RESEARCH PAPER | Economics | Volume : 4 | Issue : 5 | May 2014 | ISSN - 2249-555X

Experimental Evaluation of Online Learning Algorithms

KEYWORDS
algorithm, correlation criterion, criterion, data, error, machine learning, measurement, statistical test, value

Ana-Maria Ramona Stancu
Academy of Economic Studies, Bucharest-Romania,
Str. Podul Giurgiului, nr 10, ap.24, sector 5, Bucuresti,
Romania

Dominic Patricio Perez-Danielescu
Academy of Economic Studies, Bucharest-Romania,
Str. Av. Serban Petrescu, nr.16, sector 1, Bucuresti,
Romania

ABSTRACT *In this article I will start in the first part of the debate over the basic methods for performance evaluation of algorithms, i.e. starting with the evaluation criteria presentation online, after which I will describe the criteria that can be used for evaluation of online algorithms, after which I will describe the measurement errors. I mean, I will discuss the metrics used in Machine Learning, the average relative error, the relative root mean square of the error. In Part 3 I will discuss the methods used in the evaluation, after which I will discuss evaluation algorithms according to Friedman's test and Nemenyi*

1. CRITERIA FOR THE EVALUATION OF ONLINE ALGORITHMS

So far there are few studies on the learning algorithms. Among these, one includes STATLOG (King et al, 1995) which was very complete when it was done, but with the passage of time other algorithms have also emerged.

When we have large volume of data, the algorithms must be able to accomplish learning through a single stage (pass) as non-stationary data distribution is performed using data from achieved deviation of concepts and their exchanges.

One can say that the flow and online algorithms are built to track their evolution (data) and achieve models to indicate exactly the specific changes.

Issues that have been approached so far are totally unresponsive to Machine Learning theory, therefore, the characteristics of online learning techniques of analysis tools and methods specified shall treat temporal components of data.

Learning is a continuous process in which methodology approaches issues of online learning (Castillo and Gama).

Learning algorithms are used in different areas and levels of performance are appropriate to each area.

If we think that performance metrics accomplish various concessions (compromise) in the assumptions made in the classification, it may be that the learning methods to achieve this metric to be suboptimal in another measurement.

Therefore, it is better to evaluate the algorithms on a large set of performance indicators.

I will describe the learning methodology that applies online and I want to perform an evaluation methodology to enable monitoring the learning process in all its aspects in order to approach the problems of algorithms, the model development, the processing time, etc.

2. MEASUREMENT ERRORS

In this section I will discuss the indicators used in the generalization of the algorithm, the complexity of the models and efficiency of methods when non –stationary distributions appear.

The most used metric for a learning algorithm is the generalized error model which is an estimate of the model induced in relation to the target function, so generalization error estimates are obtained from the use of a set of tests.

In Machine Learning, metrics used are:

- the mean absolute error (mean absolute error - MAE) (Abramowitz and Stegun, 1972)
- mean square error (mean squared error - MSE) (Armstrong and Collopy, 1992)

If we have:

$\{(x_1,y_1),(x_2,y_2),\dots\} - \textit{data set}$

$f(x) - \textit{function}$

$\hat{g}(x) - \textit{estimator function } f(x)$

Then, we have the approximation formula: (given by MSE)

$$e(g) = \int [g(x) - f(x)]^2 * p(x)dx \qquad (4.1)$$

The integral can be approximated with a summation of a set of cases (size = N) used for testing purposes and developed separately.

