International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2016): 79.57 | Impact Factor (2015): 6.391

Time Dependent Analysis of Machine Learning Algorithms

Akhil Sethia

[1]Dwarkadas J. Sanghvi College of Engineering, Bhaktivedanta Swami Marg, Mumbai 400056, India

Abstract: ***This study introduces a standardisation technique for existent performance metrics which make them time dependent and independent of the number of attributes and testing/training tuples in the dataset, thus enabling a comparison of various supervised methods across different datasets. In this study, this technique has been applied to the achieved Accuracy, F1 score, ROC and Cross Entropy. Ten distinct, supervised learning based, both balanced and unbalanced datasets have been chosen, and 10 different classification algorithms have been trained and tested on this dataset. The training/testing time, the standardised performance measures and the raw accuracy is then used to analyse each algorithm and its strength and weakness based on its accuracy v/s its train/test timing. The suitability of algorithms to real-time systems has been evaluated and optimal algorithms in different time dependent scenarios are outlined.***

Keywords: Supervised Learning, Time based Evaluation, Performance Comparison

1. Introduction

Machine learning is a technique which enables computers to self learn patterns within the data. It is a rapidly growing domain with extensive applications in finance, computer vision, health care and multiple domains. Supervised learning is a subfield of machine learning which consists of algorithms that can learn to classify data into pre-defined data labels and perform regression analysis when trained on a labeled dataset. The performance of supervised learning algorithm is dependent on various factors like whether the dataset is balanced or not, the kind of relationship between target attribute and the dataset and the dependencies between attributes in a dataset.

In applications like computer vision and depth perception, applications need to detect the test tuple, pre-process it and then classify it in real-time. In such systems, the trade-off between the time taken and the accuracy is essential, a faster system with a marginally lower accuracy would be preferred over a slower but more accurate system. This study introduces a standardisation technique for different metrics to incorporate this time dependency. With these standardised metrics, we then outline the most suitable algorithm in a time sensitive environment.

Multiple studies have been performed to compare a subset of supervised learning algorithms for specific applications, but no attempt has been made to compare algorithms for a general case. This study bridges that gap by finding the optimal algorithm in a general case scenario. Many systems with low computing powers are sensitive to the amount of time required to train models. This study also outlines the models which are trained in lower amounts of time, yet output high performance.

The dependence of the success of an algorithm will be dependent on the kind of dataset used. In this study we will be comparing the performance of Naive Baye's, Decision Tree, Support Vector Machines, K-Nearest Neighbour, Randomised Forests, Adaptive Boosting, Gradient Boosting, Logistic Regression, Extra Trees and Linear Discriminant Analysis on Fisher Iris[1], Car Evaluation[1], Wine Origin[1], Breast Cancer[1], Cover-type[1], Abalone[1], Poker-hand[1], Adult, Human Activity Recognition[2] and Band Marketing[3] datasets and find the optimal general case algorithm based on both time and performance.

An empirical study of the performance of these algorithms has previously been performed by Caruana & Niculescu-Mizil (2006)[4]. In that study, 11 distinct datasets have been chosen and performance metrics are compared.

King, Feng & Sutherland(1995)[5] also perform a comparative analysis, but their studies cover a smaller range of algorithms than Caruna & Niculescu-Mizil. These studies provide a comprehensive analysis, but have a smaller subset of algorithms than this study. This study includes a greater number of algorithms and introduces a time standardised accuracy metric for comparison of the classifiers. The time standardisation technique is then applied to other metrics like ROC, F1 Score and Entropy.

The further sections of this study provide a short brief of the datasets that have used and the experimental setup of the methods that have been employed in this study. The optimal model based on the trade-off between time and accuracy is then discussed and its strengths and weaknesses have been assessed.

2. Datasets

A brief description of the datasets which have been used is given below. One hot encoding was used for converting all non numeric fields. Table 1 represents a summary of all the datasets used.

