LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

INSTITUT FÜR STATISTIK

Manuel J. A. Eugster & Friedrich Leisch

Bench Plot and Mixed Effects Models: First steps toward a comprehensiv benchmark analysis toolbox

Technical Report Number 026, 2008
Department of Statistics
University of Munich

http://www.stat.uni-muenchen.de

Bench Plot and Mixed Effects Models: First steps toward a comprehensiv benchmark analysis toolbox

Manuel J. A. Eugster and Friedrich Leisch

Department of Statistics, Ludwig-Maximilians-Universität München,
Ludwigstrasse 33, 80539 München, Germany,
firstname.lastname@stat.uni-muenchen.de

Abstract. Benchmark experiments produce data in a very specific format. The observations are drawn from the performance distributions of the candidate algorithms on resampled data sets. In this paper we introduce new visualisation techniques and show how formal test procedures can be used to evaluate the results. This is the first step towards a comprehensive toolbox of exploratory and inferential analysis methods for benchmark experiments.

Keywords: benchmark experiments, visualisation, hypothesis tests

This is a pre-print of an article which has been accepted for the *Compstat 2008-Proceedings in Computational Statistics*.

1 Introduction

In statistical learning, benchmark experiments are empirical experiments with the aim of comparing and ranking algorithms with respect to a certain performance measure. New benchmark experiments are published on almost a daily basis. Especially in the machine learning community benchmarking is the primary method of choice to evaluate new learning algorithms. However, there are surprisingly few publications on *how* to evaluate benchmark experiments. Some newer exceptions are Hothorn et al. (2005), Demsar (2006), Yildiz and Alpaydin (2006) and Hornik and Meyer (2007).

Hothorn et al. (2005) use the bootstrap method as a sampling scheme such that the resulting performance observations are iid and can be analyzed using standard statistical methods. However, their paper describes a general framework, not precise instructions for a concrete benchmark experiment. To use a metaphor, it describes how to cook in general, but contains no recipes for a nice dinner. Using the foundations laid out by the general framework,

our goal is now to implement a toolbox of exploratory and inferential methods for the analysis of benchmark experiments.

Due to space restrictions, we cannot give a comprehensive overview of all our work in this direction in this paper. Hence, we chose to describe one new visualization technique (the benchplot), and how benchmark data can be seen as coming from a blocked design and analyzed as such (using mixed effects models) as examples. All computations are done using R (R Development Core Team, 2007), the corresponding R functions are part of an R package for the analysis of benchmark experiments which is currently under development and will be released on CRAN later this year.

Following Hothorn et al. (2005), we set up a regression benchmark experiment with the mean squared error as loss function. Given a data set $\mathfrak{L} = \{z_1, \ldots, z_n\}$, we draw B learning samples using sampling with replacement

$$\mathfrak{L}^i = \{z_1^i, \ldots, z_n^i\}$$

for $i = 1, \ldots, B$ (bootstrap). Furthermore we assume that there are $K > 1$ candidate algorithms a_k $(k = 1, \ldots, K)$ available for the solution of the underlying problem. For each algorithm a_k the function $a_k(\cdot \mid \mathfrak{L}^b)$ is the fitted model based on the sample $\mathfrak{L}^b$. This function itself has a distribution $\mathcal{A}_k$ as it is a random variable depending on $\mathfrak{L}^b$:

$$a_k(\cdot \mid \mathfrak{L}^b) \sim \mathcal{A}_k(\mathfrak{L}),\ k = 1, \ldots, K$$

The performance of the candidate algorithm a_k when provided with the training data $\mathcal{L}^b$ is measured with the mean squared error function p (a scalar function):

$$p_{kb} = p(a_k, \mathfrak{L}^b) \sim \mathcal{P}_k = \mathcal{P}_k(\mathfrak{L})$$

The p_{kb} are samples drawn from the distribution $\mathcal{P}_k(\mathfrak{L})$ of the mean squared error of the algorithm k on the data set $\mathfrak{L}$. As we are not able to calculate p_{kb} analytically, we have to use the empirical analogue $\hat{p}_{kb}$ based on a test sample $\mathfrak{T}$. A common choice to define $\mathfrak{T}$ is in terms of out-of-bootstrap observations: $\mathfrak{T} = \mathfrak{L} \setminus \mathfrak{L}^b$. This leads to non-independent observations of the performance measure, but their correlation vanishes as n tends to infinity.

