Post-hoc experiments for the active learning challenge

Technical note

Reference :	**FT/RD/TECH/10/09/216**
Other reference :	
Version :	1.0
Date of edition :	September, 2010

Authors :	Christophe Salperwyck, Lemaire Vincent TECH/ASAP

Summary : An analysis in a form of a benchmark was realized to examine whether the challenge's winner and the top ranked competitors obtained good performances thanks to their active strategy or thanks to the learning machine used. Therefore this benchmark studies the ability of an algorithm to learn with only few labeled examples. The objective is to rank several learning machines. An algorithm which has a good accuracy with just few examples will require buying fewer examples. Afterwards a focus on a smaller subset of algorithms to be tuned for active learning can be performed.

Keywords : Active Learning Challenge

Contents

1. Analysis: active strategy vs learning machine

An analysis in a form of a benchmark was realized to examine whether the challenge's winner and the top ranked competitors obtained good performances thanks to their active strategy or thanks to the learning machine used. Therefore this benchmark studies the ability of an algorithm to learn with only few labeled examples. The objective is to rank several learning machines. An algorithm which has a good accuracy with just few examples will require buying fewer examples. Afterwards a focus on a smaller subset of algorithms to be tuned for active learning can be performed.

To our knowledge, the state of art presents only one related work in (Forman and Cohen, 2004). Forman and Cohen studied the performance on small and not balanced text datasets. The learners used were: SVM, naive Bayes and logistic regression. In our case, datasets are not text based but datasets from this active learning challenge. The benchmark was also performed on a larger set of learners: decision tress, larger variety of naive Bayes, boosting...

1.1 Experimental conditions

This section presents the conditions in which the experiments were run. First the learning machines will be detailed, then the datasets used to perform experiments and finally the ALC criterion used to evaluate the performances.

1.1.1 Learning Machines

Two 'public'[1] software were used: WEKA (Witten and Frank, 2005) (version 3.7.1) and Khiops (Boullé, 2004) (version 7.0).

- Weka:
  - Bayes
    - Non supervised (John and Langley, 1995): standard naive Bayes. Numeric estimator precision values are chosen based on analysis of the training data.
    - Supervised: same as before but a supervised discretization (MDL) is used to convert numeric attributes to nominal ones.
    - BayesNet (Bouckaert, 2004): Bayes network learning using various search algorithms and quality measures. Used with default parameters: SimpleEstimator and K2 search algorithm.
  - Regression:
    - Logistic regression: multinomial logistic regression model with a ridge estimator which is based on (Cessie and Houwelingen, 1992). Default parameters except from the maximal number of iterations set to 100.
  - Trees:

1. All experiments in this benchmark are reproducible.

    * ADTree (Freund and Mason, 1999): decision tree with many subtrees and combined with boosting.
    * J48 (Quinlan, 1993): C4.5 decision tree proposed by Quinlan.
    * SimpleCart (Breiman et al., 1984): binary tree based on the Gini coefficient.
    * RandomForest (Breiman, 2001): forest of random trees. Default number of trees in the forest is 10. 40 was also tried.
  - Vote
    * VFI (Demiröz and Güvenir, 1997): classification by voting feature intervals. Intervals are constructed around each class for each attribute (basically discretization). Class counts are recorded for each interval on each attribute and the classification is by voting. The default parameters are using the "weight feature intervals by confidence". It was also tried without it.

- Khiops[2]: a tool developed by Orange Labs. It implements, for supervised classification, a naive Bayes and a selected naive Bayes.

  Naive Bayes or Selective Naive Bayes (Boullé, 2006b) were tested with different pretreatments on numerical and nominal attributes. On nominal attributes two pretreatments were tested (i) Basic Grouping: a group per observed value or (ii) MODL discretization (Boullé, 2006a). On numerical attributes three different pretreatments were tested (i) Equal Frequency (ii) Equal Width or (iii) MODL grouping method (Boullé, 2005). The two unsupervised Equal Frequency and Equal Width methods are used with a fixed number of bins set to 10. If the number of observed values is below 10 the number of bins is reduced to the number of observed data. In the table 1 a method named 'K-NB_A_B' (respectively 'K-SNB_A_B') is a Naive Bayes (respectively Selective Naive Bayes) using the pretreatment 'A' for numerical attributes and the method 'B' for categorical attributes.