1) Fisher Iris: Each tuples of this dataset correspond to an Iris Flower. The flower has been categorised into 3 classes with equal prior probability for each class. Attributes which provide descriptions about the petal and sepal of the flower.
2) Abalone: A classification process is performed on the number of rings of the Abalone creature into 2 classes

from its physical attributes like height, diameter and weight.

3) Adult: Education, age, job, nationality and 14 similar attributes have been used to classify individuals earnings below or above $50,000.
4) Car Evaluation: The standard and condition of a car is evaluated in 4 distinct classes.
5) Wine Origin: Using chemical analysis of different kinds of wines, determining the origin of the wine. The wines have 3 possible origins.
6) Breast Cancer: 10 cell properties like its diameter and concavity are used to classify cells as severely or benignly affected by breast cancer.
7) Human Activity Recognition: Angular velocity and acceleration is used along with other time and frequency variables to predict the activity of the user. These measurements have been performed using smartphone sensors and a total of 561 attributes are used for classification into 6 labels.
8) Poker Hand: The suit and value of each card in a hand of poker is used to find out the poker hand. There are 10 types of hands of poker like no hand and straight and royal flush are classified.
9) Bank Marketing: This dataset includes the job, age, whether loan is taken or not and similar attributes to decide whether a given client would subscribe to a bank deposit scheme or not.
10) Cover-type: The type of forest cover on hilly surfaces and other lands are classified using land properties like soil quality, level of sunlight, proximity to roadways and similar distinct factors. The cover is classified into 7 types.

Table 1: Description of datasets used for experimentation

Dataset	No. of Records	No. of Attributes	No. of Attributes after One-hot Encoding
Adult	32561	15	108
Abalone	4177	9	10
Breast	699	11	20
Bank	41188	21	63
Car	1728	6	21
Cover Type	581012	54	54
Human Activity Recognition	10928	561	561
Iris	150	4	4
Poker	1025010	10	10
Wine	178	13	13

3. Models

The models used, their implementations and their corresponding hyper-parameters have been discussed below:

1) Naive Baye's: Gaussian implementation of Naive Baye's algorithm has been done. Each attribute is assumed to have Normal distribution. The prior class labels are determined from the data.
2) Linear Discriminant Analysis: Outputs a linear class boundary and uses singular valued decomposition as the solver. All attributes are assumed as independent.
3) Adaptive Boosting Classifier: Decision trees are used as the base classifier and 50 base estimators are chosen with a learning rate of 1.
4) Gradient Boosting: Deviance loss function is optimised with 100 boosting stages and a learning rate of 0.1 at each stage.
5) Random Forests: Results from 10 decision trees sampled for different subsets of the training data have been averaged. The splitting criterion and minimum leaf node size is same as that of decision tree used below.
6) Logistic Regression: L2 penalisation with 'liblinear' solver has been used for binary classification. The 'newton-cg' solver is used for multinomial classification. The tolerance criteria is 10^{-4} and the regularisation constant is 1.
7) K-Nearest Neighbour: 5 nearest neighbours have been chosen using Euclidean distance.
8) Decision Tree: Gini-index has been used to discern the splitting criterion and minimum leaf size of 7.
9) Extra Trees Classifier: Is an ensemble method of different highly randomised decision trees. The splitting criterion and minimum leaf node size is same as that of decision tree.
10) Support Vector Machine: Model with linear kernel has been implemented. Regularisation constant C is chosen as 1. Sigmoid, radial basis function and polynomial kernel have not been used because of their high training time and unsuitability in a real time application.

4. Evaluation and Implementation

The experimentation methodology and the pre-processing and evaluation methods are discussed in this section.

