ORB ACADEMIC PUBLISHER

Int. J. Comput. Sci. Theor. App., 2015, vol. 4, no. 3, p. 54-62.
Available online at www.orb-academic.org

International Journal of Computer Science: Theory and Application
ISSN: 2336-0984

Statistical Learning Theoretical Foundations Overview for Big Data Predictive Analytic

Smail TIGANI[1,⋆], Rachid SAADANE[2], Mohammed OUZZIF[3]

[1] *Computer Science Department, National High School of Electricity and Mechanics, Casablanca, Morocco.*
[2] *Electrical Engineering Department, HASSANIA School of Public Labors, Casablanca, Morocco.*
[3] *Computer Science Department, High School of Technology, Casablanca, Morocco.*
Email: sma.tigani@gmail.com, rachid.saadane@gmail.com, ouzzif@gmail.com
⋆ *Corresponding author.*

ABSTRACT
This paper presents a learning machine overview for Big Data Predictive Analytic. Produced data, in this decade, become bigger and bigger than ever. They have to be analyzed and processed in order to extract relevant knowledge to make predictive analytic. Learning machines comes at this stage to estimate predictors based on observed historical data. Learning algorithms performance and data quantity evolution must be parallel to keep tolerable performance. This parallelism is one of main challenges of Big Data field. For that reason, this work introduces the basic theoretical foundations of learning machines to push researchers to design new algorithms taking the data amount and performance aspect in consideration.

KEYWORDS
Statistical Learning — Algorithms Consistency — Decision Theory — Regression — Classification — Big Data Analytic

1. Introduction

Recent works in information technologies focus the cloud computing, mobility, big data and analytics, etc. In order to understand what is it exactly, Seth Earley (2015) summarized, in [1], the presentation done by Mike Kuniavsky in 2014 named "The User Experience of Predictive Analytics in the Internet of Things" in which he suggested that virtually all functionality resides, or will soon reside, in the cloud. Using different devices, the data and functionality can be accessed from any location. The access context and policy will be managed by specialized devices. Senior traders and banks for example can access to collected data from all financial markets followers by analysing comments or social networks or also collected data from binary options speculators. The data provided by the mobile devices can offer additional insights about the preferences of the user or also it location, which can be useful to propose to him some products or developing new features. This can be applied in many life's area and allow there actors to extract relevant information to there activities such us demographic data, lifestyles, financial markets, etc. This information has value to marketers, insurance companies, governmental agencies and traders.

As explained in [1], machine learning algorithms can be used to make predictive analytics based on observed historical data. Many tools are provided now For example, recovery rates for cardiac was correlated with activity data patients and also investment volume can be correlated with the trend of financial market. Other consumer devices include those that learn from voice patterns, such as a personal-assistant, or in addition systems that learn from much more complex behavior and activity patterns like Jaguar's or Land Rover monitoring system, etc. Learning performance is discussed in [2]: current artificial intelligence algorithms knows some limitations. Many recent researches focus on the building of new scalable algorithms and some other approaches uses multiple parallel processors to this end.

Sentiment classification and detection is one of the main applications of Big Data Analytics and Learning. Bingwei Liu and al (2013) affirmed, in [3], that Naive Bayes classifiers are widely used in information fusion. Many other applications are enumerated such as robotics control, imaging, text, and cyber analysis, etc. Anil K. Jain and al (2000) in [6] added some other application field of pattern recognition, which is one of main learning problems. We cite for example biometric recognition for personal identification, data mining to extract hidden information, speech recognition for many applications like a phone directory without any human intervention, and

document classifications to make research in the internet easier, etc.

This paper is organized as follows: the first section introduces the basics of machines learning then the foundation of the theory of learning. The second section presents the generalization aspect of the learning process followed, in the third section, by a brief overview of main regression and discrimination algorithms. Principal machine learning overview is reported to the fourth section and a conclusion in the last one.

2. Learning Theory Overview

The supervision and management of random environments, namely the temperature in a given time or date and the number of visitors of a ticket office at a given period namely from 12:00 to 15:00 is one of the arduous task of system decision. This is due to the non existence of deterministic models for the identification of the strategy adopted by the environment. Learning machines, one of artificial intelligence axis, is involved in this kind of scenario to identify, asymptotically, the behavior of the environment. This identification process requires observations, in the form of inputs and outputs archive, to estimate the decision rule. Technically, the archive of observations is called **Learning Data Set**.

We assume X the number of customers in the queue of an ATM from 12:00 to 15:00, and Y the amount of money taken from guichet in the same interval. The values of X and Y are not the same every day, that comes to conclude that X and Y are random variables. To ensure good quality of service, the bank must ensure that the wicket contains enough money to service customers during this critical period of the day. The bank should therefore to analyze the correlation between the number of visitors and total demand of clients. This analysis operation should be done automatically by collecting data (total deposit, sufficient or not) each day: this is a learning process.

2.1 Learning Data Set

Let suppose (X,Y) a couple of random variables independently and identically distributed on $\mathscr{X} \times \mathscr{Y}$ with a probability density P. We have collected, by observing, a set D^n containing n observations of the couple (X,Y) denoted $D^n = \{(x,y), x \in \mathscr{X}, y \in \mathscr{Y}\}$. The set D^n is called **learning data set**.

2.2 Regression Rule

The learning algorithm built from the set of observed data D^n, a strategy that combines each future entry point to the corresponding output point. This strategy is called regression rule or a predictor that formally defined by a measurable function $f : \mathscr{X} \to \mathscr{Y}$ which combines the output $y = f(x) \in \mathscr{Y}$ to $x \in \mathscr{X}$. The regression rule is estimated in hope to represent the future observations.