1.1.2 Number of labeled examples

In the test phase of the challenge a seed was given for each dataset. This seed was used in this benchmark as the first labeled example. Then, to have higher numbers of labeled examples, examples were randomly drawn in the remaining train examples to reach the desired number of labeled examples. The train set sizes are: $S = \{S_1, S_2, ..., S_{max}\} = \left\{2, 2^2, 2^3, 2^4, \ldots, 2^{(\lfloor log_2(n-1) \rfloor)}, n\right\}$,where n is the maximum budget for each dataset of the challenge (A, B, ..., F). This procedure was realized 10 times for each dataset to have a mean result and a variance on the results. The performances on the test sets are given in table 1.

Those experiments give for each dataset a AUC curve on the test dataset versus the number of train examples used to train the learning machine. Each curve is constituted of $|S|$ points whatever the learning machine is[3].

1.1.3 ALC criterion

To compare the results obtained by the different learning machines, the ALC criterion (see Section 2) proposed in this challenge is used. This criterion has good properties considering the sequence of labeled examples used in this benchmark. The x-axis is logarithmic which means that the AUC for 2 examples will contribute to the ALC as much as the AUC for 4096 examples. Indeed this criterion emphasises the AUC on smaller subsets, what is especially the focus of this study. The ALC criterion has some drawbacks when comparing learning machine results which does not used

2. `www.khiops.com`
3. This benchmark is only 'supervised learning' oriented therefore the unlabeled examples of the train dataset $(n-S_i)$ are not given to the learning machine.

the same sequence of labeled examples. But this is not the case in this benchmark as all learners are trained using the same training datasets.

1.2 Results

The results obtained by the different learning machines are presented in Table 1. The results are also compared to the challenge's competitors ones. The number of points in the AUC curve is a key element (see Sections 2 and 1.1.3) that greatly impact the ALC value. If the results obtained by the 'Passive Learning Strategy' (all labels queried at once, (Cawley, 2010)) are not considered, the results presented in Table 1 are similar to those obtained by the competitors. For example 'W-RandomForest40' could be ranked 3, 9, 5, 13, 5, 2 and 'W-VFI_NoWBC' could be ranked 6, 8, 7, 8, 14, 5 for datasets A, B, C, D, E, F. The AUC curves for 'W-RandomForest40' are presented Figure 1(b).

Study results

This study was done without a fine tuning of the algorithms but mainly using their default parameters. C4.5 (J48) and CART (SimpleCart) are references in data mining but this study shows that they are not made to perform well on small datasets. On the contrary, naive Bayes algorithms are known to perform well on small datasets (Domingos and Pazzani, 1997) and Figure 1(a) confirm this result when they are compared to C4.5. They are starting to perform well but only after seeing around $2^7(128)$ to $2^{10}(1024)$ examples and after that they rapidly reach their maximum accuracy. Even if naive Bayes are performing better than C4.5, they are not the best for the smaller datasets.

Tree classifiers are not performing well out of the box, but surprisingly in combination with bagging/boosting techniques they are on the top of this study: RandomForest, ADTree. Figure 1(b) shows the Random Forest (with a forest size of 40) results on all datasets, which is the overall best classifier in this study.

A vote by majority on a discretization technique (VFI_NoWBC) is the second best performer. A comparison of its results on the dataset D is presented on Figure 1(c). Its performance on dataset D (lots of attributes and sparsity) is much better than RandomForest40 and naive Bayes but it is not confirmed on datasets with less sparsity (dataset E).

Regularized methods are known to have low bias-variance (Cucker and Smale, 2008) and therefore, in order to maintain this low bias-variance, their variance but also AUC will be lower on the smaller datasets. On Figure 1(d), this statement is confirmed as the AUC increases later: the classifier is more conservative and prefers not to make highly variable predictions.

