Online learning applied to Autonomous Valuation of Financial Assets

Marcel Soares Ribeiro and Paulo Andre Lima de Castro
Autonomous Computational Systems Lab
Technological Institute of Aeronautics (ITA)
Sao Jose dos Campos-SP, Brazil
pauloac@ita.br, marcelsoaresribeiro@gmail.com

***Abstract*—In the context of Artificial Intelligence, Online learning is focused on environments that are not independent and identically distributed, i.e. the environment may changes its behavior as time goes by. Blum proposed a famous algorithm to this problem, which was called randomized weighted majority algorithm. In this paper, we propose an adaptation of such algorithm to autonomous valuation of financial assets. Our approach is based on learning from expert's advices, in order to create a more adaptable solution and reuse some results achieved for other researchers. We also briefly review some papers in the field. The proposed approach is materialized through an online learning algorithm that defines an analysis derived from many different analyses performed by autonomous analysts. Such analysts may be created using techniques from finance or machine learning fields. Our algorithm is able to take into account different costs of analysis errors. We believe that this skill in fundamental to an efficient analyst. We implemented the algorithm and tested it using several different techniques from finance and one (very simple) algorithm from machine learning area. This implementation was tested and the achieved results are analyzed and discussed. Furthermore, we proved that our algorithm's cost of error is limited by an expression of the cost of error of the best autonomous analyst. We believe that this algorithm may contribute to development of better systems that intend to estimate the price of financial assets in an autonomous way.**

I. Introduction

Through a review of relevant papers in the big field of autonomous investment analysis (see [1], for a short list) is possible to notice that many of the papers on the subject use historical time series of price and/or volume to motivate inferences about investment decisions. This usage of historical time series to predict future prices is controversial. Despite that this practice, usually known as Technical Analysis, is also largely adopted by analysts at least as part of the a more complex analytic process that also includes economic and market information such as profit, market share, EBITDA, price / profit relation and so on. The methods that utilize information of companies, the market and/or economic are commonly classified as fundamentalists. Actually, there are many articles that feature trading algorithms based on **technical or fundamentalist information** and on some artificial intelligence technique.

We can also notice that these algorithms feature some parameters. However, there is not a good understanding of how the value of each parameter affects the performance of algorithm and how changing the value of one parameter affects the configuration of other parameters. This makes it very difficult to define these values even for a small set of parameters.

The environment faced by the autonomous investment analyst may be classified as: partially observable, sequential, stochastic, dynamic, continuous and multiagent, according to the taxonomy of Russel and Norvig [2], which is the most complex class pointed out by them. However, it does not represent the entire problem's complexity. More than stochastic, this environment is a non-stationary process (the probability distribution changes over time) and is strategic in the sense that two investors compete for a more precise evaluation of the assets and their actions may change the behavior of other agents. Besides, the financial markets are **non-stationary environments**, i.e. the probability distributions may change over the time. Hence, a particular algorithm may have a great performance in some period of time but a terrible performance in the next one. In addition, different assets may demand different information and different algorithms, e.g. the oil companies are very sensitive to changes in the price of gasoline, but the same cannot be said about the banks.

In addition to the aforementioned questions: definition of relevant information, non-stationary processes and different character of the assets, we can observe different requirements with respect to another dimension: the investment horizon. We employ this term to refer to the period of time in which the investor intends to maintain their resources invested in the same set of assets. It ranges from many years to a few milliseconds. This wide range leads to algorithms that may be very efficient in short horizons but perform badly in the long run. Another aspect that have to be taken into consideration by any autonomous investment analyst (AIA) is that people do not have the same investment preferences, e.g. some investors may show a much greater risk-aversion than others may. An AIA must be aware of the preferences of its investor in order to provide adequate advice.

In short, it is possible to establish some important characteristics about the AIA environment: the financial market does not have a stationary probability distribution function but it changes slowly. So the optimal calibration of parameters tends to change slowly over time. Besides that, the number of probability distribution functions is finite and these functions repeats themselves from time to time. There are several pitfalls in overfitting machine learning models in the financial domain, as pointed out by Prado [3]. Overfitting must be avoided by keeping the training limited in terms of iterations or computational time and re-training must take place as new data is observed. . It consists of a problem of the *on-line* learning problems class [2]. For a more in depth discussion about *on-line* learning see [4].

