Assessing the differences in accuracy between GFSs with bootstrap tests

Luciano Sánchez
Departamento de Informática
Universidad de Oviedo
luciano@uniovi.es

José Otero
Departamento de Informática
Universidad de Oviedo
jotero@lsi.uniovi.es

Jesús Alcalá
Departamento de C.C.I.A.
Universidad de Granada
jalcala@decsai.ugr.es

Abstract

The study of the balance between linguistic interpretability and numerical accuracy in genetic fuzzy systems is an active area of research, and a rich set of procedures for comparing the understandability of fuzzy rule bases is available. Nevertheless, comparing the numerical accuracy of two GFS is relegated to a second plane, or assumed solved, as most of the researchers in this area use classical, parametric hypotheses based, statistical tests. With this paper, we intend to show that the straight use of classical tests to compare the accuracy of different machine learning algorithms may produce misleading results, and propose to substitute them by bootstrap tests in the experimental designs of the cross-validation kind.

Keywords: Genetic Fuzzy Systems, Interpretability vs. Accuracy Tradeoff, Experimental Design, Bootstrap Tests.

1 Introduction

Every proposal of a new Genetic Fuzzy System (GFS) is numerically validated by means of a set of experiments. Typically, the objective of the experimental design is to judge whether there exist a significant difference between the algorithm being evaluated, and a meaningful selection of the state of the art. The opposite can also happen, and sometimes we want to show that these differences are *not* significant. For instance, we might be interested in showing that a new, highly interpretable rule base, is not significantly less accurate than a previous, more complex one.

This concept of experimental design, based upon the numerical evaluation of an algorithm over a set data, is still a controversial point [5][9][7][12]. Besides, under general conditions, we can assume that comparing the accuracy of two algorithms consists in deciding whether their respective expected fitness (over the whole population) are equal [2]. The most frequent experimental design used to perform this comparison is the *cross validation* [10]. Under this framework, every algorithm being compared, when evaluated over a set of test partitions, eventually will produce a set of fitness values, that can be regarded as a sample of a random variable. Should we want to contrast that two algorithms are different, we use a statistical test whose null hypothesis is "the expectation of either random variable is the same", with general alternative hypothesis. Test statistics are used to measure the discrepancy between the data and the null hypothesis. In the parametric setting, we have an explicit form from the sampling distribution of the data, with some unknown parameters. Often, normality is assumed, and t-tests [4] are of widespread use.

According to our own experience, the predominance of t-tests in machine learning related experimental designs is not completely justified. t-tests are the most powerful option when the sampling distribution is gaussian. But, in many machine learning problems this assumption is not true. In case a t-test is applied to compare two non-gaussian distributions, the probability of deciding that two equivalent algorithms are different (the so called "type-I error") quickly increases. This means that, for instance, a new algorithm that only obtains a marginal result might

be incorrectly thought of as an improvement and, conversely, in the context of the accuracy-interpretability tradeoff, it could also happen that certain intelligent algorithms are incorrectly regarded as less precise than their statistical counterparts.

The customary procedure consists in choosing between a nonparametric test and the t-test, on the basis of a goodness of fit test (see Fig. 1 for an example of such a construction.) But this last construction is not optimal: on the one hand, the goodness of fit test discards a high percentage, but not all, of the non-gaussian distributions, which pass on to the t-test and influence the type-I error. On the other hand, for the most frequent nonparametric tests used in machine learning literature (Wilcoxon [13], Mann-Whitney [8]) the probability of deciding that two different algorithms are the same is higher than that of the t-test. There also exist different, less extended uses of type-t tests (see, for example, the 5x2cv [5] and 5x2cv-f [1] methods) that do not rely on a goodness of fit test, but they are not free from these two effects, as we will check later in the empirical analysis.