Comparisons

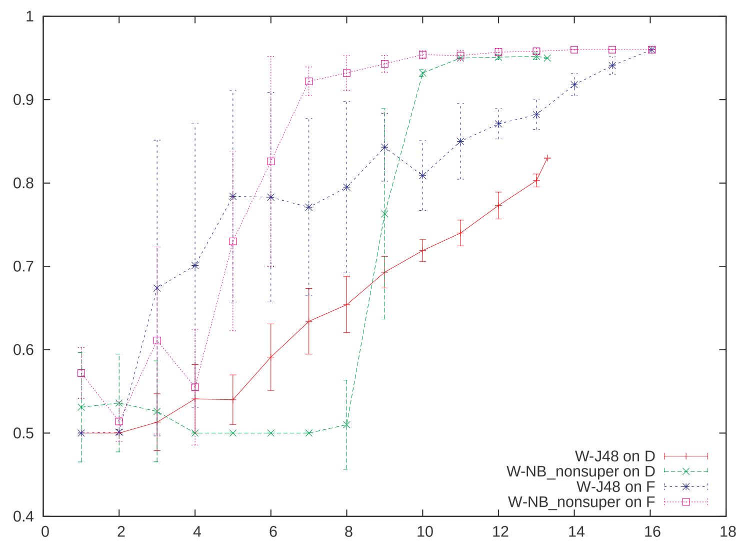
The results in the literature[4] are confirmed in this study: voting and ensemble of classifiers (even when using discriminative classifiers) performs very well (Bauer and Kohavi, 1999), generative classifiers are better than discriminative classifiers when the number of examples is low (Bouchard and Triggs, 2004), regularized methods are robust (Cucker and Smale, 2008).

The results obtained in this benchmark can be compared with some of the top 5 challenge's competitors as similar methods were used: INTEL used random forest and IDE used a logistic regression on the smaller datasets and boosted decision trees (which can probably be compared to the W-ADTree) on bigger ones. Competitors' results and ours are very similar and when comparing their AUC curves to ours the same behaviors are observable. On the smaller datasets, in which the use of unlabeled data was very beneficial, competitors performed slightly better. Figure 2 gives a view of the distribution of the competitors' results and the results obtained in this benchmark: on the left hand side (Figure 2(a)) all the competitors' results and on the right hand side (Figure 2(b)) competitors' results without those obtained with a strategy very close to a 'Passive Learning Strategy' (Cawley, 2010). The comparison between Figure 2(a) and Figure 2(b) is instructive.

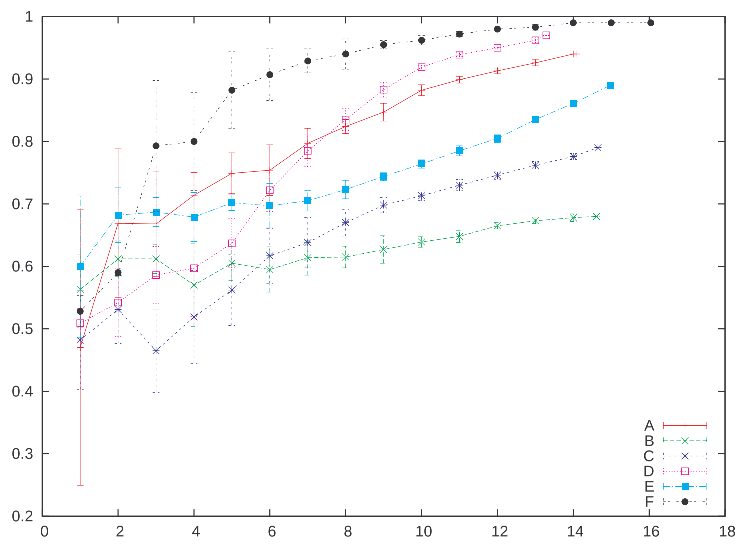
4. Most of the literature's studies are not made on such small datasets, in consequence for the sake of the comparison, results are mostly similar only after seeing $2^7(128)$ to $2^{10}(1024)$ examples.

Table 1: ALC for all datasets and methods tested. The value in () presents the AUC obtained by the method when using all the budget (final AUC). The first line of the Table gives the ALC of the winners and their final AUC.