2.3 Lost Function

In order to measure the quality of prediction, we define the loss function $l : \mathscr{Y} \times \mathscr{Y} \to \mathbb{R}^+$. This function is defined as following $l(y_i, y_i) = 0$ and $l(y_i, y_i') > 0$ if $y_i \neq y_i'$. The classic definition of the loss function depends on the type of regression. For a real regression, the loss function is defined as the following $l(y_i, y_i') = |y_i - y_i'|^p$. For the binary discrimination it is given by $l(y_i, y_i') = \frac{1}{2}|y_i - y_i'|$.

2.4 Risk Function and Empirical Risk

In general way, we are interested in the average behavior, denoted $R(f)$ of the function $l(Y, f(X))$. Formally:

$$R(f) = \mathbb{E}_{(X,Y)\sim P}[l(Y, f(X)] \tag{1}$$

The risk minimization for a prediction rule f comes to minimize $R(f)$ (equation (1)) which depends on an unknown probability distribution P. It is evident so to measure the risk $R(f)$ based on an empirical probability density estimated using observed data ranged in D^n. This empirical value is called **Empirical Risk** denoted mathematically $\widehat{R}(f, D^n)$. It is in fact sample based estimator. Is is given formally by the average of experimental values y_i and x_i which are the observations on the random variables X and Y :

$$\widehat{R}(f, D^n) = \frac{1}{n}\sum_{i=1}^{n} l\left(y_i, f(x_i)\right) \tag{2}$$

The empirical risk minimization was developed in Vapnik (1999) [5]. Let f^* be the best predictor or the **Oracle**: the rule in the collection $\mathscr{F}$ that minimizes the risk R. The main goal is to find the nearest rule $\widetilde{f}$ from the oracle in term of risk. It means that the distance between the best risk and it risk is minimal. This is formalized by $\widehat{R}(\widetilde{f}, D^N) - R(f^*)$ is minimal. In fact, this distance is due to two causes with different nature as explained in the equation:

$$\widehat{R}(\widetilde{f}, D^n) - R(f^*) = \underbrace{\widehat{R}(\widetilde{f}, D^n) - R(f)}_{Err_{est}} + \underbrace{R(f) - R(f^*)}_{Err_{app}}$$

This residue nature was discussed in [2]: These two terms Err_{app} and Err_{est}: approximation error and estimation, are of different natures. In order to evaluate them, we will use the considerations raised respectively statistics and the theory of approximation. The selection of a model from a collection of models $\mathscr{F}$ for which the risk is similar to that of the oracle will be obtained by minimizing a penalized criterion. The penalty used to penalize large models, to avoid over-adjustment. The optimal choice of penalty (according to statistical models considered) is a very active research topic in statistics. Very generally, over a model (the family of admissible functions) is complex, it is more flexible and can fit to the observed data and therefore more the bias is reduced. However, the variance part increases with the number of parameters to estimate and therefore with this complexity. The challenge, to minimize the quadratic risk thus defined is therefore to find a better compromise between bias and variance: for skew estimation such as ridge regression to reduce more favorable variance.

More sophisticated risk criteria are considered in a Bayesian context if a priori probabilities is known about the classes or the

misclassification costs. The simplest way to estimate unbiased forecast error is to calculate the empirical risk on an independent sample who did not participate in the model estimation. This needs to break out the sample into three parts called respectively learning $D_L^{n_1}$, validation $D_V^{n_2}$ and test $D_T^{n_3}$ with:

$$D^N = D_L^{n_1} \cup D_V^{n_2} \cup D_V^{n_3}$$

- $\widehat{R}(\widetilde{f}(D_L^{n_1}), D_L^{n_1})$: Empirical risk computed with a first part of data used in the construction of the predictive model.
- $\widehat{R}(\widetilde{f}(D_L^{n_1}), D_V^{n_2})$: Empirical risk computed with a second part of data, different from $D_L^{n_1}$, that serves the validation of the model constructed with the data set $D_L^{n_1}$.
- $\widehat{R}(\widetilde{f}(D_L^{n_1}), D_T^{n_3})$: Empirical risk computed with a third part of data for testing. Some contributions affirms that the testing and the validation can be seen as the same thing.

2.5 Optimal Regression Rule

The optimal rule or model is, by definition, one that minimizes the average risk R. To formalize the definition of the optimal model, we assume to have all possible regression models in the set $\mathscr{F}$. The detection of the optimal regression model, denoted $f^\star$, is done by the selection of the rule having the minimal risk. Mathematically, the minimal average risk is given by $R(f^\star) = \inf_{f\in\mathscr{F}} R(f)$. In order to do so, we must find $f^\star$ such that:

$$f^\star = \arg\min_{f\in\mathscr{F}} R(f) \tag{3}$$

Theorem 2.1. *Let $\eta : \mathscr{X} \to \mathscr{Y}$ a regression rule. If the condition $\eta(x) = \mathbb{E}(Y|X=x)$ verified, then: $R(\eta) = \inf_{f\in\mathscr{F}} R(f)$ is hold.*

Theorem 2.2. *Let $\mu : \mathscr{X} \to \mathscr{Y}$ a regression rule. If the condition $\mu(x) = \arg\min_{f\in\mathscr{F}} \mathbb{P}(Y=\mu(x)|X=x)$is verified, then: $R(\mu) = \inf_{f\in\mathscr{F}} R(f)$ is hold.*

3. Learning algorithm consistency

3.1 What is learning algorithm ?

Let $\mathscr{F}$ be a set of measurable functions or, we can also say a collection of models. A Learning algorithm is a function that builds a measurable function using the n observed data point in $(\mathscr{X}\times\mathscr{Y})^n$. It is given mathematically by:

$$\hat{f} : (\mathscr{X}\times\mathscr{Y})^n \to \mathscr{F} \tag{4}$$

The function (4) associates, for each data points collection D^n, a prediction rule f in $\mathscr{F}$.