The first step is to analyse the benchmark experiment in an exploratory way. Based on findings in this step, the second step tests hypothesis of interest and yields an ordered ranking of the candidate algorithms.

To demonstrate the methods, we use an examplar benchmark study using the motorcycle data set, see Figure 1. The candidate algorithms used (with corresponding R functions in parenthesis) are linear regression (`lm`), nonlinear least-squares regression (`nls`), neural networks (`nnet`), regression trees (`rpart`), generalized additive models (`gam`), loess regression (`loess`) (all, e.g., in Venables and Ripley, 2002), and boosted generalized additive models (`gamboost`, Hothorn & Bühlmann, 2006) as candidate algorithms. In order to

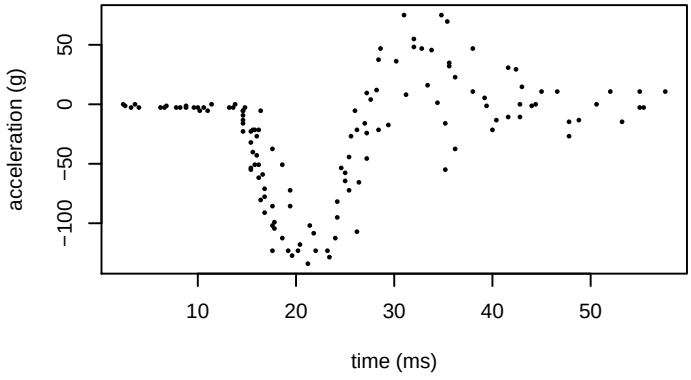

Fig. 1. The motorcycle data set (Silverman (1985)): time and head acceleration of a PTMO (post mortem human test object) after a simulated impact with motorcycles. The number of observations is $n = 133$.

	Mean	SD	95% CI	Median	IQR
`nnet`	1438.1	868.4	$[-263.9, 3140.1]$	977.2	1697.1
`lm`	2209.2	294.1	$[1632.8, 2785.5]$	2209.8	384.1
`rpart`	812.4	181.2	$[457.2, 1167.6]$	809.2	248.8
`gamboost`	583.7	116.5	$[355.2, 812.1]$	582.1	151.4
`gam`	565.2	122.6	$[324.9, 805.6]$	563.6	138.1
`nls`	1818.1	242.3	$[1343.2, 2292.9]$	1808.5	307.6
`loess`	604.3	134.6	$[340.4, 868.1]$	596.6	169.2

Table 1. Common summary statistics of the example experiment: based on the 250 benchmark experiment runs, the mean, standard deviance (SD), 95% confidence interval, median and interquartiles range (IQR) of the empirical mean squared error distributions are calculated.

present an experiment with a wide variety of algorithm performances, we included linear regression, although the data are clearly nonlinear. The number of bootstrap samples B is 250.

2 Exploratory analysis

Common analyses of benchmark experiments consist of the comparison of the empirical performance measure distributions based on some summary statistics. Table 1 shows the most established ones. In many cases, these heavily compacted numbers are the only analysis and basis for a ranking of the algorithms. But in doing so, one loses a lot of interesting and primarily important information about the experiment.

Based on the mean performance values, the order of the candidate algorithms is `gam` $<$ `gamboost` $<$ `loess` $<$ `rpart` $<$ `nnet` $<$ `nls` $<$ `lm`. As indication for the significance of differences, one can use the corresponding

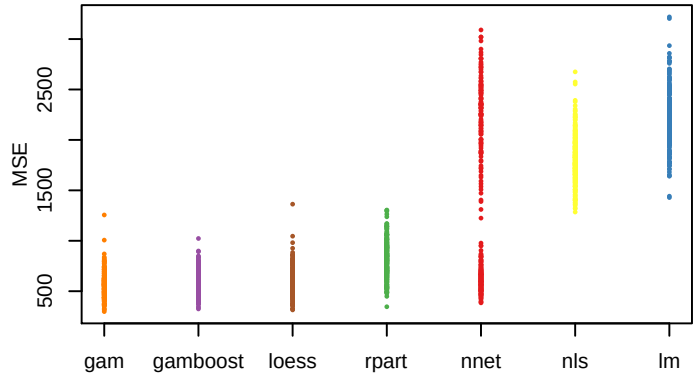

Fig. 2. Dot plot of the example experiment: the performance of each algorithm on each benchmark run is shown as a dot.

