

Benchmark experiments

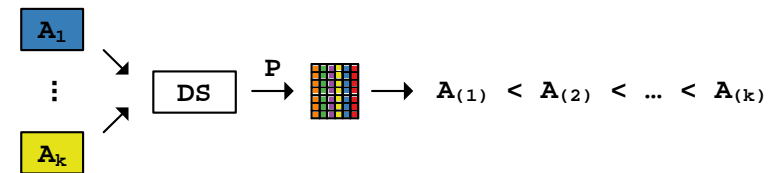
Exploratory and Inferential Analysis of Benchmark Experiments

Manuel J. A. Eugster and Friedrich Leisch

Department for Statistics
Ludwig-Maximilians-Universität München

useR!, 2008

Most popular scenario:

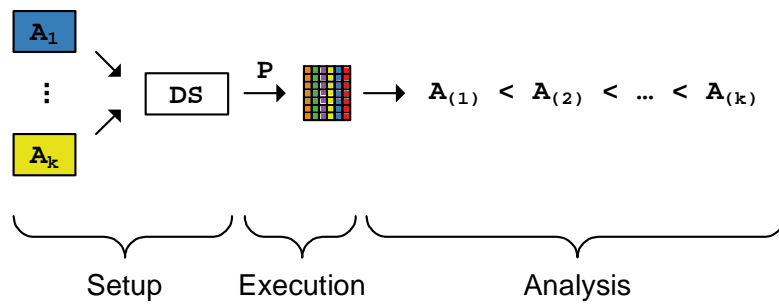


1 / 23

2 / 23

Benchmark experiments

Most popular scenario:



Implementation

3 / 23

4 / 23

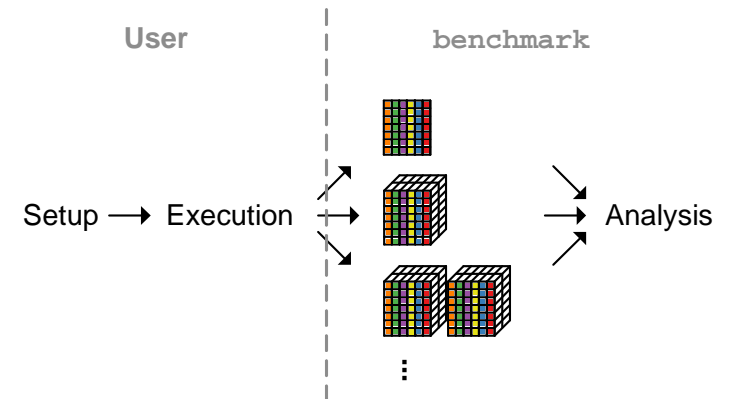
Setup and Execution layers

Domain-specific language to describe the elements of benchmark experiments using small bricks.

But ... even in our working group we have *supervised*, *cluster* and *bicluster* problems and until now it seems to be hard to reconcile them in “one language”. We have developed some rudiments, but it seems to be more manageable if the user writes the concrete problem-specific “loop” by his own.

5 / 23

Package coverage



6 / 23

“Enter the benchmark”

Benchmark experiment:

(1) classification problems {BreastCancer, monks3, musk}; (2) algorithms {lda, naiveBayes, knn, rpart, svm, nnet}; (3) misclassification; (4) bootstrap 250 samples; (5) out-of-bootstrap samples;

List of performance matrices:

```
> uciraw$monks3
      lda   nb   knn  rpart   svm   nnet
[1,] 0.0390 0.0390 0.0488 0.0195 0.0195 0.0195
[2,] 0.0498 0.0498 0.0299 0.0149 0.0149 0.0149
...
```

7 / 23

“Enter the benchmark”

```
> library(benchmark)

Loading required package: reshape
Loading required package: relations
Loading required package: sets
Loading required package: lattice
```

8 / 23

“Enter the benchmark”

```
> uci <- as.bench(uciraw, perf='Misclassification')
```

Benchmark experiment

samples	algorithms	performances	data sets
250	6	1	3

Coercing: `as.bench` tries to capture the manifoldness of raw benchmark experiment data.