All non-numeric attributes of each dataset were first converted to numeric attributes. Then algorithms were trained and validated on 80% of the entire dataset. The rest was used for testing. The training time and testing time during predictions was recorded. The ROC, Accuracy, F1 score & Cross Entropy was then computed. These metrics were then standardised using the average training/testing time per attribute for 1 million records as shown in equation 1.

$$SM = \frac{sm}{e^t}, \qquad (1)$$

$$t = \frac{tt * 1000000}{a * r}, \qquad (2)$$

$$sm \stackrel{\text{def}}{=} performance\ metric,$$
$$tt \stackrel{\text{def}}{=} training/testing\ time,$$
$$a \stackrel{\text{def}}{=} no.of\ attributes\ in\ the\ dataset,$$
$$r \stackrel{\text{def}}{=} no.of\ train/test\ records\ in\ the\ dataset$$

These standardised scores then allow us to compare algorithm performance from across different datasets which contain different number of attributes, as the standardised metrics negates dataset specific factors.

To evaluate performance, the average performance of a classifier over different datasets is calculated. The average of a standardised metric for the different datasets was recorded. In this study, both testing and training time standardised results are discussed. To compare the performance of classifiers for each dataset, an average of all performance

metrics for individual datasets have been taken. While, taking this average the cross entropy has been transformed as shown in equation 3. This has been done by is normalizing cross entropy between 0 and 1, with and then subtracting this by 1.

$$ce = 1 - normalised(Cross\ Entropy) \quad (3)$$

5. Results & Discussions

The standardised accuracy is lower than the raw accuracy, as observed in Table 2. This difference accounts to factoring out the average testing time per million records & per attribute of the dataset. Decision Tree has the highest standardised accuracy of 0.8278 amongst all the different classifiers. It also has the highest standardised ROC and F1 score value. Although, the raw accuracy of the decision is lower than 2 other algorithms. This implies that decision trees have a low testing time as confirmed by the recorded average testing time. The KNN classifier has the lowest standardised accuracy. This can be attributed to the high testing time taken. KNN performs instance based learning, and each time retrieves the entire training dataset for each training tuple. This approach is not scalable for large datasets and hence, this method has been penalised in standardised accuracy. Similarly, adaptive boosting had a low performance score due to the high average test time. Gradient Boosting outputted a similar raw accuracy as the decision tree but took 5 times more time in doing so. Hence, there is a large gap between their standardised accuracies. Random Trees and Extra Trees Classifier had a better raw accuracy than decision trees, as they are built on top of them, but the ensemble learning methods lose out on the testing time. This gives decision tree an upper hand in real time applications as compared to the other ensemble methods. Despite a higher average test time, LDA has a relatively high standardised ROC and F1 score. It also received a higher standardised accuracy, despite a lower raw accuracy and a higher testing time as compared to Extra trees and Random Forests. This is an anomaly and implies that the average testing time could be inflated because of a relatively high testing time on one of the 11 datasets, but a lower testing time on others. Naive Baye's had the lowest accuracy amongst all classifiers and due to this, despite low testing times the classifier was unable to procure a high average standardised accuracy. SVM's had a moderate accuracy and a relatively higher testing time and hence, reported a poor standardised accuracy.

The training time standardised accuracy results illustrated in Table 3 are in stark contrast with that seen with testing time. Naive Baye's Algorithm is the most optimally algorithm with respect to training time. It has a moderate raw accuracy which is achievable by other algorithms, but it has a significantly low training time. Hence, it proves optimal while minimising training time. This algorithm is followed by KNN and then decision trees. The KNN had a significantly high testing time. The nature of the algorithm, allows it to do minimal work during training and consequently do maximum amount of work during testing. LDA proved to be an efficient algorithm when standardised with training time. It has amongst the lowest standardised accuracies. A feature of LDA is that along with low training time it reports a good raw accuracy. Similarly, decision trees a high standardised score. These are unlike KNN and Naive Baye's which report high standardised accuracy solely due to low training time. Gradient Boosting although has a high raw accuracy is very low when standardised. This is because it is the most costly algorithm to train. Similarly, adaptive boosting is costly to train and so reports a lower score. Support Vector Machine is another algorithm which is costly to train and hence, reports poor standardised scores.