3.2 Learning Algorithm Risk Average

We define the risk average of a learning algorithm by the average risks of all rules constructed by the algorithm $\hat{f}$. In other words, if the algorithm $\hat{f}$ builds the rule $f(D^{n_1})$ using the observations in D^{n_1} and $f(D^{n_2})$ based on data set D^{n_2}... until the rule $f(D^{n_p})$ with observations collected in D^{n_p}, then the risk average of $\hat{f}$ is given by the average of all risks done by all those rules build by $\hat{f}$. Formally: it is given by the expectation of all risks $\mathbb{E}[R(f(D^n))]$.

3.3 Consistency According to J.C. Stone:

A prediction algorithm is called **universally consistent** if, for any probability distribution P:

$$\lim_{n\to\infty} \mathbb{E}[R(f(D^n))] = \inf_{f\in\mathscr{F}} R(f) \tag{5}$$

Based on the definition (5), Charles J. Stone (1977) introduced, in [8] and [9], a theorem which concluded the consistency of an algorithm that checks conditions. Let suppose $\{\omega_{n,i}\}$ a positive weight family built based on observations $x_1,...,x_n \in \mathscr{X}$. Those weights must verify family complementarity formally translated by $\omega_{n,1} + \omega_{n,2} + ... + \omega_{n,n} = 1$.

Definition 3.1. We define the real regression rule $\widehat{\eta}_n(D^n)$ that takes values in $\mathscr{Y}$ ($\mathbb{R}$ for example) as the following:

$$\widehat{\eta}_n(D^n) : x \in \mathscr{X} \longrightarrow \sum_{i=1}^{n} \omega_{n,i}(x) y_i \tag{6}$$

Definition 3.2. We define the binary discrimination rule $f_{\widehat{\eta}_n}(D^n)$, with $S(.)$ is the sign of the argument, that takes values in $\mathscr{Y} = \{-1,+1\}$ as the following:

$$f_{\widehat{\eta}_n}(D^n) : x \in \mathscr{X} \longrightarrow S(\widehat{\eta}_n(D^n)(x)) \tag{7}$$

Theorem 3.1. *Let the following assumptions for all $n \in \mathbb{N}$ and any function $f \in \mathscr{F}(\mathscr{X}, \mathbb{R}^+)$ and $\mathbb{E}[f(X)] < \infty$:*

$$\exists c > 0, \mathbb{E}\left[\sum_{i=1}^{n} \omega_{n,i}(x) f(X)\right] \leq c\mathbb{E}[f(X)] \tag{8}$$

$$\forall a > 0, \mathbb{E}\left[\sum_{i=1}^{n} \omega_{n,i}(x) \mathbf{1}_{\{||X_i - X|| > a\}}\right] \to 0 \tag{9}$$

$$\mathbb{E}\left[\sum_{i=1}^{n} \omega_{n,i}^2(x)\right] \to 0 \tag{10}$$

Charles J. Stone affirms that if hypotheses (8)*,* (9) *and* (10) *are verified, then:*

- *If $\mathscr{Y} \subset \mathbb{R}$ and $l(y,y') = (y-y')^2$, then $\hat{\eta}$ is universally consistent.*
- *If $\mathscr{Y} = \{-1,1\}$ and $l(y,y') = \mathbf{1}_{(y=y')}$, then $f_{\hat{\eta}}$ is universally consistent.*

4. Theory of generalization

4.1 Consistency as Asymptotic Convergence

The theory of consistency gives the necessary and sufficient conditions for the convergence of the algorithm when the observations number increases. By convergence we mean that the obtained results, using the proposed algorithm, are the best possible. The sufficient and necessary conditions for convergence of the empirical risk minimization (ERM) principle are hold even for small sample size when the consistency conditions are verified. This section focuses on the main concept that defines the generalization details for ERM principle so-called Vapnik-Cervonenkis entropy. This is formally translated by the fact that the probability, when the observation number is very big, of the event $\sup_{f\in\mathscr{F}}[R(f)-\widehat{R}(f)]>\varepsilon$ is close to 0. The previous event means that the largest distance between the empirical risk and the true risk must to be less than some small value ε. That comes to write (11):

$$\lim_{n\to\infty} p\left(|R(f)-\widehat{R}(f)|>\varepsilon\right)=0 \tag{11}$$

The famous inequality developed by Chernoff (1952) allows the characterization of how well the empirical average can approximates the expected value. This formula is applied to study how well the empirical risk, computed on a sample of n data point, can be close to the true risk that we are interested in. Formally:

$$p\left(|R(f)-\widehat{R}(f)|>\varepsilon\right)\leq 2e^{-2n\varepsilon^2} \tag{12}$$

The Chernoff bound in (12) seems is enough to prove consistency when the function f is fixed and with a sufficient large data point number n. The Chernoff bound tell us that, for a fixed function f, that the deviation between the empirical risk and the true one is minimal with a quantified probability value. It is possible that, in some few unlucky cases, that the observed environment knows some perturbations and gives some misleading data. If the decision rule is build using only those data, then the ERM principle can go completely wrong.