II. Modeling the Autonomous Investment Analysis as a Classification Problem

The problem of carrying out the analysis may be simplified to a classification by means of the discretization of the returns. As pointed out in the Modern Portfolio Theory, trying to predict returns has the advantage of being dimensionless and dealing with narrower domain. The return is lower bounded by -1 and upper bounded by the impossibility of infinity return, but it is frequently less than 1 given reasonable horizons. This is simpler than the direct prediction of the asset prices, which may vary from some tenths to several thousands of some monetary unity. The current price is always known, which makes it trivial to calculate the future price given the return. The discretization of the returns makes close values of return indistinguishable, which is compatible with the general preferences of an investor. If the analyst signals a return of 85% and in reality it turns out to be of 84,9% or 85,1%, the investor will probably interpret as a correct analysis, not as a mistake. Besides that, small variations in the predicted return would hardly impact the investment decision, being thus unimportant to the performance of the investment. On the other hand, modeling the the autonomous analysis as a classification problem allows the usage of various known and tested techniques of Machine Learning, as well as facilitating the understanding of the problem. It is also interesting to notice that the degree of simplification can be controlled, since it is always possible to arbitrate the number of classes employed in the discretization, from the simplest possible case with two classes: positive and negative return up to cases with a high number of classes. In fact, since the very real numbers are always discretized to be represented in the finite memory of a computer, by means of the floating-point notation, it is possible to assert that the predicted return will always be discrete, even when the modeling approach is continuous.

Besides the discretization of the return, another considerable simplification of the autonomous analysis' work is the discretization of the investment horizon. It is assumed that the investment decisions will be different for different horizons, so it is assumed that the is an specific system to analyze each one of the possible horizons, which by simplicity will be restricted to three possible values: one day (D), one week (W) and one month (M).

Another fundamental characteristic to the modeling is the specification of the **Analyst's Memory**. This is the time period for which the analyst is capable of storing information it considers relevant to its decision process, be it prices, trading volumes or any other. Which information are relevant change from analyst to another, but the maximum time for which the information are retained is kept the same. For instance, if the current time is t and the analyst has memory m and stores data about past prices and volumes, this means that it may use in its decision process the prices $p_{t-m}, ..., p_{t-1}, p_t$ and volumes $v_{t-m}, ..., v_{t-1}, v_t$).

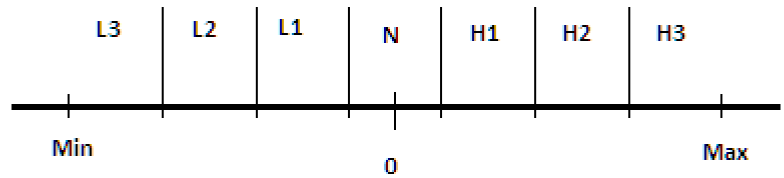

Fig. 1. Example of discretization with five classes.

A. Discretization Approach

The discretization of the return can be done in various ways. Our approach adopts some premises: one class identifies small alterations in the current price (neutral class or N), it is centered on zero with a small interval and there is symmetry in the number of classes above (H) and below (B) the neutral class. The classes that indicate positive return $(H_1, H_2, ...H_n)$ and the classes that indicate negative return $(L_n, ...L_2, L_1)$ need not have the same dimension due to the possible difference in the positive and negative domain, which is limited to -1, provided that assets with price limited to zero are considered.

$$Classes = \left[\begin{array}{ccccccc} L_n, & ....L_2, & L_1, & N, & H_1, & H_2, & ... & H_n \end{array}\right]$$

Thus, our approach presupposes establishing a lower bound (negative) and an upper bound for the return in a given investment horizon and dividing the intervals between the classes, considering the neutral class divided among the two intervals. In the example of the figure 1, which has seven classes, the interval of the classes L is given by —Min—/(3+1/2), while for the classes H it would be given by Max/(3+1/2).

Once established the classes, it is easy to observe that there will be errors perceived by an investor as much graver than others are. For example, if the analyst points to a return H2 for a given asset when the correct return would be H3 would be much less alarming than if he had predicted L2. In another words: the cost of error is not uniform. It is possible to address this issue with a cost of error matrix, select in a way that it represents the preferences of the investor. This is further discussed in the next session.