Table 4 illustrates the assessment of classifiers based on their performance on individual datasets. The Iris, Wine and Abalone datasets have significantly low standardised accuracies than all other datasets. Almost all algorithms had optimal results working on the Iris Dataset, but due to high testing time, except LDA and Extra Trees classifier all other models reported weak results. Abalone and Wine dataset suffered poor results due to the ineptness of models to achieve good results during classification. Except KNN, all classifiers achieved optimal results with the human activity recognition dataset, the classifiers were both fast and accurate with this problem. Logistic regression, Naive Baye's and decision trees were the most consistent models. They did not completely falter with any dataset and produced moderate to optimal results in each case. Apart from them, almost all other models produced abysmal results in at least 1 dataset. KNN produced the worst results of the lot, and was moderately capable of classification only on the Adult and Breast dataset. This is caused by to the high testing time masking the accuracy achieved during standardisation.

6. Conclusion

Considering both training and testing time, decision tree algorithm and LDA have outperformed their counterparts. These algorithms have reported high raw accuracies and standardised ROC and F1 scores and have been accompanied with low training and testing time. Extra trees algorithm provides a high raw accuracy and is suitable where testing time needs to be minimized, but it is time consuming to train. Similarly, gradient boosting and random forests are preferred in environments where training time is unconstrained and moderate testing time along with high accuracy is desired. In training time sensitive applications, KNN and instance based learners in general provide good results. Decision trees have proven to be optimal in terms of applicability to distinct datasets. It has given acceptable results across all datasets, even where other classifiers have failed.

In both testing and training time sensitive environments, Decision tree & LDA's are the optimal general case algorithms to work with, subject to individual datasets.

Table 2: Performance metrics scaled with average test time per attributes, per million records.

Classification Algorithm	Standardised Accuracy	Raw Accuracy	Standardised AUC ROC	Standardised F1 Score	Standardised Cross Entropy	Average Test Time
Naive Baye's	0.478735	0.681504	0.477966	0.468249	0.477966	0.480331
Linear Discriminant Analysis	0.589154	0.800932	0.536378	0.525270	0.536378	2.718438
Adaptive Boosting	0.26758	0.689262	0.304318	0.322039	0.304318	8.984850
Gradient Boosting	0.435687	0.827839	0.414555	0.431439	0.414555	1.773466
Random Forest	0.501059	0.84036	0.465909	0.489958	0.465909	2.606073
Logistic Regression	0.553129	0.749713	0.475954	0.492707	0.475954	0.649609
K-Nearest Neighbours	0.093427	0.767765	0.082254	0.089356	0.082254	10.515635
Decision Tree	0.648003	0.82946	0.626496	0.642644	0.626496	0.388132
Extra Trees	0.488981	0.837897	0.487625	0.491648	0.487625	1.700676
Support Vector Machines	0.38919	0.636574	0.430234	0.422018	0.430234	1.227740

Table 3: Performance metrics scaled with average train time per attributes, per million records.

Classification Algorithm	Standardised Accuracy	Raw Accuracy	Standardised AUC ROC	Standardised F1 Score	Standardised Cross Entropy	Average Train Time
Naive Baye's	0.503420	0.681504	0.507398	0.492625	5.457912	0.384850
Linear Discriminant Analysis	0.377626	0.800932	0.343174	0.333783	2.685291	1.175850
Adaptive Boosting	0.038623	0.689262	0.034631	0.039346	0.209746	55.756082
Gradient Boosting	0.004022	0.827839	0.003430	0.004004	0.014620	151.541431
Random Forest	0.242387	0.84036	0.218595	0.239958	0.769738	18.800053
Logistic Regression	0.260197	0.749713	0.220839	0.242062	2.235554	3.617175
K-Nearest Neighbours	0.423713	0.767765	0.393230	0.411008	3.399166	1.911767
Decision Tree	0.414422	0.82946	0.395804	0.412623	1.982589	1.268641
Extra Trees	0.316087	0.837897	0.317681	0.321205	0.687705	16.842784
Support Vector Machines	0.088916	0.636574	0.092576	0.091615	0.466140	56.799344