Let suppose that the set $\mathscr{F}$ consists just of finitely many functions, that is $\mathscr{F}=\{f_1,f_2,...,f_m\}$. Each of the functions $f_i\in\mathscr{F}$ satisfies the standard law of large numbers in form of the Chernoff bound (12). Now we want to transform these statements about the individual functions f_i into a uniform law of large numbers. To this end, note that we can rewrite:

$$p\left(|R(f)-\widehat{R}(f)|\geq\varepsilon\right)\leq p\left(\bigcup_{i=1}^{m}[|R(f_i)-\widehat{R}(f_i)|\geq\varepsilon]\right) \tag{13}$$

$$\leq\sum_{i=1}^{m}p\left(|R(f_i)-\widehat{R}(f_i)|\geq\varepsilon\right) \tag{14}$$

$$\leq\sum_{i=1}^{m}2e^{-2n\varepsilon^2} \tag{15}$$

$$\leq 2.m.e^{-2n\varepsilon^2} \tag{16}$$

Let comment those calculations one by one. Suppose that we have m bad events $|R(f_i)-\widehat{R}(f_i)|\geq\varepsilon$. It is a bad event because the distance between the empirical risk $\widehat{R}(f_i)$ and the true risk that we are interested in $R(f_i)$ is greater than some tolerable small value ε: it means that the empirical risk does not really represent the unknown true risk. The first inequality comes from the fact that the probability of one event in the set of bad events is less than the probability of all event in the set combined by the OR logical operator. That is why the union operator is used. The second inequality is due to the union bound states that says the probability of a union of events is smaller or equal to the sum of the individual probabilities. According to the Chernoff assumption, the probability of each bad event is less than $2e^{-2n\varepsilon^2}$. Formally:

$$p\left(|R(f)-\widehat{R}(f)|\geq\varepsilon\right)\leq 2\mathbf{m}.e^{-2n\varepsilon^2} \tag{17}$$

The factor m represents the cardinal of the function space $\mathscr{F}$. If the function space $\mathscr{F}$ is fixed, this factor m can be regarded as a constant, and the term $2m\exp(-2n\varepsilon^2)$ still converges to 0 when $n\to\infty$. Hence, the empirical risk converges to 0 uniformly over $\mathscr{F}$ as $n\to\infty$. That is, we have proved that empirical risk minimization over a finite set $\mathscr{F}$ of functions is consistent. For infinite cases, Vapnik and Chervonenkis introduce the so-called the ghost sample to reduce the case of an infinite function class to the case of a finite one. It consists of introducing the shattering coefficient discussed in the next subsection. It will enable us to replace the factor m in (17) by more general capacity measures that can be computed for infinite function classes.

4.2 The shattering coefficient

The shattering coefficient, denoted $N_{\mathscr{F}}(D^n)$, is the number of functions in $\mathscr{F}$ that can classify the sample D^n in different ways. In other words, the cardinality of $\mathscr{F}$ when it is restricted to the sample D^n. This coefficient characterizes the **diversity** of this set of functions on this sample.

The quantity $N_{\mathscr{F}}(D^n)$ is referred to as the shattering coefficient of the function class $\mathscr{F}$ with respect to sample size n. It has a particularly simple interpretation: it is the number of different outputs $(Y_1,...Y_n)$ that the functions in $\mathscr{F}$ can achieve on samples of a given size n. In other words, it measures the number of ways that the function space can separate the patterns

into two classes. In other words, shattering means that there exists a sample of n patterns which can be separated in all possible ways. The following paragraphs show how the shattering coefficient are used to find a generalization bound for empirical risk minimization on infinite function classes $\mathscr{F}$.

Let suppose an arbitrary function class which may be infinite, we now want to evaluate the right hand side of (17). Given a sample of n data points, arranged in the set D^n, where n points as the ghost sample. The goal is to replace the best learning rule over $\mathscr{F}$ by the best rule over D^n denoted F_{D^n}. This one contains at most $N_{\mathscr{F}}(D^n) \leq 2^n$ different functions, then apply the union bound on this finite function set: $p(|R(f) - \widehat{R}(f)| \geq \varepsilon) \leq 2.N_{\mathscr{F}}(D^n).e^{-2n\varepsilon^2}$. This leads to replace the m with the highest value of $N_{\mathscr{F}}(D^n)$ which is 2^n and that gives:

$$p\left(|R(f) - \widehat{R}(f)| \geq \varepsilon\right) \leq 2.N_{\mathscr{F}}(D^n).e^{-2n\varepsilon^2} \leq 2.2^n.e^{-2n\varepsilon^2} \quad (18)$$

Now we can use the expression (18) to draw conclusions about consistency of empirical risk minimization. Namely, ERM is consistent for function class $\mathscr{F}$ if the right hand side of this expression converges to 0 as $n \to \infty$.