I can say that MSE is defined as being the difference between the expected value and the corresponding value, ie:

- The expected value:
  $\hat{g}_i = \hat{g}(x_i)$ *(4.1.1)*
- The appropriate value:
  $y_i = f_i = f(x_i)$ *(4.1.2)*

Average relative error (relative mean squared error - RE) can be used instead of MSE as the scale depends on the magnitude of the function.

$$MSE = \frac{\Sigma(f_i - \hat{g}_i)^2}{N} \Rightarrow \sum (f_f - \hat{g}_i)^2 =$$

$$= MSE * N$$

(4.2)

But,

$$MSE = \frac{\Sigma(f_i - \hat{g}_i)^2}{\Sigma(f_i - \bar{f})^2}$$ *(4.2.1)*

So,

$$RE = \frac{MSE * N}{\Sigma(f_i - \bar{f})^2}$$ *(4.3)*

I know that,

$$\bar{f} = \frac{1}{N} * \sum_{i=1}^{N} f_i$$ *(4.4)*

where,

$$\bar{f} = media\ scorului\ pentru\ funcția\ f$$

Next, I will calculate the relative root mean square of the error (relative root mean squared error - RRSE)

$$RRSE = \sqrt{\frac{1}{N} * \sum \frac{f_i - g(x_i)}{(f_i - \bar{f})^2}}$$ *(4.5)*

I used:

1. The absolute mean value (Mean absolute error – MAE)

$$MAE = \frac{1}{N} * \sum_{i=1}^{N} |f_i - g(x_i)|$$ *(4.6)*

2. The relative mean of the absolute value (Relative mean absolute error – RMAE)

$$RMAE = \frac{N * MAE}{\Sigma|f_i - \bar{f}|}$$ (4.7)

The correlation coefficient of Pearson:

$$CCP = \frac{S_{f\hat{g}}}{S_f * S_{\hat{g}}}$$ *(4.8)*

$$S_{f\hat{g}} = \frac{\Sigma_{i=1}^{N}[(f_i - f) * (\hat{g}_i - \hat{g})]}{N - 1}$$

(4.8.1)

$$S_f = \frac{\Sigma_{i=1}^{N}(f_i - \bar{f})^2}{N - 1}$$ *(4.8.2)*

$$S_{\hat{g}} = \frac{\Sigma_{i=1}^{N}(\hat{g} - \bar{g})^2}{N - 1}$$ *(4.8.3)*

where:

$$\bar{f} = \frac{1}{N} * \sum_{i=1}^{N} f_i$$ *(4.8.4)*

$$\bar{\hat{g}} = \frac{1}{N} * \sum_{i=1}^{N} \hat{g}_i$$ *(4.8.5)*

The correlation coefficient ∈ [-1,1]

I notice that:

- learning algorithm maximizes the correlation coefficient
- mean square error and mean absolute error must be minimized
- the correlation coefficient of Pearson measures the linear correlation between two variables

We have:

$$\sum_{i=1}^{N}(f_i - \bar{f})^2 = \sum_{i=1}^{N}\left(f_i - \frac{1}{N} * \sum_{i=1}^{N} f_i\right)^2 =$$

$$= \sum_{i=1}^{N} f_i^2 + \frac{1}{N} * \left(\sum_{i=1}^{N} f_i\right)^2$$

(4.9)

As

$$\bar{f} = \frac{1}{N} * \sum_{i=1}^{N} f_i \qquad (4.9.1)$$

To evaluate an online algorithm we must estimate:

- Adaptive method of algorithm
- Performance of change detection

In case of probabilistic thinking, I am interested in:

1. The problem of false alarms - in a certain interval of time measured false
2. The issue of truth - is the method which detects all changes
3. Delay in detection

3. METHODS USED IN EVALUATION

There are two methods that correspond to the basic method used to evaluate the algorithm on line:

1. Regular assessment of learning model
   - Called also holdout
   - Measurements are estimated by means of a set that was not used in training
   - The best estimate on the performance of algorithms
   - Assessment is implemented using a data flow buffer on holding a set of training examples.

2. Sequential Evaluation

- The model is tested on an example that is used for training,
- At the beginning of the learning statistical errors can be made
- Either $L(f,\hat{g})$ and it indicates the loss function that evaluates the function induced, and we have:

$$S_i = \sum_{i=1}^{N} L(f,\hat{g}_i) \qquad (4.10)$$

I know that learning algorithms always updates and information at the beginning of learning is poor performance and then measured quantities give a pessimistic picture on algorithm performance.