Table 4: Dataset wise performance of individual classifiers

Dataset	NB	LDA	AB	GB	RF	LR	KNN	DT	ET	SVM
Adult	0.671282	0.764765	0.714875	0.781301	0.745108	0.681773	0.274900	0.753418	0.719324	0.534849
Abalone	0.142368	0.247404	0.010430	0.002792	0.116003	0.265123	0.043480	0.133395	0.069907	0.285510
Breast	0.524725	0.845346	0.125754	0.787313	0.610972	0.421677	0.330560	0.867816	0.607531	0.414066
Car	0.733045	0.822964	0.168010	0.390561	0.696990	0.830432	0.056304	0.934678	0.773001	0.834417
Poker	0.300360	0.346316	0.029495	0.028556	0.184470	0.343686	0.000000001	0.584608	0.108859	0.318871
Wine	0.302975	0.00000000002	0.0000001	0.059937	0.001920	0.018506	0.017782	0.094772	0.018188	0.000538
Iris	0.089567	0.803381	0.0000104	0.005204	0.00000005	0.092431	0.003745	0.284127	0.894943	0.030855
Bank	0.768985	0.126799	0.732902	0.825271	0.790821	0.833993	0.013187	0.815120	0.000085	0.840507
Human Activity Recognition	0.622328	0.962250	0.779950	0.867837	0.938507	0.963989	0.000001	0.920547	0.937075	0.963535
Cover Type	0.580001	0.653339	0.438164	0.475607	0.813215	0.671061	0.093236	0.922573	0.783470	0.394458

7. Acknowledgement

All datasets used in this experiment were sourced from UCI Machine Learning Repository. To reproduce recorded results, datasets from this repository can be used.

References

[1] Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science

[2] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra and Jorge L. Reyes-Ortiz. A Public Domain Dataset for Human Activity Recognition Using Smartphones. 21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2013. Bruges, Belgium 24-26 April 2013.

[3] Moro et al., 2014] S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems, Elsevier, 62:22-31, June 2014.

[4] Rich Caruana and Alexandru Niculescu-Mizil. 2006. "An empirical comparison of supervised learning algorithms". In Proceedings of the 23rd international conference on Machine learning (ICML '06). ACM, New York, NY, USA, pp 161-168.

[5] R & Feng, C & Sutherl, A. (2000). "STATLOG: Comparison of Classification Algorithms on Large Real-World Problems." Applied Artificial Intelligence.

[6] Ashari, Ahmad & Paryudi, Iman & Min, A. (2013). "Performance Comparison between Naïve Bayes, Decision Tree and k-Nearest Neighbor in Searching Alternative Design in an Energy Simulation Tool." International Journal of Advanced Computer Science and Applications.

[7] Nigel Williams, Sebastian Zander, and Grenville Armitage. 2006. "A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification." SIGCOMM Comput. Commun. Rev. 36, 5 (October 2006),pp 5-16.

[8] Gerard Escudero, Lluis Marquez, and German Rigau. 2000. "A comparison between supervised learning algorithms for word sense disambiguation." In Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning - Volume 7 (ConLL '00), Vol. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 31-36.
[9] Mohammed J. Zaki and Wagner Meira, "Data Mining and Analysis: Fundamental Concepts and Algorithms", Publisher: Cambridge University Press (May 12, 2014)

Author Profile

Akhil Sethia is currently pursuing his engineering in Information Technology from Dwarkadas J. Sanghvi College of Engineering, and will graduate in 2019. His research interests include Machine learning, AI and its applications to banking and finance