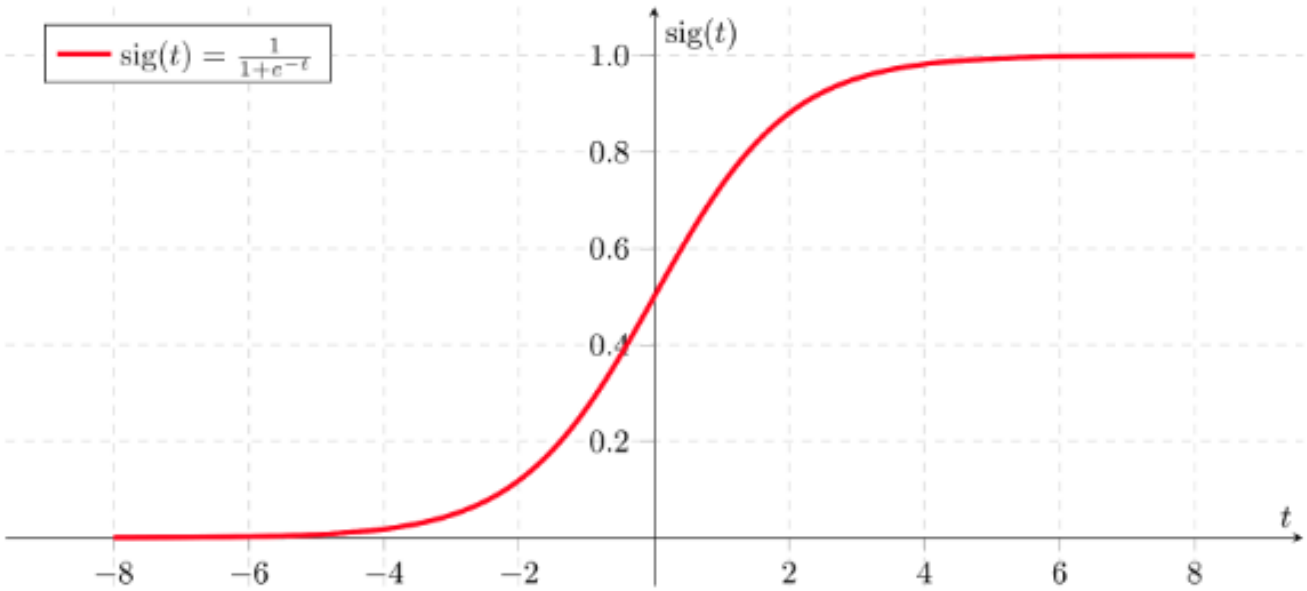


Figure: Sigmoid function

Regression

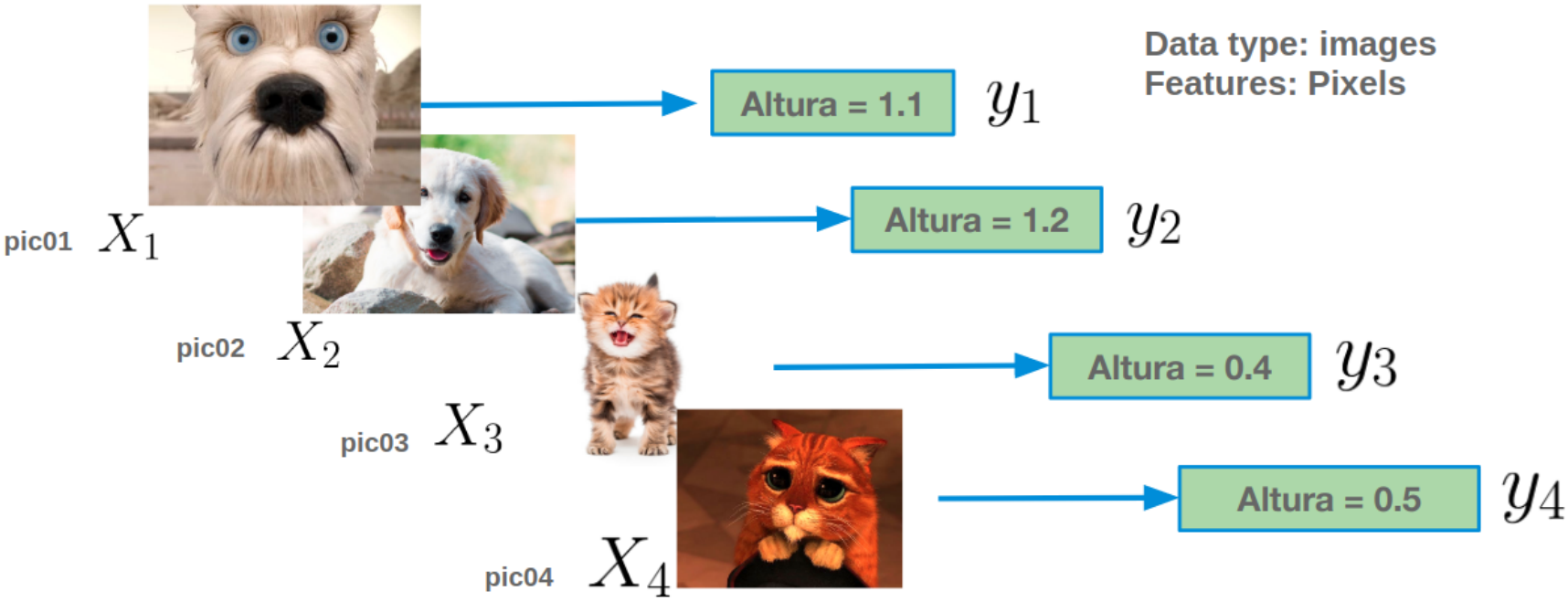


Figure: Regression

Regression