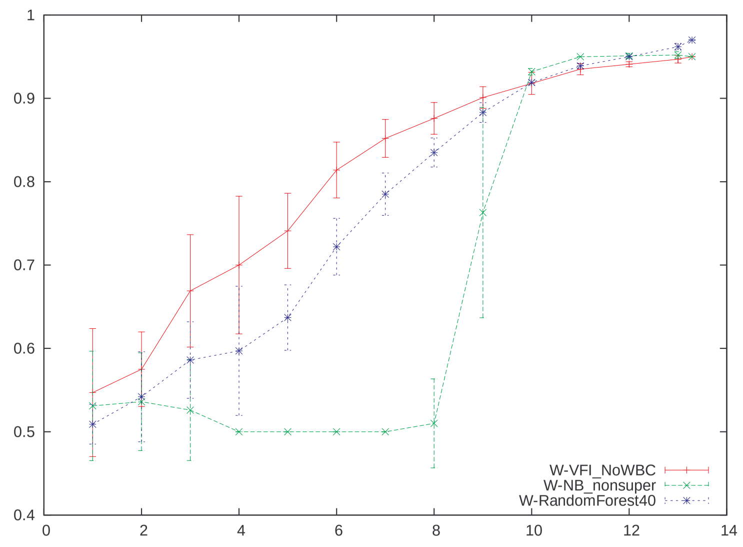
Method (W-xxx : Weka, K-xxx : Khiops)	A 0.6289 (0.8622)	B 0.3757 (0.7327)	C 0.4273 (0.7994)	D 0.8610 (0.9641)	E 0.6266 (0.8939)	F 0.8018 (0.9990)
W-ADTree	0.5052 (0.8620)	0.2603 (0.7420)	0.2667 (0.7600)	0.3550 (0.8110)	0.3889 (0.8060)	0.7419 (0.9700)
W-BayesNet	0.3906 (0.8360)	0.2196 (0.6810)	0.2479 (0.7200)	0.3242 (0.9500)	0.3504 (0.7500)	0.7716 (0.9700)
W-J48	0.2272 (0.8000)	0.0168 (0.5340)	0.0527 (0.5610)	0.2670 (0.8300)	0.1677 (0.7500)	0.5814 (0.9600)
W-Logistic100	0.3771 (0.8800)	0.2198 (0.7100)	0.2213 (**0.8100**)	0.4747 (0.8570)	0.3527 (0.7900)	0.6880 (0.9800)
K-NB_EqualFreq_BasicGroup	0.5166 (0.8350)	0.2582 (0.6686)	0.1697 (0.6940)	0.4910 (0.8477)	0.2830 (0.7192)	0.7047 (0.9707)
K-NB_EqualFreq_MODL	0.5188 (0.8350)	0.2579 (0.6689)	0.1697 (0.6940)	0.4910 (0.8477)	0.2830 (0.7192)	0.6672 (0.9706)
K-NB_EqualWidth_BasicGroup	0.5072 (0.8340)	0.2572 (0.6671)	0.1763 (0.7014)	0.5476 (0.9357)	0.2987 (0.7248)	0.7038 (0.9666)
K-NB_EqualWidth_MODL	0.5102 (0.8340)	0.2573 (0.6670)	0.1763 (0.7014)	0.5476 (0.9357)	0.2987 (0.7248)	0.6644 (0.9666)
K-NB_MODL_BasicGroup	0.3684 (0.8383)	0.1705 (0.6850)	0.2225 (0.6948)	0.3803 (0.9563)	0.2907 (0.7282)	0.7463 (0.9728)
K-NB_MODL_MODL	0.3733 (0.8383)	0.1765 (0.6814)	0.2225 (0.6948)	0.3803 (0.9563)	0.2907 (0.7282)	0.6908 (0.9728)
W-NB_nonsuper	0.4255 (0.8240)	0.2089 (0.6600)	0.2369 (0.7200)	0.3319 (0.9520)	0.2983 (0.7400)	0.6729 (0.9600)
W-NB_super	0.3913 (0.8360)	0.2420 (0.6800)	0.2485 (0.7200)	0.3319 (0.9520)	0.3516 (0.7500)	0.7698 (0.9700)
W-RandomForest	0.5064 (0.9100)	0.2086 (0.6700)	0.2350 (0.7300)	0.4142 (0.9300)	0.4113 (0.8300)	0.7645 (**0.9900**)
W-RandomForest40	**0.5940** (**0.9400**)	0.2505 (0.6800)	0.2878 (0.7900)	0.5314 (**0.9700**)	**0.4873** (**0.8900**)	**0.7915** (**0.9900**)
W-SimpleCart	0.1453 (0.8400)	0.0735 (0.6900)	0.0247 (0.5800)	0.1190 (0.7050)	0.2129 (0.7700)	0.6033 (0.9700)
K-SNB_EqualFreq_BasicGroup	0.4204 (0.8546)	0.2108 (0.7158)	0.2804 (0.7875)	0.3055 (0.8513)	0.3077 (0.7656)	0.6998 (0.9725)
K-SNB_EqualFreq_MODL	0.4212 (0.8548)	0.2082 (0.7128)	0.2804 (0.7875)	0.3055 (0.8513)	0.3077 (0.7656)	0.6673 (0.9724)
K-SNB_EqualWidth_BasicGroup	0.4405 (0.8571)	0.2215 (0.7088)	0.2607 (0.7798)	0.3672 (0.9679)	0.3037 (0.7729)	0.6931 (0.9678)
K-SNB_EqualWidth_MODL	0.4429 (0.8571)	0.2203 (0.7082)	0.2607 (0.7798)	0.3672 (0.9679)	0.3037 (0.7729)	0.6653 (0.9677)
K-SNB_MODL_BasicGroup	0.3812 (0.8612)	0.2207 (**0.7353**)	**0.2884** (0.7858)	0.3537 (0.9633)	0.3470 (0.8159)	0.7233 (0.9739)
K-SNB_MODL_MODL	0.3811 (0.8612)	0.2162 (0.7349)	**0.2884** (0.7858)	0.3537 (0.9633)	0.3470 (0.8159)	0.6903 (0.9739)
W-VFI	0.4249 (0.8200)	0.2036 (0.6400)	0.1840 (0.6700)	0.4757 (0.9000)	0.2173 (0.6850)	0.7302 (0.9420)
W-VFI_NoWBC	0.5198 (0.8210)	**0.2706** (0.6510)	0.2523 (0.6810)	**0.6182** (0.9500)	0.2588 (0.6700)	0.7545 (0.9560)