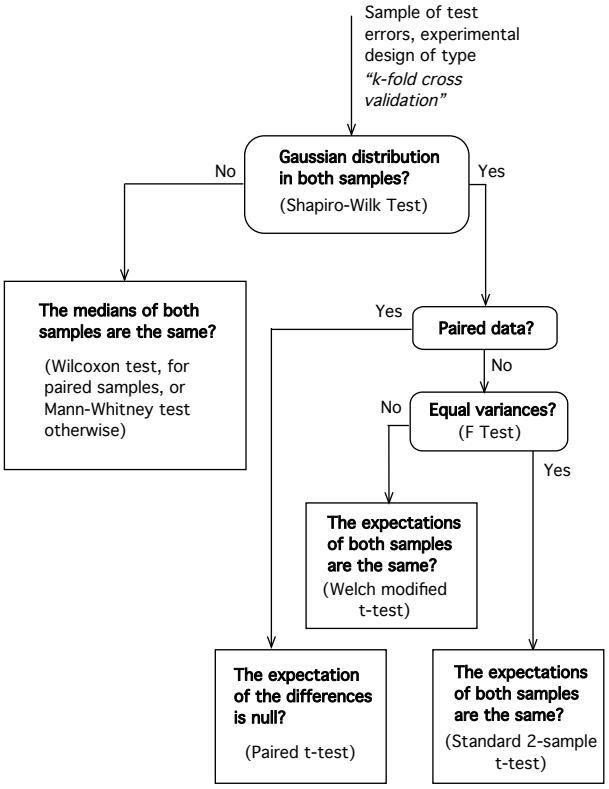

Figure 1: Combination of tests in the classical framework. A goodness of fit test is used to decide whether the t-test can be applied, and a nonparametric test is applied if not. This setup will be used in the experimental part of this article.

2 Bootstrap Tests

Recent advances in so-called *computer-intensive* statistics make use of extensive repeated calculations to explore the sampling distribution of a parameter estimator. In particular, the *bootstrap* procedure [14][15], construct estimates based on the replacement of the unknown distribution F of the data by its Empirical Distribution Function (EDF) $\widehat{F}$.

Let us suppose, for example, that we want to obtain the density function of an estimator of a parameter θ of certain distribution F; its sample median, say. We are given a random sample $(y_1, \ldots, y_n)$, $y_i \mapsto F$. First, we draw a large number R of *bootstrap samples* y^*_r, with $y^*_{ri} \mapsto \widehat{F}$ (these are resamples of y, taken *with replacement*.) The density function of the sample median is then approximated by the histogram of the bootstrap sample medians θ^*_r, or by an estimation taken from them, for instance a kernel smoothing.

Bootstrap techniques have been applied to estimate the error rate of an algorithm and confidence intervals for it, [16][17], and used to derive relations between bias and variance of classifiers [18]. The use of some early semiparametric bootstrap based tests is also discussed in artificial intelligent related works [3] but, up to our knowledge, recent bootstrap techniques like the "exponential tilts" based test, that will be discussed in section 2.1.2, have not been yet applied to judge the relevance of the differences between algorithms in the machine learning literature.

A statistical test is based on a *test statistic* T, which measures the discrepancy between the data and the null hypothesis H_0. If the observed value of the test statistic is t, then the level of evidence against H_0 is measured by the p-value

$$p = \Pr(T \geq t \mid H_0). \tag{1}$$

Bootstrap tests numerically approximate Eq. (1) from a sample y. The technique is similar to the method outlined before: we first build R boot-

strap resamples of the data *that fullfill* H_0. The test statistic t_r^* is evaluated in all of them, and the fraction of these samples for which this value is greater or equal than T approximates the desired p-value:

$$p_{\text{boost}} = \Pr^*(T^* \geq t \mid \widehat{F}_0). \tag{2}$$

The key difference with the basic bootstrap is in the selection of the resampling distribution. Since all bootstrap samples must fulfill H_0, the EDF is not valid, and we must use a suitable *null resampling distribution* $\widehat{F}_0$. The selection of $\widehat{F}_0$ is the main difficulty of the design of a bootstrap test.

2.1 Comparison of two means

We have mentioned that two algorithms are regarded as equally precise if their mean test errors are not significantly different. In the following, let us suppose we are given two samples containing the test errors of the two algorithms being compared, $(y_{11}, \ldots, y_{1n})$ and $(y_{21}, \ldots, y_{2n})$, and let H_0 be "$\mu_1 = \mu_2$", where μ_1 and μ_2 are the means of the respective populations. We want to test H_0 using the test statistic $t = \overline{y}_2 - \overline{y}_1$, and admit that the one-sided alternative $H_A = \overline{y}_2 > \overline{y}_1$ is appropriate.

There exist a wide catalog of semiparametric and fully non parametric bootstrap tests that can be applied to this problem. In the next two subsections, a basic, easy to program test is proposed, along with a state of the art one.