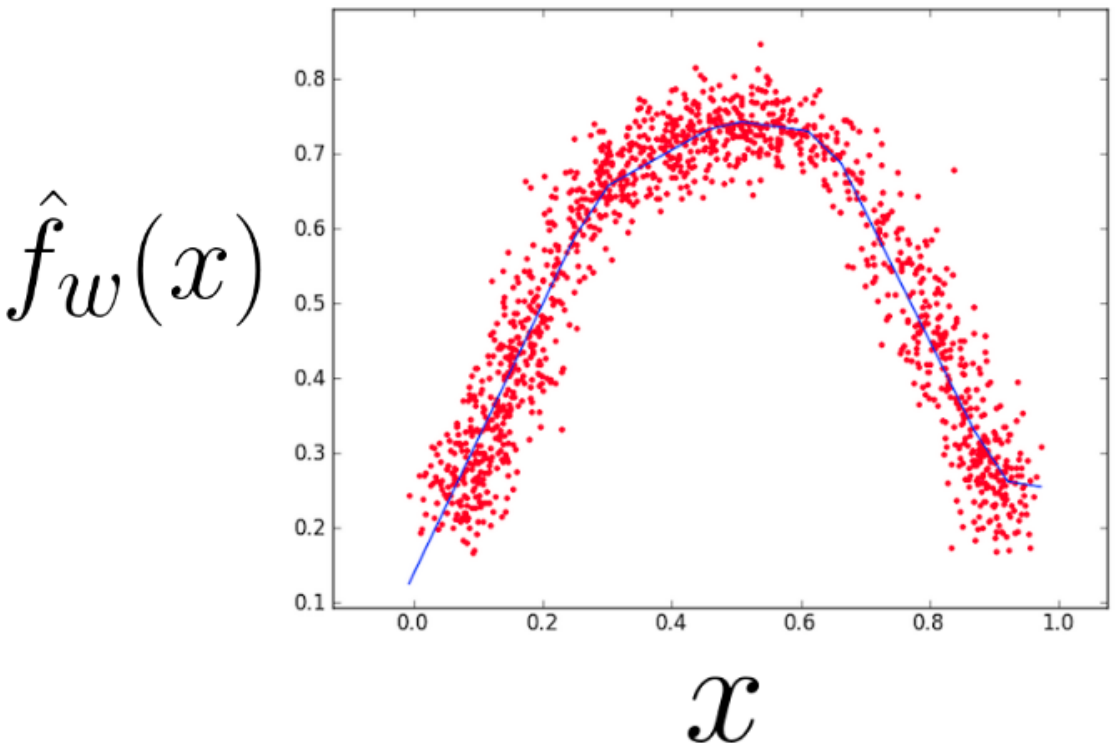


Figure: Regression function

Instead of learning a separating function now a regression function is learned.