B. Cost of Error

As already mentioned previously, an error may have a cost significantly different to the cost of another error in the context of autonomous analysis. This way, it would be inadequate to evaluate an analyst only by its accuracy rate. A more suitable form would be to evaluate by the accuracy rate weighted by the cost of error. Since the objective of an analyst is not to obtain return, but to make correct predictions, it is reasonable that its performance be measured by the accuracy rate, but also taking the cost of error into consideration. For this purpose, we will employ a cost of error matrix suited to the investment preferences/profile of the investor. An uniform cost of error matrix for five classes would have the format below, where the **columns** represent the **real classes** and the **lines** represent the **predicted classes**. The error rate can be calculated as the number of errors divided by the total number of classifications, with the accuracy rate being the complement.

However, when the cost of error is not necessarily the same, it is necessary to employ a cost of error matrix adjusted by the gravity of the error, such as seen below. Where $c_{1,2}$ is the attributed cost to the error made by the analyst of predicting the class L2 (1) when the correct would have been L1 (2). That is, the cost of error of predicting the class i when the real class would be j is represented by $c_{i,j}$, a positive number for any i different from j. That defines an adjusted cost of error matrix C, as presented below.

$$\begin{bmatrix} L2 & L1 & N & H1 & H2 & \\ 0 & c_{1,2} & c_{1,3} & c_{1,4} & c_{1,5} & L2 \\ c_{2,1} & 0 & c_{2,3} & c_{2,4} & c_{2,5} & L1 \\ c_{3,1} & c_{3,2} & 0 & c_{3,4} & c_{3,5} & N \\ c_{4,1} & c_{4,2} & c_{4,3} & 0 & c_{4,5} & H1 \\ c_{5,1} & c_{5,2} & c_{5,3} & c_{5,4} & 0 & H2 \end{bmatrix}$$

It is then possible to define the classification distribution (hits and misses) of any analyst as a NxN matrix (N is the number of classes) with the format seen below, still for the five classes example. $d_{i,j}$ corresponds to the number of times the classifier predicted i and the real class was j.

With the classifier's distribution matrix and the adjusted cost matrix, it is possible to determine the expression for the average cost of error of this classifier (eq. 1).

$$AverageClassifierError = \frac{1}{\tau} * \frac{\sum_i \sum_j c_{i,j} * d_{i,j}}{c_{max}} \quad (1)$$

Where τ corresponds to the total number of realized classifications, c_{max} represents the greatest value of $c_{i,j}$ for any i and j, $c_{i,j}$ e $d_{i,j}$ correspond to the elements of the cost matrix and the classifier's distribution matrix, respectively. In the same manner, it is possible to define an accuracy rate adjusted by the cost of error, equation 2. It is easy to verify that this rate belongs to the [0, 1] interval, being equal to one when the classifier has not made any errors and equal to zero when the classifier has only made errors with the worst possible cost in all the instances. It can also be noticed that the greater the value of the rate, the better the performance of the classifier, accounting for the cost of its errors.

$$AdjustedAccuracyRate = 1 - \frac{1}{\tau} * \frac{\sum_i \sum_j c_{i,j} * d_{i,j}}{c_{max}} \quad (2)$$

The 2 provides us with a simple and objective way to evaluate the performance of an autonomous analyst and judge which one is better suited for a given market situation. However, as noted in section I, the investment analysis is a non-stationary problem, so the agent better adapted or with better performance in a given instant may quickly become unsuited. In the same manner, another agent with a poor performance may become one of the best in a way that is hard to predict beforehand. A way to tackle this challenge would be to have many autonomous analysts operating simultaneously and select the decision of one of them (or the aggregate of a small subset) as the final decision of the system. This is precisely the problem addressed in on-line learning, the prediction based on experts' advice. The next session deals with the theme and its adaptation to AIA.

III. On-line Learning: Predicting based on experts' advice

The problem of making predictions based on the advice of many experts is better explained by means of simple and intuitive example. A learning algorithm should predict if it is going to rain in a given day or not. The algorithm receives as input the advice of n experts. Each expert predicts yes or no and the learning algorithm has to use this information to make its own prediction. No other information is given to the algorithm besides the yes/no predictions produced by the expert. The algorithm receives the information if it has or has not rained after making its own prediction. Suppose there is no previous knowledge as to the quality or independence of the experts. A reasonable objective would then be to perform as well as the better expert to that moment. Naturally, which is the best expert changes over time. In the context of autonomous analysis, the issue is similar. Even though more complex, since the predictions would be over a real valued function (return) or of many values, if we employ the discretization proposed in section II. In the context of AI, this kind of situation is commonly referred as *on-line learning*.