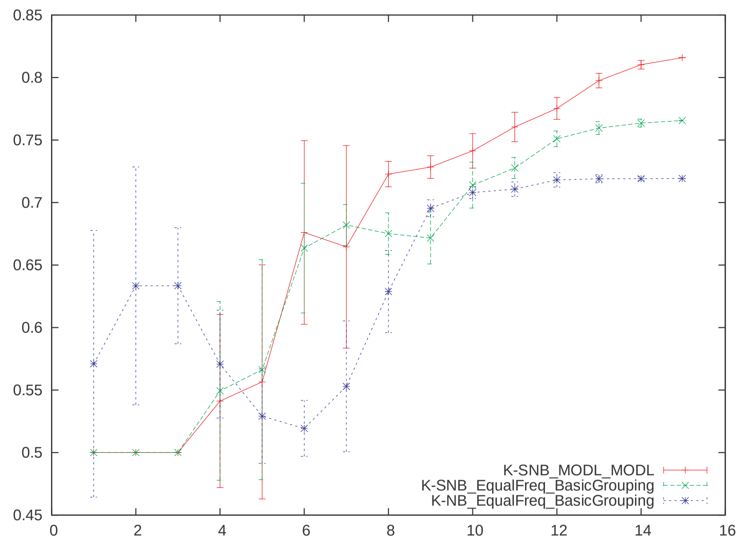
(a) J48 vs Naive Bayes on datasets D and F