Regression

Regression problem

$$\text{RSS}(w) = \sum_i^n (y_i - f(x_i))^2$$

$$\text{RSS}(w) = \sum_i^n \left(y_i - w_0 - \sum_j^d x_{ij} w_j\right)^2$$

Linear regression

$$\min_w \|Xw - y\|^2$$

The regression problem aims to find a vector w that minimizes the error between the predicted labels and the true ones.

Regression

Linear regression solution

$$\hat{w} = (XX^T)^{-1}X^T y$$

The regression problem aims to find a vector w that minimizes the error between the predicted labels and the true ones.

$$R^2 = \frac{\text{TSS} - \text{RSS}}{\text{TSS}}$$

$$\text{TSS} = \sum_{i=0}^{i=n} (y_i - \bar{y})^2 \qquad \text{RSS} = e_1^2 + e_2^2 + \cdots + e_n^2$$

Figure: How to compute the R squared metric.

Regression

$$\boldsymbol{MSE} = \frac{\sum (\widehat{y}_t - y_t)^2}{n}$$

$$\boldsymbol{RMSE} = \sqrt{\frac{\sum (\widehat{y}_t - y_t)^2}{n}}$$

Figure: How to compute the MSE metric.

Regression

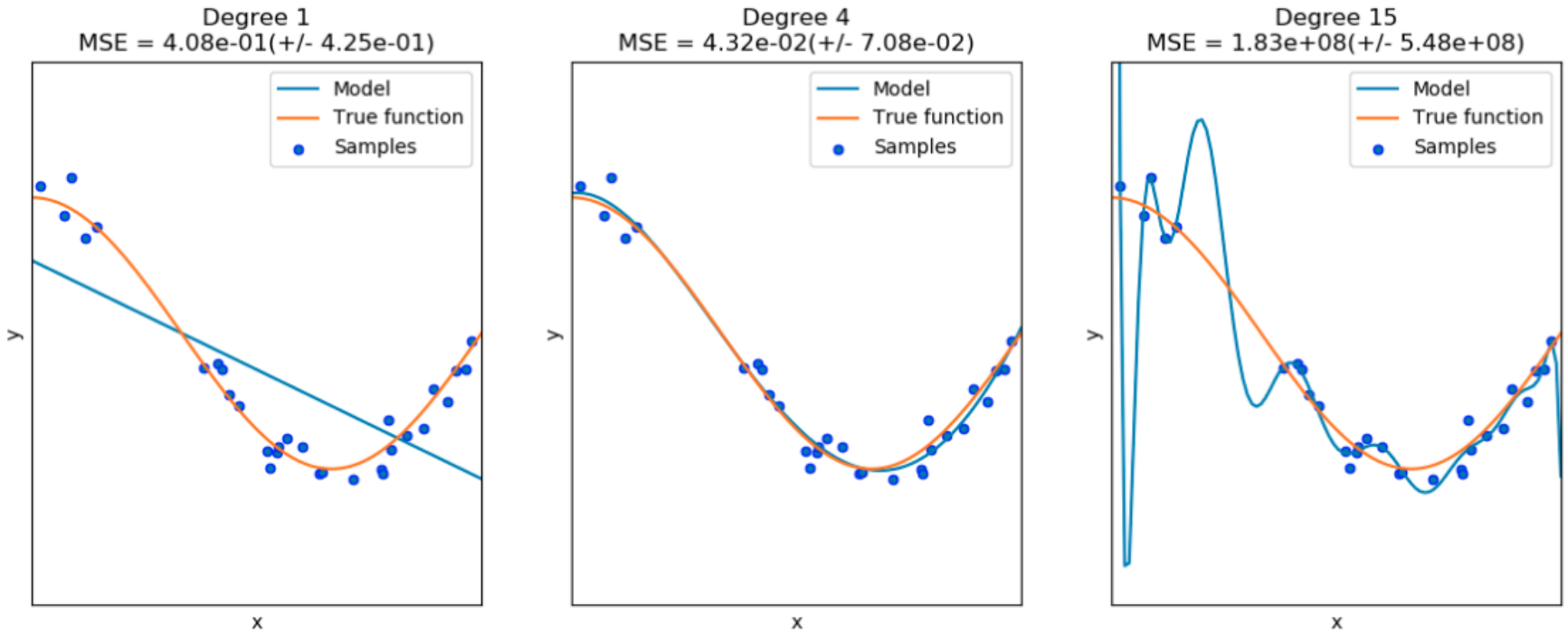


Figure: There exists a trade-off between complexity and simplicity of the learned function.