Blum [4] proposed two algorithms to solve the problem with on-line learning. The first is quite simple and direct. It initially establishes equal weights to each expert and halves the weight of any expert that makes a mistake. The chosen output is that with the biggest weight. Blum has shown that the errors made by this algorithm are never greater than 2,41 (m + log n), where m is the number of errors made by the best expert and n is the number of experts. The second algorithm is an improved version of the first. Blum has shown that the number of errors of the second algorithm (called Randomized Weighted Majority Algorithm) is limited to (m*ln(1/β) + ln(n))/(1 -β), where β is a parameter of the algorithm [4].

Both methods implicitly assume that the cost of error is always the same. In another words, they employ a standard cost matrix, as show in section II-B. However, in the context of autonomous analysis, this supposition is not acceptable, as discussed in section II-B, the algorithm should take into account that some errors are worse than other in AIA.

For this reason, we propose an adapted version of Blum's algorithm to deal with adjusted cost of error matrices that we call **Error Adjusted Randomized Weighted Majority**

Algorithm. Besides that, we prove that this algorithm has a cost of error (sum of all the cost of the errors made) limited to (m * ln (1 / β_{min}) + ln (n)) / (β_{min}), where β_{min} is 1 / (c_{max} + 1) and c_{max} refers to the greatest element of the adjusted cost matrix. In order to formalize the adapted version of the algorithm, we need to redefine β, as shown in equation 3.

$$\beta(i,j) = \frac{1}{c_{i,j}+1} \quad (3)$$

$c_{i,j}$ refers to the cost of predicting class i when the real class is j, as defined in the adjusted cost matrix. Since $c_{i,j}$ is greater than or equal to zero, $b_{i,j}$ belongs to (0, 1]. Using equation 3, we need to some small changes in Blum's original algorithm to get the Error Adjusted Randomized Weighted Majority Algorithm, shown in figure 2.

Theorem 1. In any sequence of experiments, the expected cost of error M made by the Error Adjusted Randomized Weighted Majority Algorithm (EARWMA) satisfies the following:

$$M \leq \frac{m * ln(1/\beta_{min}) + ln(n)}{\beta_{min}} \quad (4)$$

Where M is defined as the sum of costs of all the errors made by the algorithm and m is the sum of the cost of all the errors of the best expert so far. The proof is similar to the original proof of the Randomized Weighted Majority Algorithm [4] and is given below.

Proof of Theorem 1. Let $F_{i,j}$ be the fraction of the total weight of the wrong answer j in the i^{th} test. Let τ be the number of examples. We define M as the expected sum of the cost of all errors made up to this moment, i.e. $M = \sum_{x=1}^{\tau}\sum_{j=1}^{E} F_{x,j} * c_{x,j}$, where E is the number of classes.

In the i^{th} iteration, the total weight W changes by:

$$W \leftarrow W(1 - \sum_{j=1}^{E}(F_{i,j} - \beta_{i,j}F_{i,j})) \quad (5)$$

So, the final weight after τ iterations is:

$$W = n\prod_{i=1}^{\tau}(1 - \sum_{j=1}^{E}(F_{i,j} - \beta_{i,j}F_{i,j})) \quad (6)$$

Using the fact that the total weight must be greater than or equal to the weight of the best expert b and letting m be the number of errors of the best expert, we have that:

$$n\prod_{i=1}^{\tau}(1 - \sum_{j=1}^{E}(F_{i,j} - \beta_{i,j}F_{i,j})) \geq \prod_{i=1}^{m}\beta_{i,b} \quad (7)$$

Taking natural logarithms:

$$ln(n) + \sum_{i=1}^{t} ln(1 - \sum_{j=1}^{E}(F_{i,j} - \beta_{i,j}F_{i,j})) \geq \sum_{i=1}^{m} ln(\beta_{i,b}) \quad (8)$$

$$-ln(n) - \sum_{i=1}^{t} ln(1 - \sum_{j=1}^{E}(F_{i,j} - \beta_{i,j}F_{i,j})) \leq \sum_{i=1}^{m} ln(\frac{1}{\beta_{i,b}}) \quad (9)$$