4.3 Entropy and Growth function

Based on works of Vapnik (1999), the entropy describes the diversity of the set of functions on the given data. This quantity is a random variable since it was constructed with random i.i.d. data point. Let consider the next expectation form over the joint distribution function $p(D^n)$:

$$\mathscr{H}(\mathscr{F}, n) = \mathbb{E}(\ln N_{\mathscr{F}}(D^n)) \quad (19)$$

Vladimir N. Vapnik discussed in [5] the main result of the theory of consistency for the pattern recognition problem. He introduced the entropy condition for convergence. See the theorem :

Theorem 4.1. *For uniform two-sided convergence of the frequencies to their probabilities* $\lim_{n\to\infty} p\left(\sup_{f\in\mathscr{F}}[R(f) - \widehat{R}(f)] > \varepsilon\right) = 0$*, the necessary and sufficient condition* $\lim_{n\to\infty} \frac{\mathscr{H}(\mathscr{F},n)}{n} = 0$ *must be hold.*

Describing the necessary and sufficient condition for consistency of the ERM principle. This equation is the first milestone in learning theory: any machine minimizing empirical risk should satisfy it. However, this equation says nothing about the rate of convergence of obtained risks to the minimal one. The question now is: **Under which conditions is the asymptotic rate of convergence fast ?** That come to define the **growth function**.

Now we consider a new function, also based on $N_{\mathscr{F}}(D^n)$, named the **growth function**. It is formally given by:

$$\mathscr{G}(\mathscr{F}, n) = \ln \sup_{D^n} N_{\mathscr{F}}(D^n) \quad (20)$$

The equation (21) gives the necessary and sufficient conditions for **consistency** of ERM for any probability measure and also a sufficient condition for **fast convergence**:

$$\lim_{n\to\infty} \frac{\mathscr{G}(\mathscr{F}, n)}{n} = 0 \quad (21)$$

It describes the conditions under which the learning machine implementing ERM principle has an asymptotic high rate of convergence.

Let suppose an arbitrary, possibly infinite function class and a sample of $2n$ points, that is a set $N_{\mathscr{F}}(D^{2n})$. We interpret the first n points as the original sample and the second n points as the ghost sample. By fixing ε on a tolerable value and postulating the probability that the empirical risk deviates from the true risk by more than ε, we find how close we can expect the risk to the empirical risk. This can be achieved by setting the right hand side of (18) equal to some $\gamma > 0$. Then the statement that with a probability at least $1 - \gamma$, any function $f \in \mathscr{F}$ satisfies:

$$R(f) \leq \widehat{R}(f) + \sqrt{\frac{4}{n}.(\log 2N_{\mathscr{F}}(D^n)) - \log \gamma} \quad (22)$$

5. Local Average Algorithms Overview

The previous two theorems in subsections 2.1 and 2.2 introduce the optimal regression rules in the sense that it minimizes the average risk. This formalization assumes the knowledge of the probability density P_X because of the fact that the expectation is calculated based on this law. In an empirical case, the probability distribution is not always known. To this end, research will focus on the construction of forecasting algorithms Independent of the probability distribution P_X. The latter formalization consumes all training data D^n. We talk about **local average algorithms** and they are:

- The k-Nearest Neighborhood Algorithm (k-NN).
- Kernel based Algorithms.
- Algorithms by partitions.

5.1 k-Nearest Neighborhood Overview

Definition 5.1. We call the k-Nearest Neighborhood a local average algorithm whose weights verify $\omega_{n,i}(x) = \frac{1}{k_n}$ when x_i is included in the k nearest neighborhood of x and equals to 0 in the other case.

Theorem 5.1. *Let suppose* $\mathscr{X} = \mathbb{R}^d$ *and* $(k_n)_{n\geq 1}$ *a logical sequence. If* $k_n \to \infty$ *and* $k_n/n \to 0$ *then the algorithm of* k_n *nearest neighborhood is universally consistent for a given norm associated to* $\mathscr{X}$*.*

5.2 Kernel Algorithms Overview

Definition 5.2. Let suppose that K is a function, so-called the kernel function, having positive values and $(h_n)_{n\geq 1}$ a positive logical sequence. Some classical kernel functions over $\mathscr{X} = \mathbb{R}^d$ associated to the Euclidian norm $||.||$ are:

- Windows Kernel $K(x) = \mathbf{1}_{\{||x||\geq 1\}}$.

- Gaussian Kernel $K(x) = e^{-||x||^2}$.

We call a kernel algorithm, every locale average algorithm having weights that verify the condition in (23):

$$\omega_{n,i}(x) = K\left(\frac{x_i - x}{h_n}\right) / \sum_{j=1}^{n} K\left(\frac{x_j - x}{h_n}\right) \tag{23}$$

Theorem 5.2. *With $\mathscr{X} \in \mathbb{R}^d$ and $(h_n)_{n\geq 1}$. If $h_n \to 0$ and $n.h_n^d \to \infty$ then the kernel based algorithm* (23) *is universally consistent.*

5.3 Partition Algorithms Overview

Definition 5.3. Let suppose $V_n = (v_n^1, ..., v_n^k)$ a partition sequence having values in $\mathscr{X}$ and $v(x)$ the partition element containing $x \in \mathscr{X}$. For every part E in $\mathscr{X}$. The diameter of the part E is defined by $diam(E) = \sup_{x,y\in E} ||x - y||$. We call partition algorithm all local average algorithm having the form:

$$\omega_{n,i}(x) = \frac{\mathbf{1}_{\{x_i \in v(x)\}}}{Card(v(x))} \tag{24}$$

Theorem 5.3. *If $diam(V_n(x)) \longrightarrow 0$ and $\frac{Card(v(x))}{n} \longrightarrow 0$ then the partition algorithm defined with the coefficients $\omega_{n,i}$ is universally consistent.*

6. Learning Machines Overview

6.1 Perceptron Classifier Overview

In machine learning, the perceptron is an algorithm for supervised learning of binary classifiers: functions that can decide whether an input represented by a vector in $\mathbb{R}^d$ belong to one class or another. It is a type of linear classifier, i.e. a classification algorithm that makes its predictions based on a linear predictor function combining a set of weights with the feature vector. The algorithm allows for online learning, in that it processes elements in the training set one at a time. The perceptron algorithm dates back to the late 1950s. its first implementation, in custom hardware, was one of the first artificial neural networks to be produced.