4. Benchmarking

To evaluate the algorithm we must evaluate existing learning methods, thus, developed algorithm needs to be improved to the existing algorithm.

Statistical tests are applied regarding checking of performance assumptions:

1. Friedman's test (1937.1940)

- It was made by Milton Friedman
- It is used to find the differences in test trials
- It is used in multiple hypothesis testing
- It is a non-parametric test
- It is used to analyze tests categories
- It consists of algorithms and in each set of data - the algorithm achieves good results, ie rank 1 or 2

If r = rank *j* from *K* algorithm on the position i of the *N* sets of data

Either:

$$R_j = \frac{1}{N} * \sum_{i=1}^{N} r_i^j \qquad (4.11)$$

indicates the average of the algorithm for *j=1, …, k*

The statistical test is:

$$\chi_F^2 = \frac{12 * N}{k * (k+1)} * \left[\sum\nolimits_j R_j^2 - \frac{k * (k+1)^2}{4}\right]$$

(4.12)

Then:

$$\bar{r} = \frac{1}{n * k} * \sum_{i=1}^{n}\sum_{j=1}^{k} r_i^j \qquad (4.13)$$

$$SS_t = n * \sum_{j=1}^{k} (\bar{r}_j - \bar{r})^2 \qquad (4.14)$$

$$SS_e = \frac{1}{n * (k-1)} * \sum_{i=1}^{n} * \sum_{j=1}^{k} (\bar{r}_{ij} - \bar{r})^2$$

(4.15)

Then the statistical test is:

$$T = \frac{12}{n} * \frac{1}{k * (k+1)} *$$

$$* \sum R_j^2 - 3 * n * (k+1)$$

(4.16)

where:

K = the number of columns

n = the number of rows

Rj = sum rows of column j

In 1980, Davenport and Iman shows a statistical formula that is used to compare algorithms.

This is:

$$F_F = \frac{(N-1) * \chi_F^2}{N * (k-1) - \chi_F^2} \qquad (4.17)$$

and is called the correction D.I.

After the application of the test, if the result is relevant to the data obtained from the different distribution, then I need to use other tests in order to provide a reference algorithm, and this has to be better than the others.

If the hypothesis test is rejected, checking is continued by means of a test, and if comparing the mean rows of an algorithm is desired, then I must use Nemenyi test.

2. Test Nemenyi

If we have the ranking difference higher than the critical difference, then I calculate by:

$$AB = q_{\propto} * \sqrt{\frac{k * (k+1)}{6 * N}} \qquad (4.18)$$

where:

$q_{\propto} = valoare\ critică$

You can use the following statistical test to compare the algorithm *i* and algorithm *j*:

$$z = \frac{R_i - R_j}{\sqrt{\frac{k * (k+1)}{6 * N}}} \qquad (4.19)$$

where:

z = is used to find the probability that is found distribution in the normal distribution table

5. Conclusion

As described in this article, it can be said that the learning Algorithm maximizes the correlation coefficient.

The mean quadratic error and average absolute error need to be reduced to a minimum. Pearson's correlation coefficient measures the linear correlation between two variables. Friedman's test cannot be applied if it rejects the hypothesis and if you want to compare the average of rows from an algorithm, then you must use the Nemenyi test.

REFERENCE [1] Amir Bar – Or, Assaf Schuster, Ran Wolff, daniel Keren – Decision Tree Induction in high dimensional, hierarchically distributed databases | [2] Fernando Berzal, Juan-Carlos Cubero, Maria J. Martin – Bautista – "On the quest for easy-to-understand splitting rules" | [3] Rich Caruana, Alexandru niculescu Mizil – An empirical comparison of supervised learning algorithms | [4] Wray Buntine - Decision tree induction systems: a bayesian analysis | [5] http://www.codeproject.com/Articles/5276/ID3-Decision-Tree-Algorithm-in-C | [6] http://dms.irb.hr/tutorial/tut_dtrees.php | [7] http://vserver1.cscs.lsa.umich.edu/~spage/ONLINECOURSE/R4Decision.pdf |