Since -ln(1-x)¿x for all x, we can write:

$$-ln(n) + \sum_{i=1}^{t}\sum_{j=1}^{E}(F_{i,j}(1 - \beta_{i,j}))) \leq \sum_{i=1}^{m} ln(\frac{1}{\beta_{i,b}}) \quad (10)$$

Since $M = \sum_{x=1}^{t}\sum_{j=1}^{E} F_{x,j} * c_{x,j}$ and $\beta(i,j) = \frac{1}{c_{i,j}+1}$, then $(1 - \beta_{i,j}) = c_{i,j}/(1 + c_{i,j})$. Making the substitution in the last equation we have:

$$-ln(n) + \sum_{i=1}^{t}\sum_{j=1}^{E}\frac{F_{i,j} * c_{i,j}}{(1 + c_{i,j})})) \leq \sum_{i=1}^{m} ln(\frac{1}{\beta_{i,b}}) \quad (11)$$

Using the definition of M and taking c_{max} as the greatest of all $c_{i,j}$ for all i and j, we have:

$$-ln(n) + \frac{M}{c_{max}+1} \leq m * ln(c_{max}+1) \quad (12)$$

Solving for M and using the fact that $\beta_{min} = 1/(c_{max}+1)$, we may conclude that:

$$M \leq \frac{ln(n) + m * ln(1/\beta_{min})}{\beta_{min}} \quad (13)$$

This concludes the proof of Theorem 1.

A. Error Adjusted Randomized Weighted Majority Algorithm with Self-exclusion

The usage of advice from various experts may contribute significantly to make an autonomous investment system more adaptable to many situations while keeping the experts relatively simple. A characteristic of one of these techniques is that they point out moments for buying and selling when some market patterns are observed. In the absence of occurrence of these patterns, they solely indicate the maintenance of the current price, even though this indication not being a factor that should make the price maintenance scenario any more likely, only shows the lack of conviction of the expert about what will happen. The stochastic indicator [1] is an example of this kind of behavior. This indicator outputs a buy signal when the indicator is less than 20 and a sell signal when it is greater than 80. Between values 20 and 80 there is a considerable uncertainty as to which should be the path to follow by the indicator, so its indication should be taken with reservations or simply disregarded.

This way, it would be interesting to have experts that always indicated a prediction, but also informed the degree of certainty of it. Thus it would be possible to enhance the performance of the algorithm EARWMA excluding the the opinion of the experts with a low degree of confidence in their own prediction from the weighted majority. The algorithm EARWMA with Self-exclusion (EARWMA2) is presented in the figure 3.

The cost of error of the algorithm would then be altered to $M' = \sum_{x=1}^{\tau}\sum_{j=1}^{E} F'_{x,j} * c_{x,j}$, where $F'_{x,j}$ is the fraction of the weight with wrong answer j in test x, accounting only for the non-excluded experts. It should be noted that the updating

1. Initialize the weights $w_1, w_2, ...w_n$ to 1
2. Given a set of predictions $x_1, ...x_n$ of the respective experts,
return x_i with probability w_i/W, where W is the sum of all weights
3. Receive the correct answer L and penalize each mistaken expert multiplying its weight by $\beta_{i,j}$.
4. Go back to step 2.

Fig. 2. Error Adjusted Randomized Weighted Majority Algorithm (EARWMA)

1...
2. Given a set of predictions $x_1, ...x_n$ of the respective experts, if there are at least two predictions were the respective experts have a degree of self-certainty greater than or equal to δ, eliminate from the set S the predictions with self-certainty smaller than δ, return x_i with probability w_i/W, where W is the sum of all weights of the remaining experts
3....
4.....

Fig. 3. Error Adjusted Randomized Weighted Majority Algorithm (EARWMA) with Self-exclusion. Only step 2 changes from EARWMA.

of the weights is not altered, even though some experts are filtered; the weight update is done for them all. Let R_x and $R_x^{'}$ be the fractions of the weight with correct answer in the test x with the algorithms ARWMA and ARWMA2, respectively. It is easy to verify that the sum of the weights of the selected experts will be equal or less than the sum of the weights of all the experts, so it is possible to wright:

$$R_x^{'} + \sum_{j=1}^{E} F_{x,j}^{'} \leq R_x + \sum_{j=1}^{E} F_{x,j} \tag{14}$$