The perceptron algorithm was invented in 1957 at the Cornell Aeronautical Laboratory by Frank Rosenblatt [10]. The perceptron was intended to be a machine, rather than a program, and while its first implementation was in software for the IBM 704. This machine was designed for image recognition: it had an array of, randomly connected, photocells. Weights were encoded in potentiometers and updated during learning process: it was done using electric motors. The classic perceptron algorithm is shown in 1:

In this section we present the consistency proof of the perceptron algorithm. The proof is based mainly on the theorem bellow:

Theorem 6.1. *Assume that there exists some parameter vector $\omega \in \mathbb{R}^d$ such that $||\omega|| = 1$. Let suppose some $\gamma > 0$ such that*

Input: D^n: Learning Data Set.
Output: ω: Updated Parameters Vector.

```
ω ← 0_{R^d}
for i = 1 to T do
    foreach (x, y) ∈ D^n do
        y' ← sign(x, ω)
        if y' ≠ y then
            ω ← ω + yx
        end
    end
end
return ω
```

Algorithm 1: Perceptron Algorithm

for all $(y,x) \in D^n$, we have $y(x,\omega) \geq \gamma$. Assume in addition that $||x|| \leq R$. The perception algorithm makes at most $\frac{D^2}{\gamma^2}$ errors. An error occurs whenever predicted label y' is different from observed label $y : (y' \neq y)$.

Proof. Let $\omega^{(k)}$ be the parameter vector when the algorithm makes it's k-th error. Note that we have $\omega^{(1)} = \mathbf{0}_{\mathbb{R}^d}$. The vector ω is changed with the equation $\omega^{(k+1)} = \omega^{(k)} + yx$ according to the perceptron definition. That comes to say that: $(\omega^{(k+1)}, \omega^{(1)}) = (\omega^{(k)} + yx, \omega^{(1)}) = (\omega^{(k)}, \omega^{(1)}) + (yx, \omega^{(1)})$. The expression $(yx, \omega^{(1)})$ can be written as $y(x, \omega^{(1)})$ because y is a constant. Based on assumptions on the theorem (6.1), the quantity $y(x, \omega^{(1)})$ is bounded by some γ. Formally we have: $(\omega^{(k+1)}, \omega^{(1)}) \geq (\omega^{(k)}, \omega^{(1)}) + \gamma$. By recurrence we find: $(\omega^{(k+1)}, \omega^{(1)}) \geq k\gamma$. In addition, using the Cauchy Schwarz inequality we conclude that $||\omega^{(k+1)}||.||\omega^{(1)}|| \geq (\omega^{(k+1)}, \omega^{(1)})$. With assumption $||\omega^{(1)}|| = 1$, we find $||\omega^{(k+1)}|| \geq (\omega^{(k+1)}, \omega^{(1)}) \geq k\gamma$. Finaly:

$$||\omega^{(k+1)}|| \geq k\gamma \tag{25}$$

In an other side we have:

$$\begin{aligned} ||\omega^{(k+1)}||^2 &= ||\omega^{(k)} + xy||^2 && (26)\\ &= ||\omega^{(k)}||^2 + ||yx||^2 + 2yx\omega^{(k)} && (27)\\ &= ||\omega^{(k)}||^2 + y||x||^2 + 2yx\omega^{(k)} && (28)\\ &\leq ||\omega^{(k)}||^2 + R^2 && (29)\\ & && (30) \end{aligned}$$

By induction we have:

$$||\omega^{(k+1)}||^2 \leq k.R^2 \tag{31}$$

By combining the two results in (25) and (31) we find that: $k^2.\gamma^2 \leq ||\omega^{(k+1)}||^2 \leq k.R^2$. Finally: $k \leq \frac{R^2}{\gamma^2}$ witch means that the perceptron algorithm errors number is bounded by $\frac{R^2}{\gamma^2}$.

6.2 Artificial Neural Networks Learning Overview

Vapnik (1999) affirms in [5] that the idea behind the Neural Network (NN) let us consider the method of minimizing the empirical risk $\widehat{R}(f)$. The use of regular gradient-based methods of optimization to minimize the empirical risk is impossible (The gradient of the indicator function is either null or is undefined). The solution is to approximate the set of indicator function by the so-called sigmoid functions. They are smooth and monotonic functions where there limit in $(-\infty)$ is equal to zero and in $(+\infty)$ is 1. Let give as examples the next functions: $s_1(x) = \frac{1}{1-\exp x}$ or $s_2(x) = \frac{2\arctan x+p}{2p}$... Let define the decision rule (regression or discrimination function) with:

$$f_w(x) = S\left(\sum_{i=1}^{n} w_i.x_i\right) \tag{32}$$

Where $S(.)$ is a sigmoid. The decision rule f_w is smooth in w then it has a gradient ∇ and therefore can be minimized using gradient-based methods. The gradient descent method uses the following update equation:

$$w_{n+1} = w_n - \gamma(n)\nabla f_w(x) \tag{33}$$

Where $\gamma(n)$ is a positive logical sequence that depends on the iteration. In order to assure the convergence of the gradient descent method to a local minimum, it is enough that $\gamma(1)+...+\gamma(n) = \infty$ and $\gamma(1)^2+...+\gamma(n)^2 \leq \infty$. A method for calculating the gradient of the empirical risk for the sigmoid approximation of NN's, called the back-propagation method described by S. HEYKING. Using this gradient descent method, one can determine the corresponding weights of all elements of the neural network.