2.1.1 Nonparametric, Permutation-based

If H_0 is true, then the expectation of the differences is null $E(y_1 - y_2) = 0$. Therefore, the empirical distribution function is not a valid null resampling model, since $\overline{y}_1 - \overline{y}_2 \neq 0$. Following the idea under some permutation tests [3], we can design an augmented sample containing the differences $y_1 - y_2$ and its negated values $y_2 - y_1$. It is clear now that the resamples of the augmented sample fulfill the null hypothesis. The corresponding test consists in applying Eq. (2) to the sample obtained by the following algorithm:

For r in $1 \ldots R$ do

1. Build $(y_1^*, \ldots, y_{2\cdot n}^*)$ by resampling with replacement the vector $(y_{11} - y_{21}, \ldots, y_{1n} - y_{2n}, \ldots, -y_{11} + y_{21}, \ldots, -y_{1n} + y_{2n})$
2. Compute $t_r^* = \overline{y}^*$

2.1.2 Tilted EDFs based

A better approximation to the problem consists in defining the null sampling distribution $\widehat{F}_0$ by means of two sets of probabilities $(p_{11}, \ldots, p_{1n})$ and $(p_{21}, \ldots, p_{2n})$ such that

$$\sum_{i=1}^{n} p_{1i} y_{1i} = \sum_{i=1}^{n} p_{2i} y_{2i}. \tag{3}$$

It is clear than samples of data, drawn with these weights, fulfill H_0. Now we select the set of p_{ki} that forms the valid probability distribution nearest to the EDF, making the Kullback-Leibler divergence between $\widehat{F}_0$ and $(1/n, \ldots, 1/n)$ to be minimum (the selection of the KL divergence leads to the maximum likelihood estimation of the unknown parameters.) We are posed a constrained optimization problem, that can be solved with the help of three Lagrange multipliers

$$\begin{array}{rcl} E & = & \sum p_{1i} \log(n \cdot p_{1i}) \\ & + & \lambda(\sum_{i=1}^{n} p_{1i} y_{1i} - \sum_{i=1}^{n} p_{2i} y_{2i}) \\ & + & \sum_{k=1}^{2} \alpha_k (\sum_{i=1}^{n} p_{ki} - 1) \end{array} \tag{4}$$

Dropping constant terms and differentiating E w.r.t. p_{ki} we obtain that

$$\log p_{ki} + 1 + \lambda p_{ki} y_{ki} + \alpha_1 p_{ki} = 0 \tag{5}$$

and from the constrains $\sum_i p_{ki} = 0$ we obtain that

$$p_{ki} = \frac{\sum y_{ki} \exp(-\lambda y_{ki})}{\sum \exp(-\lambda y_{ki})} \tag{6}$$

where λ must be determined numerically, after combining (3) and (6). These expressions of the resampling weights are known as *exponential tilts* of the empirical distribution function.

The test consists in applying Eq. (2) to the sample obtained by the algorithm that follows:

For r in $1 \ldots R$ do

1. Build $(y_{11}^*, \ldots, y_{1n}^*)$ by resampling with probabilities $(p_{11}, \ldots, p_{1n})$ the vector $(y_{11}, \ldots, y_{1n})$

2. Build $(y_{21}^*, \ldots, y_{2n}^*)$ by resampling with probabilities $(p_{11}, \ldots, p_{1n})$ the vector $(y_{21}, \ldots, y_{2n})$
3. Compute $t_r^* = \overline{y}_1^* - \overline{y}_2^*$

3 Empirical Study

3.1 Synthetical data

To compare the tests been discussed, we need to use problems and learning methods with known statistical properties. We have chosen Haykin's two gaussians problem [6]. This problem comprises two samples of the same size, of two bivariate gaussian distributions with means $(0, 0)$ and $(2, 0)$ and variances 2I and 4I. The samples are crafted to have a linear suboptimal solution very near to the optimal one, which is quadratic.

We will use our tests to assess the difference between Linear Discriminant Analysis (LDA) and Quadratic D. A. (QDA) methods, and estimate the power of a test by counting how many times it fails to distinguish between LDA and QDA. Moreover, since we know the true distribution that originated the samples, we can analytically derive the optimal Bayesian classifier, which is not different in expected error from QDA (it has lower variance, thus it a better algorithm, but current experimental designs only look at the mean.) The number of times a test judges that QDA is different than the true optimal classifier, is an estimator of the type I error of the test.