Assuming that the confident experts have higher or equal accuracy to the exclude experts, i.e. $R_x \leq R_x^{'}$, then:

$$R_x^{'} + \sum_{j=1}^{E} F_{x,j}^{'} \leq R_x^{'} + \sum_{j=1}^{E} F_{x,j} \tag{15}$$

So:

$$\sum_{j=1}^{E} F_{x,j}^{'} \leq \sum_{j=1}^{E} F_{x,j} \tag{16}$$

Multiplying both sides by cx, j and summing over all tests we have:

$$M^{'} \sum_{x=1}^{\tau} \sum_{j=1}^{E} F_{x,j}^{'} * c_{x,j} \leq \sum_{x=1}^{\tau} \sum_{j=1}^{E} F_{x,j} * c_{x,j} = M \tag{17}$$

Therefore, if the assumption that the confident experts are more accurate than the excluded ones is valid, the ARWMA2 should have a better performance than its previous version.

IV. Implementation and Results

One of the advantages of modeling the autonomous analysis of investments as a classification problem is the possibility of utilizing many already available algorithms and tools to deal with this class of problem. Among this tools, we chose the Weka [5] framework to carry out our implementation.

We employed five Technical Analysis based agents: Stochastic, RSI, MACD, MA and Price Oscillator [6] and two classifiers available in Weka adapted to be employed as autonomous analysts. This reuse of classifiers is an example of the advantages of modeling AIA as a classification problem. In Weka's specific case, this reuse is made possible by our creation of a class derived from the class *AbstractClassifier* that implements the methods *buildClassifier* and *classifyInstance*.

The data for training is generated by the price (open, close, high, low and mean) and volume time series with predefined analyst's memory (see section II) and stored in arff format. In the experiments presented in section IV, we use a memory of 40 days.

In this section, we present the performance data obtained by the autonomous analysts implemented, including the Error Adjusted Randomized Weighted Majority Algorithm (EARWMA). The performance is measured with the error adjusted accuracy rate (equation 2). The simulations were made with technical data from 2012 to 2016 with the eighteen most traded stocks in the Brazilian stock exchange (BOVESPA, recently renamed B3). The discretization was done with seven target classes (L3,L2,L1,N,H1,H2,H3). Therefore, a complete random classifier would have an expected accuracy rate of 14.2%. However, its error adjusted accuracy rate would be 36,97%. For that calculation we use equation 2 and assume a cost matrix where $c_{i,j} = (1,1)^a bs(i-j)$, if $ineq$j and $c_{i,j} = 0$, otherwise. That defines a symetric cost matrix with higher costs for bigger erros. For instance, Let be H2 the predicted class and N2 the actual class, that would cost $1, 1^6 \approx 1, 77$, while if the actual class were H1 the cost would be only 1,1. It is consistent with the preferences of any investor. It is interesting to note adjusted accuracy rate tends to be higher than the traditional accuracy rate, except if the classifier is biased to high cost errors.

In table I, we present the risk adjusted accuracy rate for each autonomous analyst in the evaluated period (2012-2016) for three horizons of investment: 1 day (D), 1 week (W) and 1 month(M). Taking as reference a random analyst which would present approximately 36,97% of risk adjusted accuracy rate, it is possible to notice that the analysts had a better performance, but there is much room for improvement. In

