Machine Learning Formulas*

García Prado, Sergio
sergio@garciparedes.me

Taboada Rodero, Ismael José
ismaeljose.taboada@alumnos.uva.es

May 28, 2017

Abstract

[TODO]

1 Introduction

[TODO]

2 Supervised Learning

[TODO]

Error Measures [TODO]

- **Error Rate**: [TODO]
- **Resubstitution error**: [TODO]
- **Generalization Error**: [TODO]

Experimental Strategies [TODO]

- **Holdout**: [TODO]
- **Repeated Holdout**: [TODO]
- **Cross Validation**: [TODO]
- **Repeated Cross Validation**: [TODO]

2.1 Decision Trees

[TODO]

2.1.1 Information Theory

[TODO]

2.1.2 ID3

[TODO]

2.1.3 C4.5

[TODO]

*This document is being maintained in https://github.com/garciparedes/machine-learning-formulas

2.2 Rule Based Systems

[TODO]

2.2.1 1R

[TODO]

2.2.2 PRISM

[TODO]

2.2.3 IREP

[TODO]

2.2.4 RIPPER

[TODO]

2.2.5 PART

[TODO]

2.3 Instance Based Learning

[TODO]

2.3.1 K - Nearest Neighbors

[TODO]

2.3.2 IB3

[TODO]

2.4 Bayes Learning

[TODO]

2.4.1 Naive Bayes

[TODO]

2.4.2 K2

[TODO]

2.4.3 TAN

[TODO]

2.5 Linear Classifiers

[TODO]

2.5.1 Linear Regression

[TODO]

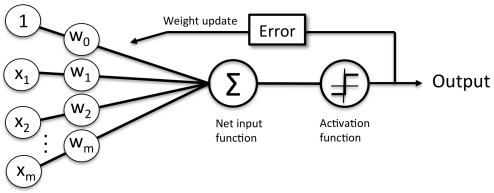

Figure 1: *General concept of perceptron*

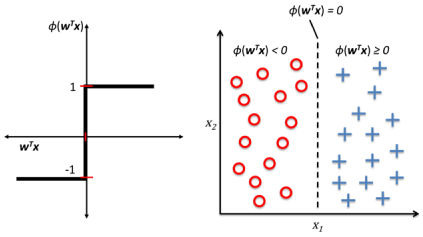

Figure 2: *Activation function*

2.5.2 Logistic Regression

[TODO]

2.5.3 Support Vector Machines

[TODO]

2.6 Neural Networks

2.6.1 Perceptron model

Perceptron's structures

$$w = \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix}, x = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix}$$

Ouput equation

$$z = w_1 x_1 + \cdots + w_m x_m = \boldsymbol{w^T x}$$

Activation function

$$\phi(z) = \begin{cases} 1 & \text{if } z \geq \theta \\ -1 & \text{otherwise} \end{cases}$$

Update of weight vector

$$w_j := w_j + \Delta w_j$$
$$\Delta w_j = \eta(y^{(i)} - \hat{y}^{(i)}) x_j^{(i)}$$

2.6.2 Adaptive linear neurons (ADALINE)

Cost function

$$J(w) = \frac{1}{2} \sum_i (y^{(i)} - \phi(z^{(i)}))^2$$

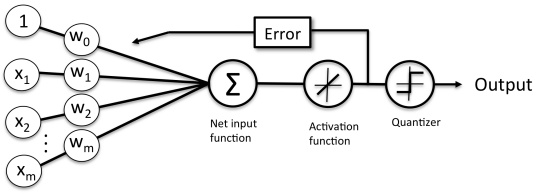

Figure 3: *General concept of adaline perceptron*

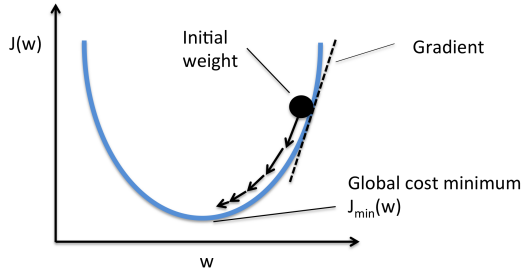

Figure 4: *Gradient descent*

Update of weight vector

$$w_j := w_j + \Delta w_j$$
$$\Delta w_j = -\eta \nabla J(w) = \eta \sum_i (y^{(i)} - \phi(z^{(i)}))x_j^{(i)}$$

Features standarization

$$x'_j = \frac{x_j - \mu_j}{\sigma_j}$$

2.6.3 Single Layer Neural Networks

Simple Perceptron [TODO]

ADALINE [TODO]

2.6.4 Multi Layer Perceptron

[TODO]

2.6.5 Radial Basis Functions

[TODO]

2.6.6 Convolutional Neural Networks

[TODO]

2.6.7 Recurrent Neural Networks

[TODO]

3 Unsupervised Learning

[TODO]

References

[CCAG17] Teodoro Calonge Cano and Carlos Javier Alonso GonzáLez. Técnicas de Aprendizaje Autómatico, 2016/17.

[Ols16] Dr. Randal S. Olson. Python Machine Learning, 2016.