The bench object

Subsets: [`samp`, `alg`, `perf`, `ds`] or subset.

```
> monks3 <- uci[, , 'monks3']
```

Benchmark experiment

samples	algorithms	performances	data sets
250	6	1	1

Reshape: `melt` melts an object into a form suitable for easy casting (see `reshape` package).

```
> melt(monks3)
```

samp	alg	perf	ds	value
1	1 lda	Misclassification	monks3	0.0390
2	2 lda	Misclassification	monks3	0.0498

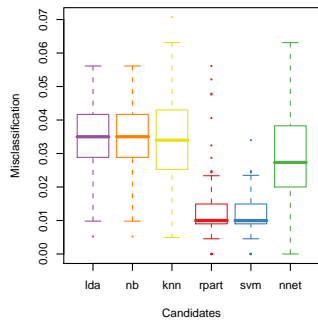
...

9 / 23

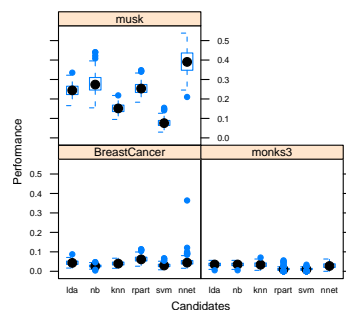
10 / 23

Basic plots

```
> boxplot(monks3)
```



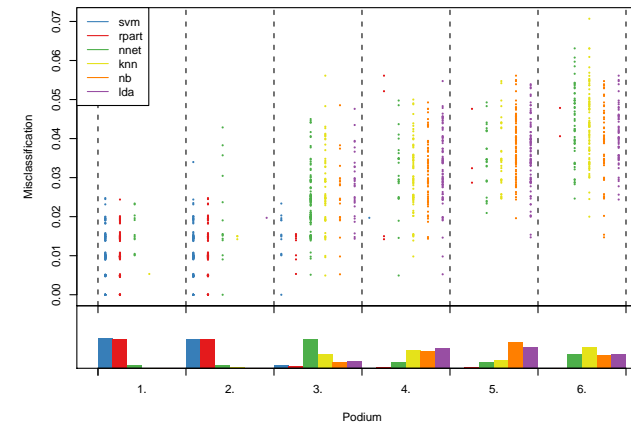
```
> bwplot(uci)
```



Other basic plots: `densityplot` and `stripplot`.

Benchmark experiment plot

```
> beplot(monks3)
```

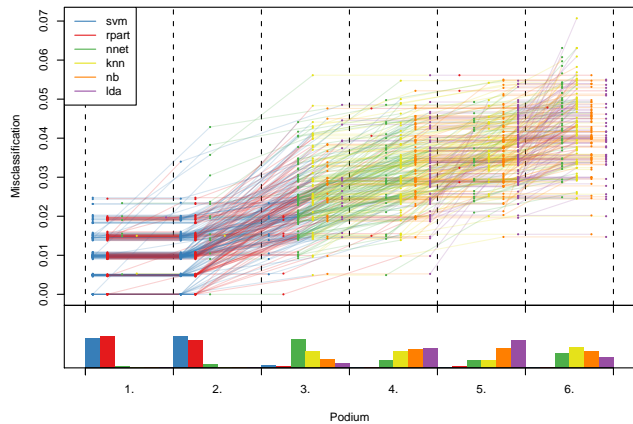


11 / 23

12 / 23

Benchmark experiment plot

```
> beplot(monks3, lines.show=TRUE)
```



13 / 23

Simple rankings

Mean performance:

```
> m <- apply(monks3, 'alg', mean)
      lda  nb  knn rpart  svm  nnet
0.0352 0.0353 0.0344 0.0116 0.0110 0.0293
```

```
> as.ranking(m)
```

```
      svm rpart  nnet  knn  lda  nb
      1    2    3    4    5    6
```