6.3 Support Vector Machines Overview

Let suppose $\mathscr{X} = \mathbb{R}^p$ and $\mathscr{Y} = \{-1,+1\}$ and a training data set D^n. An hyperplane is represented by the equation $(w,x)-b = 0$. We say that set D^n of vectors is separated by the optimal hyperplane if all vectors in it are separated correctly and the distance between the nearest vectors to the hyperplane is maximal.

For a pattern recognition problem, the classification is done according to the value of the quantity $(w,x_i)-b$: the data x_i is classified $y_i = +1$ if $(w,x_i)-b \geq 1$ and $y_i = -1$ if $(w,x_i)-b \leq -1$. Data in the range $]-1,+1[$ are not classified by this model. Find the optimal hyperplane comes to an optimization problem: we have to minimize the quantity $\frac{1}{2}||w||^2$ under constraints $y_i[(w,x_i)-b] \geq 1$ for each i. This minimization is equivalent to the SRM principle. Based on (Minoux, 1989), the solution is given by the saddle point of the Lagrangian:

$$\mathscr{L}(w,b,\alpha) = \frac{1}{n}||w||^2 - \sum_{i=1}^{n} \alpha_i\left(y_i[(w,x_i)-b]-1\right) \tag{34}$$

Where the α_i are Lagrange multipliers. Lagrangian has to be minimized with respect to w,b and maximized with respect to $\alpha_i \geq 0$, formally: $\max_\alpha(\min_{w,b}\mathscr{L}(w,b,\alpha))$. The minimum if the Lagrangian respect to w and b is given by:

$$\left(\frac{\partial\mathscr{L}(w,b,\alpha)}{\partial w}, \frac{\partial\mathscr{L}(w,b,\alpha)}{\partial b}\right) = (0,0) \tag{35}$$

Rewriting these equations in explicit form one obtains the following properties of the optimal hyperplane: In the first hand, the coefficients α_i for the optimal hyperplane should satisfy the constraints $\sum_i^n \alpha_i y_i = 0$. In the second hand, the parameters of the optimal hyperplane w should be equals to $\sum_i^n x_i\alpha_i y_i$. Now let maximize (34) respect to α but considering the minimization result respect to w and b obtained by resolving (35). After simplifications, That comes to define the optimal coefficients as:

$$\alpha^* = \arg\max_\alpha \sum_{i=1}^{n}\alpha_i - \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i.\alpha_j.y_i.y_j.(x_i.x_j) \tag{36}$$

The case that the data are linearly nonseparable, we introduce nonnegative variables and we follow the same rezoning. Details are not included in this paper. On overview in [1] presents more technical clarifications.

6.4 Bayesian Learning Overview

In this sub-section we present the brief concentrated overview about Bayesian learning is extracted from [11]. Let suppose that there is a fixed unknown generative probability distribution $p(x,y)$ over the pair (X,Y). The goal of the prediction problem can be defined by the selection of a prediction function $f(x)$ having the lowest expected loss respect to a conditional probability distribution $p(x|y)$. Formally:

$$f(x) = \arg\min_{\widehat{y}} \int l(y,\widehat{y}).p(y|x)dy \tag{37}$$

Benjamin M. Marlin (2008) explained, in [11], that prediction frameworks differ in how they approximate the Bayes optimal prediction function as the following:

- Bayesian methods are closest in spirit to the Bayes optimal prediction rule and replace $p(y|x)$ in (37) with the posterior distribution over a set of models distributions.
- Maximum a posteriori approximations replace $p(y|x)$ in (37) with the single model distribution having the highest posterior probability among a given set of model distributions.
- The classical maximum likelihood principle replaces $p(y|x)$ in (37) with the single model distribution with the highest likelihood given the training data.

6.4.1 Bayesian Framework

The Bayesian prediction function given in Equation (38) with the true conditional distribution. Let call y^* the predicted output for the given input x:

$$f_{Bays} = \arg\min_{\widehat{y}} \int l(y,\widehat{y}).p(y|x,D^n)dy \tag{38}$$

The Bayesian approximation plan relies on the ability to analytically compute the integrals in Equations (38) and (37). In practice, the applications of the Bayesian approach rely on an additional layer of approximations provided by Markov chain Monte Carlo methods. Monte Carlo methods compute integrals and expectations by transforming them to sums over a big finite number of sample points. In Markov chain Monte Carlo methods, the sample points are generated by Markov chain methods like the Metropolis Hastings algorithm and the Gibbs sampler. Suppose, for the moment, that we have a method for generating independent samples θ_k. The Monte Carlo Markov Chains approximation for the quantity $p(y|x,D^n)$ is given by:

$$p(y|x,D^n) \approx \frac{1}{K}\sum_{k=1}^{K} p(y|x,\theta_k) \tag{39}$$

6.4.2 Maximum a Posteriori Framework

The maximum a posteriori (MAP) approach to the prediction problem is based on selecting the single distribution with highest posterior probability from a set of probability distributions given the observations history D^n and a prior distribution $p(\theta)$. Let suppose that we have a family of distributions indexed by a parameter θ such that each distribution in the family has the form $p(y|x,\theta)$ with prior probability $p(\theta)$. The posterior distribution of θ is again found using Bayes rule:

$$p(\theta|D^n) = \frac{p(\theta)}{p(D^n)}\prod_{i=1}^{n} p(y_i|x_i,\theta) \tag{40}$$