	J48			MA			MACD			PRIOSC			RSI			Stochastic			ZeroR			EARWMA				
Asset	D	W	M	D	W	M	D	W	M	D	W	M	D	W	M	D	W	M	D	W	M	D	W	M	Avg.	Std. Dev.
BBAS3	48,4	53,1	44,3	48,9	40,2	35,4	48,4	55,4	53,2	46,7	55,2	52,4	45,9	42,0	56,8	47,4	40,3	52,2	48,4	45,8	41,4	49,7	55,4	53,2	48,3	5,7
BBDC4	47,7	62,2	49,9	43,1	38,3	32,6	54,8	65,4	58,6	54,4	63,2	57,8	37,8	35,2	44,7	41,6	33,9	42,9	54,8	41,1	33,0	39,3	65,4	58,6	48,6	11,0
BVMF3	53,3	47,0	41,0	34,2	37,0	28,2	55,9	64,7	56,6	56,6	63,7	54,8	36,5	34,3	43,8	39,6	32,6	36,9	55,9	44,2	34,8	41,4	64,7	56,6	46,4	11,3
CIEL3	46,2	75,2	65,9	30,4	36,1	29,0	57,9	77,1	67,7	57,8	75,6	66,8	34,3	31,4	29,5	44,5	30,6	29,3	57,9	51,6	44,2	35,1	77,1	67,7	50,8	17,7
CMIG4	36,7	50,6	44,0	40,6	38,4	31,3	44,1	63,5	55,7	43,4	62,4	54,6	45,9	43,4	49,9	40,8	42,2	46,3	44,1	40,6	34,9	47,7	63,5	55,7	46,7	8,8
CSNA3	41,5	47,3	45,1	44,3	46,7	32,6	42,7	51,5	47,4	43,0	49,8	48,1	49,8	47,3	56,6	45,0	45,6	50,8	42,7	51,4	36,0	48,9	51,5	39,2	46,0	5,3
GGBR4	39,4	46,5	45,8	45,1	40,9	46,0	44,1	49,8	40,4	42,5	48,9	40,8	51,4	48,7	68,1	50,7	49,1	68,8	44,1	53,7	43,2	52,5	49,8	40,4	47,9	7,6
GOAU4	27,8	34,3	45,6	46,1	46,9	55,1	41,3	46,9	45,0	40,3	47,4	45,4	50,9	52,5	75,6	49,0	49,3	71,6	41,3	50,2	46,9	41,3	46,9	45,0	47,6	9,9
HYPE3	49,2	46,7	39,3	36,5	40,8	37,8	59,0	71,5	68,8	58,1	70,7	67,0	37,2	31,4	33,6	44,6	30,4	29,7	59,0	52,2	49,6	39,1	71,5	68,8	49,7	14,5
ITSA4	49,2	59,7	65,8	39,8	35,6	35,4	57,8	69,6	62,1	55,6	67,7	60,0	38,0	35,4	39,7	42,6	35,2	39,3	57,8	48,3	35,8	38,3	69,6	62,1	50,0	12,7
ITUB4	46,3	60,4	64,3	42,8	34,6	34,7	61,4	64,2	64,2	60,9	63,9	63,7	41,5	34,4	44,0	41,9	32,8	42,3	61,4	35,5	34,6	42,1	64,2	64,2	50,0	12,7
JBSS3	41,6	47,0	50,1	36,7	33,2	25,7	53,1	67,5	47,4	52,8	66,0	47,0	41,0	36,9	40,4	42,7	36,7	40,4	53,1	43,0	32,9	41,7	67,5	47,4	45,5	10,7
PETR3	41,1	56,5	61,0	47,5	49,2	47,0	48,2	61,0	56,7	47,0	58,6	54,9	43,6	39,5	51,9	46,4	37,6	50,9	48,2	42,7	48,0	44,8	61,0	56,7	50,0	6,9
PETR4	47,0	49,5	51,3	48,8	48,7	46,4	45,7	56,9	50,1	44,7	55,4	50,0	47,5	42,9	61,4	47,4	42,9	61,4	45,7	49,5	57,7	47,7	56,9	50,1	50,2	5,3
SUZB5	41,7	58,4	49,4	45,7	40,4	33,2	49,8	63,4	59,2	48,9	63,3	59,3	42,4	34,6	39,7	40,0	34,6	39,7	49,8	37,2	28,2	43,2	63,4	59,2	46,9	10,7
USIM5	33,5	42,0	64,8	47,8	51,7	44,4	37,5	47,9	48,1	35,4	47,0	48,8	48,0	48,5	68,6	38,5	44,8	58,9	50,8	51,0	42,8	46,2	47,9	48,1	47,7	8,3
VALE3	35,1	53,6	46,7	43,8	41,3	36,2	44,8	55,9	56,2	43,9	54,7	56,6	47,6	46,8	63,4	45,0	46,8	63,4	44,8	47,7	35,5	45,2	55,9	56,2	48,6	8,0
VALE5	37,0	44,9	46,8	49,0	45,2	43,5	46,1	52,6	51,4	45,8	50,7	51,8	47,1	51,0	68,8	47,4	47,7	66,8	46,1	46,3	47,4	46,9	52,6	51,4	49,4	6,8
Avg.	42,37	51,94	51,16	42,84	41,41	37,47	49,58	60,27	54,93	48,77	59,11	54,43	43,68	40,89	52,03	44,17	39,61	49,53	50,32	46,23	40,39	46,9	60,27	54,47	-	-
Std. Dev.	6,63	9,21	8,98	5,44	5,47	7,88	6,96	8,62	7,84	7,20	8,45	7,26	5,33	6,99	13,53	3,44	6,57	13,16	6,27	5,34	7,60	46,9	8,62	8,51	-	-