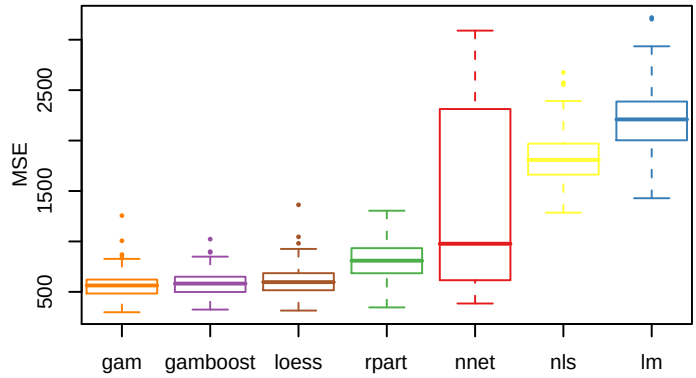

Fig. 3. Box plot of the example experiment: outliers are identified. In comparison to the dot plot, information about local minima is lost.

95% confidence intervals. For our data the mean is approximately equal to the median for all except `nnet`, i.e. the performance of the latter seems to be skewed. Figure 2 shows a dot plot with the algorithms on the abscissa (sorted after their mean performance) and their performances on the ordinate, represented with a dot for each benchmark run (i.e., bootstrap sample). It can bee seen that the distribution for `nnet` is not only skewed, but also bimodal. Manual replications of fitting neural networks to the data show that training often gets stuck in local minima. Figure 3 shows a box plot with a box for each algorithm. This plot allows the indication of outlier performances, but information about local minima is lost.

Both, Figure 2 and 3, also give an idea about the overall order of the algorithms. `gam`, `gamboost` and `loess` have basically the same performance, the small differences in mean performance being caused mostly by a few

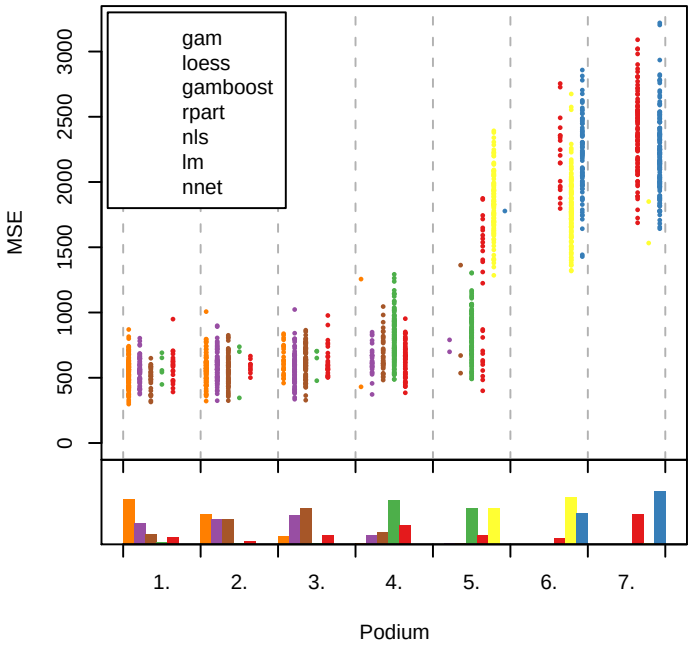

Fig. 4. Benchmark experiment plot of the example: the abscissa is a podium 7 places. For each benchmark run, the algorithms are sorted according to their performance values and a dot is drawn on the corresponding place. To visualise the count of an algorithm on a specific position, a bar plot is shown for each of podium places.

outliers. `rpart` has slightly worse performance, but with an isolated point close to the best value of the other three algorithms. `nls` and `lm` are in the upper MSE range, whereas `nls` has a lower minimal value as `lm`, but similar variance. `lm` also has some outliers near to the minimal value from `nls`. As said above, `nnet` ranges in two areas, whereas the lower MSE range (which corresponds to good performance) is similar to `gam`, `gamboost` and `loess`.