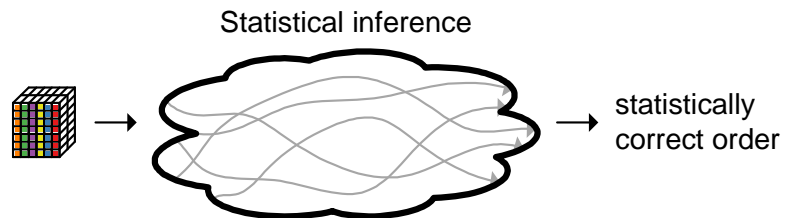
Minimax:

```
> as.ranking(apply(monks3, 'alg', max))
```

```
      svm  lda  nb rpart  nnet  knn
      1    2  2  2    5    6
```

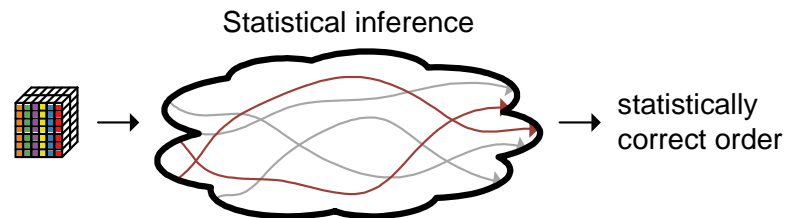
14 / 23

Inferential analysis



15 / 23

Inferential analysis



Implemented "paths":

1. based on linear mixed effects models.
2. based on Friedman-based rank tests.

16 / 23

The ibea object

The *inferential benchmark experiment analysis* frameworks encapsulate functions belonging to one “paths”.

```
> ibea <- make.lmer.ibea()
```

```
Loading required package: lme4  
Loading required package: Matrix  
Loading required package: multcomp  
Loading required package: mvtnorm
```

17 / 23

The “lmer-path”

```
> summary(ibeas)
```

```
Lmer inferential benchmark experiment analysis framework:
```

```
Available functions are
```

```
* model : function (bench)  
* relation : function (x, alpha)  
* relation.pairwise : function (test, alpha)  
* test.global : function (model)  
* test.pairwise : function (model)
```

18 / 23

The “lmer-path”

Individual steps:

1. `model(bench) → lme4::mer`
2. `test.pairwise(lme4::mer) → multcomp::glht`
3. `relation.pairwise(multcomp::glht, alpha) → relations::relation`

All-in-one:

```
> rel <- ibea$relation(monks3, 0.05)
```

A binary relation of size 6 x 6.

19 / 23

The “lmer-path”

Statistically correct order:

```
> ord <- tsort(rel)
```

```
rpart - svm < nnet < knn - lda - nb
```

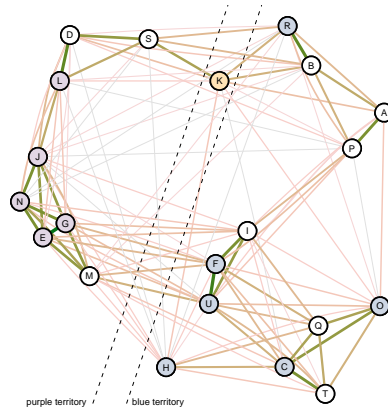
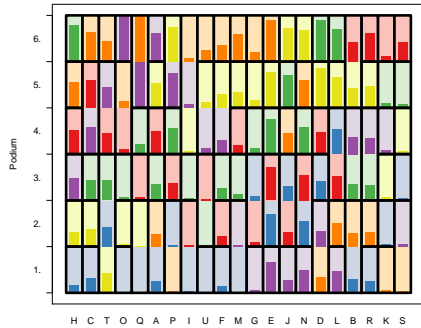
```
> as.ranking(ord)
```

rpart	svm	nnet	knn	lda	nb
1	1	3	4	4	4

20 / 23

Further benchmark functionality

Exploratory and inferential analysis assistance for benchmark experiments with more than one performance measure and/or more than one data set.



“Enter the benchmark”?!?

21 / 23

22 / 23

Take the red pill ...

... at <http://statistik.lmu.de/~eugster/benchmark/>.

Package:

benchmark version 0.01 – useR! 2008 source code release.

Reports:

Exploratory and Inferential Analysis of Benchmark Experiments.

Manuel J. A. Eugster, Torsten Hothorn and Friedrich Leisch. Technical Report 30, LMU Munich. **R supplement “The *uci621* benchmark experiment”.**

23 / 23