Using the total probability theorem, the probability $p(D^n)$ is given by: $p(D^n) = \int p(\theta)\prod_{i=1}^{n} p(y_i|x_i,\theta)d\theta$. Let θ_{MAP} be the parameter maximizing the posterior probability. Note that, it is possible to many θ that maximizes the posterior probability. It is given by:

$$\theta_{MAP} = \arg\max_{\theta} p(\theta|D^n) \tag{41}$$

The computation of the maximum a posteriori parameters θ_{MAP} comes to an optimization problem of the posterior probability. Let suppose that all first order partial derivatives, respect to the parameters θ_k, of the posterior distribution exists. The optimum is obtained by solving the following gradient equations: $\nabla p(\theta|D^n) = (0,...,0)$. The maximum a posteriori prediction function is obtained by substituting the single distribution $p(y|x)$ in (38) with $p(y|x,\theta_{MAP})$. The The maximum a posterior prediction function is:

$$f_{MAP}(x) = \arg\min_{\widehat{y}} \int l(y,\widehat{y}).p(y|x,\theta_{MAP})dy \tag{42}$$

6.4.3 Maximum Likelihood Framework

Let θ_{ML} be the parameter maximizing the likelihood between the observed data and the distribution probability. Note that, it is possible to many θ maximizes also the likelihood. Some approaches talk about log-likelihood because the addition of the logarithmic function makes it concave and facilitates the optimization process. It is given formally by:

$$\theta_{ML} = \arg\max_{\theta} \log p(D^n|\theta) \tag{43}$$

The computation of the maximum likelihood parameters θ_{ML} comes to an optimization problem by finding the extremum points. The ML predictor is given by:

$$f_{ML}(x) = \arg\min_{\widehat{y}} \int l(y,\widehat{y}).p(y|x,\theta_{ML})dy \tag{44}$$

6.4.4 Expectation Maximization Algorithm

The Expectation Maximization algorithm (EM), introduced by Dempster and al in [12], is an iterative numerical procedure that estimates the maximum a posteriori (MAP) or (ML) parameters. This algorithm is applicable when the probability is obtained by integrating over an unobserved variables Z. The EM algorithm starts by initializing the parameters vector to θ_0 by random vector. On each iteration t, the posterior probability of the missing variables Z is computed given the values of the observed variable X and introduced variable z and the current parameters θ_t. In the maximization step, θ_{t+1} is set to the value which maximizes the expected complete log posterior. These two updates are iterated until the posterior converges:

- **E-Step:** $q_{t+1} \leftarrow \mathbb{E}_{Z|X,\theta_t}(\log p(X,z|\theta_t))$
- **M-Step:** $\theta_{t+1} = \arg\max_{\theta} q_{t+1}$

7. Conclusion

We have proposed in this work a brief statistical learning overview for predictive analytic. The main goal of this contribution is to expose basic theoretical foundation of learning theory to Big Data researchers in order to think deeply and propose new learning algorithms that can go with Big Data constraints.

Next work version of this paper will incorporate the application field of learning machines and will more developed. Some performance analysis and comparison will be added to help the choice according to the applications aspect.

The hope is that this very fast growing area of research will contributes to the development of all branches of big data analysis and build new features to make human life easier.

Acknowledgments

We are grateful to anonymous reviewers for their constructive comments which improved significantly the quality of the article.

References

[1] EARLEY, Seth. Analytics, Machine Learning, and the Internet of Things. *IT Professional*, 2015, no 1, p. 10-13.

[2] O'LEARY, Daniel E. Artificial intelligence and big data. *IEEE Intelligent Systems*, 2013, no 2, p. 96-99.

[3] LIU, Bingwei, BLASCH, Erik, CHEN, Yu, et al. Scalable Sentiment Classification for Big Data Analysis Using Naive Bayes Classifier. In : *Big Data, 2013 IEEE International Conference on*. IEEE, 2013. p. 99-104.

[4] SLAVAKIS, Konstantinos, KIM, Seung-Jun, MATEOS, Gonzalo, et al. Stochastic Approximation vis-a-vis Online Learning for Big Data Analytics [Lecture Notes]. *Signal Processing Magazine, IEEE*, 2014, vol. 31, no 6, p. 124-129.

[5] VAPNIK, Vladimir N. An overview of statistical learning theory. *Neural Networks, IEEE Transactions on*, 1999, vol. 10, no 5, p. 988-999.

[6] JAIN, Anil K., DUIN, Robert PW, et MAO, Jianchang. Statistical pattern recognition: A review. P*attern Analysis and Machine Intelligence, IEEE Transactions on*, 2000, vol. 22, no 1, p. 4-37.

[7] COVER, Thomas M. et HART, Peter E. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 1967, vol. 13, no 1, p. 21-27.

[8] STONE, Mervyn. An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion. *Journal of the Royal Statistical Society*. Series B (Methodological), 1977, p. 44-47.

[9] STONE, Charles J. Consistent nonparametric regression. *The annals of statistics*, 1977, p. 595-620.

[10] ROSENBLATT, Frank. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 1958, vol. 65, no 6, p. 386.

[11] MARLIN, Benjamin M. *Missing data problems in machine learning*. 2008. Thèse de doctorat. University of Toronto.

[12] DEMPSTER, Arthur P., LAIRD, Nan M., et RUBIN, Donald B. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, 1977, p. 1-38.