Instead of learning a separating function now a regression function is learned.

Regularization with ridge regression

Regularization can be used in any learning algorithm. Ridge regression shrinks the regression coefficients by imposing a penalty on their size [3]. The penalization term serves as a regularization policy that helps to make smoother functions.

$$\beta_{ridge} = argmin(\sum_{i=1}^{n}(y_i\beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2)$$

Ridge regression

By changing the λ value we change the β coefficients and thus the regression error. The higher the λ parameter, the higher the regularization and the simpler the model.

Regularization with ridge regression

Then, the solution of the beta coefficients will be

$$\beta_{ridge} = (XX^T + \lambda I)^{-1}.Xy$$

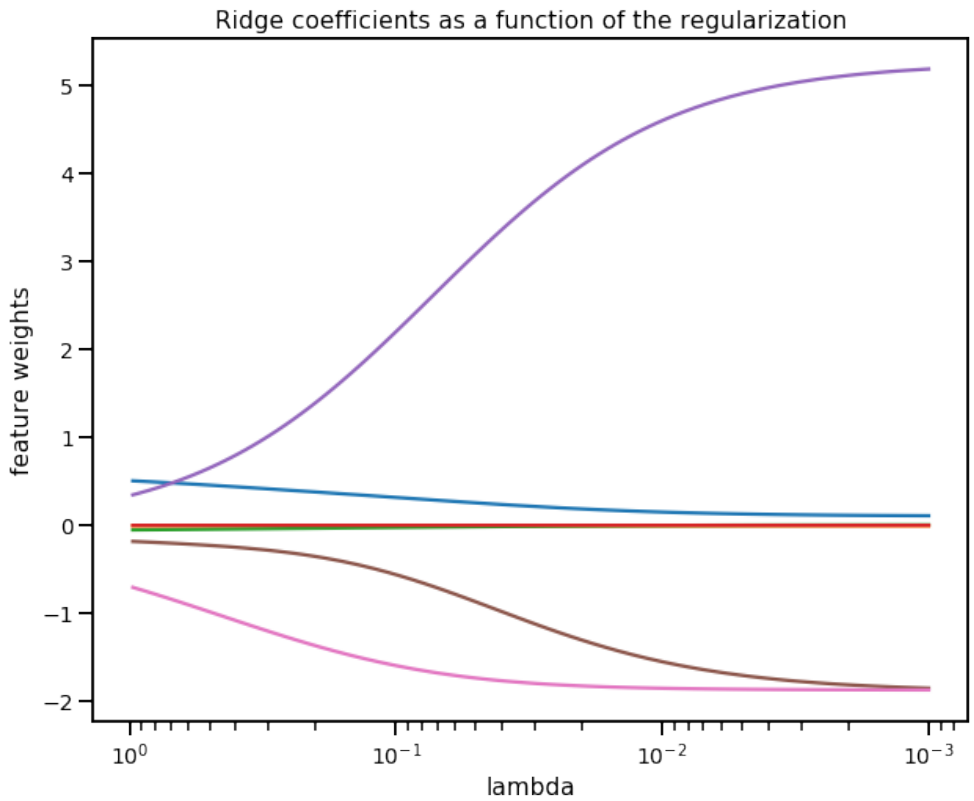


Figure: Ridge regression regularization for 6 features.

Regularization with ridge regression

Exercise 12

TBD

Exercise 13

TBD


Chollet, F. (2018). Keras: The python deep learning library. Astrophysics Source Code Library.


Bishop, C. M. (2006). Pattern recognition and machine learning. springer.


Friedman, J., Hastie, T., Tibshirani, R. (2001). The elements of statistical learning (Vol. 1, No. 10). New York: Springer series in statistics.