We have studied the influence of the number of folds in cross validation (10, 50 and 100), the number of examples in the training set (250 and 500), the confidence level (0.975, 0.95, 0.90) and the distance between classes. Two additional problems were generated by displacing all points in the second class 0.25 units to the left (problem "B") or 0.1 units to the right (problem "C"). The results are summarized in two Tables: (1) –datasets of size 250– and (2) –size 500.– The datasets of size 250 are too small, therefore the tests do not have power values near to 1. Observe the high sensibility of all tests to small displacements in the data (columns A, B and C.) In this case, bootstrap tests were better in type I error, and also the best compromise between power (absence of false equals between LDA and QDA) and type I error (presence of false differences between QDA and the optimum). In Table (2) the effect of the number of folds is shown. Now, the learning algorithms are given information enough, thus the estimations of power and error at 95% are near 1 and 0, and improve as cross validation tends to be a leave-one-out. Again, the presence of not gaussian distributions confuses parametric tests (in particular, 5x2cvf, which relies on a F test.) Bootstrap tests were again the best compromise, followed by classical tests.

3.2 Benchmark data

In table 3, three fuzzy rule learning algorithms (Fuzzy Genetic Programming, Fuzzy Adaboost and Fuzzy Logitboost) and LINear classifiers were compared to conjugate-gradient trained multilayer perceptrons. The letter 'Y' means that the algorithm is less accurate than the net, and 'N' means that there are not significant differences ($\alpha = 0.05$.) Observe that, if classical tests are used, the linear classifier is regarded as equal to the neural net in 7 of 10 datasets, but bootstrap tests show that the net was more precise in 8 of 10 sets. The same happens to Fuzzy Adaboost, which classical tests regarded as similar to the net 4 of 10 times, but only 2 times when bootstrap tests are used.

4 Concluding Remarks

Previous works combining bootstrap and machine learning suggested alternatives to cross validation when estimating the error of an algorithm. But many benchmark data is already organized into train and test partitions, thus these alternatives have not been very used in practice. With this work, a different approach is proposed: we recommend the use of certain bootstrap tests as a direct "plug-in replacement" for t-tests, used in combination with cross validation. According to our results, bootstrap tests collect more information from the results of an experimentation than classical tests, thus being less prone to conclude that two significantly different algorithms are the same, or, by the contrary, that two similar algorithms are different.

	Problem A			Problem B			Problem C		
	0.025	0.05	0.10	0.025	0.05	0.10	0.025	0.05	0.10
Combined classical tests, k-fold cv									
10 fold, power	0.16	0.34	0.75	0.71	0.90	1	0.08	0.12	0.28
10 fold, type-I err	0.35	0.38	0.75	0.02	0.05	0.13	0.02	0.03	0.15
5x2cv tests									
5x2cv power	0.09	0.21	0.38	0.25	0.44	0.64	0.05	0.08	0.19
5x2cv, type-I err	0.08	0.15	0.26	0.01	0.02	0.03	0.05	0.08	0.14
5x2cvf power	0.37	0.57	0.84	0.76	0.87	1	0.19	0.28	0.61
5x2cvf, type-I err	0.27	0.48	0.72	0.21	0.40	0.65	0.25	0.43	0.70
Bootstrap tests									
permutation, 10 fold, power	0.05	0.19	0.50	0.51	0.82	1	0.00	0.01	0.11
permutation, 10 fold, type-I err	0	0	0	0	0	0	0	0	0
tilt, 10 fold, power	0.12	0.34	0.68	0.72	0.90	1	0.01	0.05	0.26
tilt, 10 fold, type-I err	0	0	0	0	0	0	0	0	0

Table 1: Analysis of sensibility of the tests w.r.t the selection of the problem. Problem A is the original one, problems B and C have less/more overlapped classes, respectively. The power of the tests should be higher in B, and lower in C, than they are in the original problem "A". The number of samples is 250, which is small, therefore the mean error has a high variance and the overall error is high. Conservative tests are best for this framework. According to our oppinion, the best tests for these problems were the two bootstraps, followed by 5x2cv, by the classical combination t/Wilcoxon and lastly by 5x2cvf, which was too dependant on the normality of the samples.

	10 fold			50 fold			100 fold		
	0.025	0.05	0.10	0.025	0.05	0.10	0.025	0.05	0.10
Combined classical tests, k-fold cv									
power	0.82	0.94	0.99	1	1	1	1	1	1
type-I err	0.04	0.06	0.25	0.00	0.03	0.19	0.03	0.05	0.18
5x2cv tests									
power	0.30	0.43	0.62	-	-	-	-	-	-
type-I err	0.05	0.07	0.20	-	-	-	-	-	-
power	0.74	0.92	0.99	-	-	-	-	-	-
type-I err	0.18	0.37	0.60	-	-	-	-	-	-
Bootstrap tests									
permutation, power	0.64	0.91	1	0.85	1	1	0.92	1	1
permutation, type-I err	0	0	0	0	0	0	0	0	0
tilted, power	0.78	0.98	1	0.96	1	1	0.97	1	1
tilted, type-I err	0	0	0	0	0	0	0	0	0

Table 2: Numerical estimations of power and type-I error for all the considered experimental designs. "Power" rows show the percentage of times the test detected the linear classifier was different from the quadratic one. "Type-I Error" show the percentage of times the test was wrong and signaled that the optimal classifier had a different mean than the quadratic one. The number of different partitions evaluated for each experimental setup is 100. Tilted boostrap was uniformly better at 90% and 95% levels. Comparing the first column of this table with the preceding one, the effect of an adequate sample size when drawing conclusions from error data is clear.