(b) RandomForest40 on all datasets

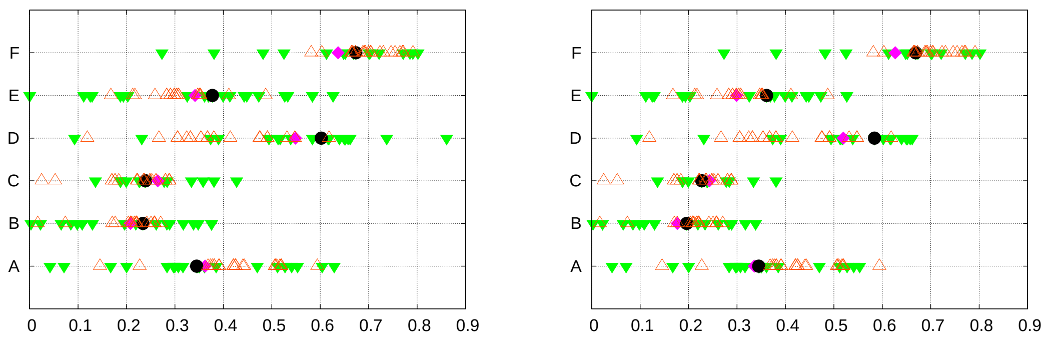


(c) VFI_NoWBC vs Naive Bayes vs RandomForest40 on dataset D



(d) NB vs SNB and EqualFreq/BasicGrouping vs MODL on dataset E

Figure 1: AUC as a function of the $\log_2$ number of examples



(a) All results (b) Unpassive results

Figure 2: Distribution of the results. The horizontal axis represents the ALC value and the vertical axis the different datasets. Legend: ▼ Competitor's results; ♦ and ● respectively mean and median results of the competitors; △ this benchmark's results.

The results presented in this section indicate that on this challenge using those datasets and this protocol that active strategies had a small impact on the final ALC value.

References

E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine learning*, 36(1):105–139, 1999.

G. Bouchard and B. Triggs. The tradeoff between generative and discriminative classifiers. In *IASC International Symposium on Computational Statistics (COMPSTAT)*, pages 721–728, 2004.

R. Bouckaert. Bayesian Network Classifiers in Weka, 2004. URL `http://weka.sourceforge.net/manuals/weka.bn.pdf`.

M. Boullé. Khiops: A Statistical Discretization Method of Continuous Attributes. *Machine Learning*, 55(1):53–69, April 2004.

M. Boullé. A grouping method for categorical attributes having very large number of values. *Machine Learning and Data Mining in Pattern Recognition*, pages 228–242, 2005.

M. Boullé. MODL: A Bayes optimal discretization method for continuous attributes. *Machine Learning*, 65(1):131–165, May 2006a.

M. Boullé. Regularization and Averaging of the Selective Naive Bayes classifier. *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pages 1680–1688, 2006b.

L. Breiman. Random forests. *Machine learning*, 25(2):5–32, 2001.

L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and regression trees*. Chapman and Hall/CRC, 1984.

G. Cawley. Some Baseline Methods for the Active Learning Challenge. *JMLR: Workshop and Conference Proceedings - Workshop on Active Learning and Experimental Design*, 10, 2010.

S. L. Cessie and JC V. Houwelingen. Ridge estimators in logistic regression. *Applied Statistics*, 1992.

F. Cucker and S. Smale. Best Choices for Regularization Parameters in Learning Theory: On the Bias-Variance Problem. *Foundations of Computational Mathematics*, 2(4):413–428, 2008.

G. Demiröz and H. Güvenir. Classification by voting feature intervals. *Machine Learning: ECML-97*, pages 85–92, 1997.

P. Domingos and M. Pazzani. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine learning*, 130:103–130, 1997.

G. Forman and I. Cohen. Learning from little: Comparison of classifiers given little training. *Knowledge Discovery in Databases: PKDD 2004*, pages 161–172, 2004.

Y. Freund and L. Mason. The alternating decision tree learning algorithm. In *Machine learning*, pages 124–133, 1999.

G.H. John and P. Langley. Estimating continuous distributions in Bayesian classifiers. *Proceedings of the eleventh conference on*, 1995.

J. R. Quinlan. *C4.5: programs for machine learning.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

I. H. Witten and E. Frank. *Data mining: practical machine learning tools and techniques.* Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann, second edition, 2005.