One massive problem of the dot plot is the overdrawing of dots. We do not know how many "lower" outliers of `rpart` there are, but the number of them really influences the impression of an order. This could be partly solved by jittering the plots, i.e., adding some random noise to the data. Additionally, the standard dot plot suggests the independence of the bootstrap samples. Indeed we know that, for example, `gam`, `gamboost` and `loess` perform similarly over all benchmark runs, but we do not know their ranking per benchmark run, which algorithm is on which rank and how often. The benchmark experiment plot was developed to overcome these limitations and to get a better understanding of benchmark experiments.

Instead of random jittering, we use the ranks of the algorithms on each bootstrap sample to horizontally "stretch out" the dots. For each benchmark run, the algorithms are ordered according to their performance value, and we draw seperate dot plots for each rank, ties are broken at random. This can be seen as creating a "podium" with K places, and having separate dot plots for each podium place, see Figure 4. Note that the plot is much easier to read when in color.

While the mean performances of `gam`, `gamboost` and `loess` (as shown in Table 1), and the performace distributions of these three (as shown in Figure 2 and Figure 3) all look very similar, we see in Figure 4 that `gam` is by far most often the best algorithm for single bootstrap samples. Another aspect that is impossible to infer from the marginal distributions of the performance measures alone is that there are a few bootstrap samples where `rpart` works best.

The dots in Figures 2 and 4 are not independent from each other, because all algorithms were evaluated on each bootstrap sample. This dependency can be displayed by connecting the dots corresponding to one bootstrap sample with a line, resulting in a modified version of a parallel coordinates plot. In our implementation, the line segment between two podium places is drawn with the color of the algorithm in the lower position. To overcome the problem of overdrawing lines we use transparency (alpha shading). In this "full benchmark experiment plot" one can also see correlations between algorithm performances (parallel vs. crossing lines). In greyscale the plot looks like a big mess of grey lines and dots, and hence had to be excluded from this manuscript (a color version is available from `http://www.statistik.lmu.de/~eugster/`).

3 Inference

To make a statistically correct order and ranking we need more formal tools: statistical inference and primarily the testing of hypothesis provides them. The design of a benchmark experiment is a random block design. This type of experiment has two classification factors: the experimental one, for which we want to determine systematic differences, and the blocking one, which represents a known source of variability. In terms of benchmark experiments, the experimental factor is the set of algorithms and the blocking factor is that all algorithms perform on the same bootstrap samples.

We use a mixed effects model (e.g. Pinheiro and Bates, 2000) to analyze the output of a benchmark experiment. The variable of primary interest, i.e., the set of algorithms, is modelled as fixed effect β_j. The blocking factor, i.e. the sampling, is modelled as random effect b_j:

$$p_{ij} = \beta_0 + \beta_j + b_i + \epsilon_{ij}$$

with $b_i \sim N(0, \sigma_b^2)$, $\epsilon_{ij} \sim N(0, \sigma^2)$ and $i = 1, \ldots, B$, $j = 1, \ldots K-1$. Hence, we estimate only one parameter σ_b^2 for the effect of the data set. A modelling,

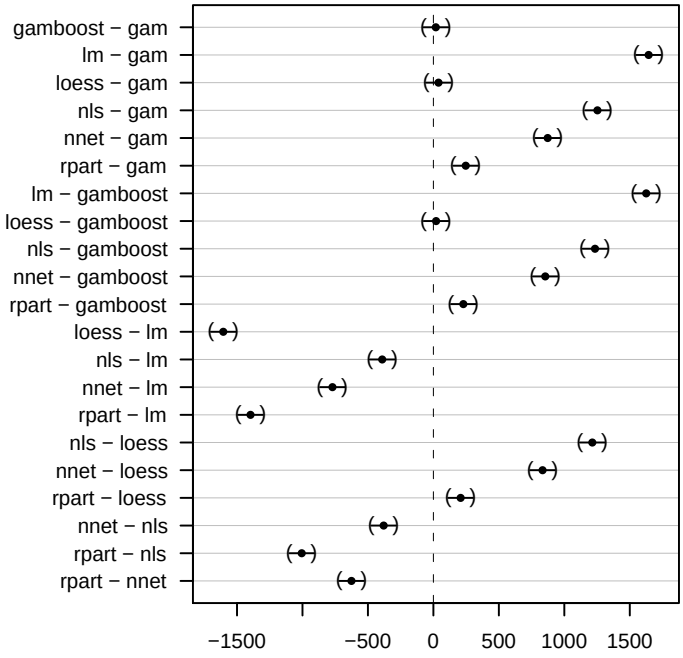

Fig. 5. Simultaneous 95% confidence intervals for multiple comparisons of means using Tukey constrast based on the mixed effects model of the example experiment.