TABLE I

ADJUSTED ACCURACY RATE (%) PER ASSET AND HORIZONS 1 DAY (D), 1 WEEK (W) AND 1 MONTH(M) WITH AVERAGES AND STANDARD DEVIATION

order to compare analysts' performances, the table I presents the average and standard deviation of the Adjusted accuracy rate of each classifier in its last two rows. The two columns in the right present the average and standard deviation of the Adjusted accuracy rate per asset.

Analyzing the table I, it is possible to observe that there is little variation by horizon, but greater deviation for larger horizons. A sizable difference in performance is noticed between the analysts, from which tge best possible are MACD and EARWMA, but even for those there is still room for improvement, since the better accuracy rates were around 60%. It is also possible to notice that the EARWMA follows closely the better analyst (MACD) for larger horizons, which is compatible with the idea of the EARWMA indicator itself. Possibly, if other analysts with better performance were employed, the EARWMA's performance would also be higher. In table I is possible to notice that the variations in performance with respect to asset is relatively small.

V. CONCLUSIONS AND FUTURE WORKS

In this paper, we presented a discrete multiagent approach based on Online learning that uses information from several different analysis algorithm to produce an aggregate analysis. This approach was adopted based on some remarkable aspects of the autonomous analysis environment. Such aspects make the problem specially complex for autonomous agents and can be described are: stochastic environment, partially cooperative multiagent, partially observable and the fact that the environment is non-stationary. Our approach is based on the discretization of time and return. This way AIA can be treated as a classification problem. This simplifies the problem in a controlled way (we can reduce the discrete steps, if needed). It also makes easier to treat the fact the cost of errors may be significantly different between distinct errors. Furthermore, the non-stationary nature of the problem may cause some agents to have a good performance in a certain period of time and than be outperformed by other agents. This fact is similar to the one faced in the context of on-line learning, but the know algorithms to deal with on-line learning do not deal with situations in which the cost is non-uniform [4]. We developed an adapted version of the algorithm proposed by Blum to deal with non-uniform costs and prove that there is an upper bound for the cost of error of the new algorithm (section III). This constitutes one of the main contributions of the present work.

Another contribution is the discretization approach, along with its implementation and the tests employing the Weka [5] framework. The results obtained single out a substantial variation of performance between the analysts, with MACD and EARWMA showing good results, but we admit that there is still much room for improvement, since the better rates lay around 60%. The performance spread across assets is relatively small. We believe that a way to improve the system's performance is to introduce new analysis techniques, especially Fundamental Analysis based algorithms. This would probably enhance the performance of the Error Adjusted Randomized Weighted Majority Algorithm (EARWMA) proposed in this work, since this algorithm tends to follow the performance of the best analysts, as pointed out in section IV and confirmed by the obtained results. It is clear, however, that there is a long way to achieve an efficient autonomous investment analyst.

ACKNOWLEDGMENTS

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) -Finance Code 001 and it is also partially funded by CNPq. Grant 311838/17-0.

REFERENCES

[1] Paulo Andre Lima Castro and Jaime Simao Sichman. Automated asset management based on partially cooperative agents for a world of risks. *Applied Intelligence*, 38:210–225, 2013.

[2] Stuart Russell and Peter Norvig. *Artificial Intelligence A Modern Approach Third Edition.* Prentice Hall, Englewood Cliffs - NJ, 2013.

[3] Marcos Lopez de Prado. *Advances in Financial Machine Learning.* Wiley, New York, 2018.

[4] Avrim Blum. On-line algorithms in machine learning. *In Online Algorithms: The State of the Art, edited by A. Fiat and G. Woeginger*, 1442:306–325, 1998.

[5] Mark A. Hall Eibe Frank and Ian H. Witten. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, New York, 2016.

[6] Paulo Andre Lima de Castro and Ronnald Annoni Jr. Towards autonomous investment analysts - helping people to make good investment decisions. In *Proceeding of the Future Technologies Conference 2016*, san Francisco, CA, December 2016. FTC 2016.