	Tilted Bootstrap				t + Wilcoxon			
Dataset	LIN	Fuzzy GP	Fuzzy ADA	Fuzzy LOG	LIN	Fuzzy GP	Fuzzy ADA	Fuzzy LOG
aut	Y	Y	Y	Y	N	Y	Y	Y
bre	N	Y	Y	N	N	Y	N	N
gls	N	N	Y	Y	N	N	N	Y
h-h	Y	Y	Y	Y	N	Y	Y	Y
ion	Y	Y	Y	Y	N	Y	Y	Y
lym	Y	N	Y	Y	Y	N	Y	Y
pim	Y	N	N	N	N	N	N	N
prt	Y	N	N	Y	Y	N	N	Y
wbcd	Y	N	Y	Y	N	N	Y	Y
zoo	Y	Y	Y	Y	Y	Y	Y	Y

Table 3: Assessment of differences in accuracy between a neural network and a selection of algorithms with a different degree of linguistic interpretability, under bootstrap tests, and under the classical setup. 'Y' means that the algorithm is less accurate than the net, and 'N' means that there are not significant differences ($\alpha = 0.05$.)

Acknowledgments

This work was funded by Spanish M. of Science and Technology and by FEDER funds, under the grant TIC-04036-C05-05.

References

[1] Alpaydin E.: Combined 5x2cv-F test for Comparing Supervised Classification Learning Algorithms. Neural Computation 11 (1999) 1885-1892

[2] Barr, R. S., Golden, B. L., Kelly, J. P., Resende, M. G. C., Stewart Jr., W. R.: Designing and Reporting on Computational Experiments with Heuristic Methods. Journal of Heuristics, 1 (1995) 9-32

[3] Cohen, P. R., Empirical Methods for Artificial Intelligence. MIT Press (1995)

[4] Cox, D.R. and Hinkley, D.V. Theoretical statistics. London: Chapman & Hall (1974)

[5] Dietterich, T. G.: Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. Neural Computation 10 (7) (1998) 1895-1923

[6] Haykin, S. Neural Networks, A Comprehensive Foundation.Prentice Hall, 1999

[7] Kohavi, R.: A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. Proceedings of International Joint Conference on Artificial Intelligence (1995)

[8] Mann, H. B. and Whitney, D. R. On a test of whether one of two random variables is stochastically larger than the other. Ann. Math. Statist. 18, (1947) 50-60.

[9] Salzberg S. L.: On Comparing Classifiers: Pitfalls to Avoid and a Recommended Approach. Data minig and Knowledge Discovery 1 (1997) 317-328

[10] Stone, M.: Cross-validatory choice and assesment of statistical predictions. J. Roy. Statist. Soc. 36 (1974) 111-147

[11] Tjen-Sien Lim, Wei-Yin Loh, Yu-Shan Shih: A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-Three Old and New Classification Algorithms. Machine Learning 40(3) (2000) 203-228

[12] Whitley D., Watson J. P., Howe A., Barbulescu L.: Testing, Evaluation and Performance of Optimization and Learning Systems. Keynote Address: Adaptive Computing in Design and Manufacturing (2002)

[13] Wilcoxon, F. Individual Comparisons by Ranking Methods. Biometrics 1, (1945) 80-83

[14] Efron, B. and Tibshirani, R. An introduction to bootstrap. New York. Chapman & Hall. (1993)

[15] Davison, A.C., Hinkley, D. V. Bootstrap Methods and Their Application. Cambridge University Press (1997)

[16] Jain, A. K., Dubes, R. C., Chen, C.C. Bootstrap Techniques for Error Estimation. IEEE Trans. PAMI, (9) 5 628-633 (1987)

[17] Weiss, S. Small Sample Error Rate Estimation for k-NN Classifiers. IEEE Trans. PAMI, (13) 3 285-289 (1991)

[18] R. Tibshirani. Bias, variance and prediction error for classification rules. Technical report, Department of Statistics, University of Toronto, 1996