by contrast, with the effect of the data set as main effect, would have lead to B parameters. Since we are able to draw as many random samples B from the performance distributions as required, we can rely on asymptotic normal theory. In case of our example the estimates for the parameters have been calculated as

$$\hat{\sigma}_b = 121.31,\ \hat{\sigma} = 353.73,$$

and

Intercept	Δ`gamboost`	Δ`lm`	Δ`loess`	Δ`nls`	Δ`nnet`	Δ`rpart`
$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\beta}_4$	$\hat{\beta}_5$	$\hat{\beta}_6$
565.24	18.41	1643.91	39.03	1252.85	872.86	247.16,

with Δ denotes the difference between the Intercept and the corresponding algorithm.

The global test, whether there are any differences between the algorithms which do not come from the sampling, can be performed with ANOVA and the F-test. For our model this test rejects the null hypothesis that all algorithms have the same performance. We then use Tukey contrasts to test pairwise differences. Figure 5 shows the corresponding 95% family-wise confidence intervals. The differences between `gam`, `gamboost` and `loess` are not

significant, the corresponding confidence intervals intersect 0 and overlap each other. As we can not establish a strict total order $<$ or a total order $\leq$, we define a reflexive and symmetric order relation $\approx$: two algorithms are $\approx$-related if their difference is not significant. The differences between all other algorithms are significant and we can establish a strict total order $<$ for each pair. Based on this set of ordered pairs ($\approx$- and $<$- ordered) we can use use a topological sort to define an overall order of the algorithms. In case of our benchmark experiment example, the final order of the candidate algorithms is `gam` $\approx$ `loess` $\approx$ `gamboost` $<$ `rpart` $<$ `nnet` $<$ `nls` $<$ `lm`.

4 Summary and future work

In this paper we gave a short introduction to our current work on formal statistical analysis of benchmark experiments and introduced the benchmark experiment plot as a new visualisation method. The random block design of a benchmark experiment has been modelled using mixed effects. This allows to test various hypothesis of interest, amongst others, the pairwise differences. We introduced an order relation for algorithms with non-significant differences, and inferred a statistically correct order of the candidate algorithms.

This paper is the first step towards a comprehensive toolbox for exploratory and inferential analysis of benchmark experiments. There are lots of things to do. Two examples are (1) sequential testing to reduce computation time and (2) alternative order mechanisms like the minimax principle. Besides the analysis of a set of candidate algorithms on one data set, the extension to a set of data sets is obvious.

Acknowledgments

The authors want to thank Torsten Hothorn for discussions and ideas.

References

DEMSAR, J. (2006): Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research 7, 1–30.*

HORNIK, K. and MEYER D. (2007): Deriving consensus rankings from benchmarking experiments. In: R. Decker and H.-J. Lenz (Eds.): *Advances in Data Analysis.* Springer-Verlag, 163–170.

HOTHORN, T. and BÜHLMANN, P. (2006): Model-based boosting in high dimensions. *Bioinformatics 22 (22), 2828–2829.*

HOTHORN, T., LEISCH, F., ZEILEIS, A. and HORNIK, K. (2005): The design and analysis of benchmark experiments. *Journal of Computational and Graphical Statistics 14 (3), 675–699.*

PINHEIRO, J. C. and BATES, D. M. (2000): *Mixed-Effects Models in S and S-PLUS.* Springer-Verlag.

R DEVELOPMENT CORE TEAM (2007): *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

SILVERMAN, B.W. (1985): Some aspects of the spline smoothing approach to non-parametric regression curve (with discussion). *Journal of the Royal Statistical Society Series B (47), 1–52*.

VENABLES, W. and RIPLEY B. (2002): *Modern Applied Statistics with S*. Springer-Verlag.

YILDIZ, O. T. and ALPAYDIN, E. (2006): Ordering and finding the best of k>2 supervised learning algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (3), 392–402*.