Benchmark Experiments – Comparing Statistical Learning Algorithms

Manuel J. A. Eugster

Institut für Statistik
Ludwig-Maximiliams-Universität München

September 28th 2009, TU Dortmund

Benchmark experiments

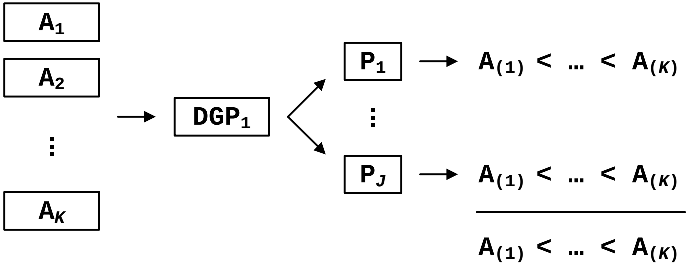
“Statistical decathlon”:
Empirical investigations with the aim of comparing and ranking (learning) algorithms with respect to certain performance measures.

In essence:
Draw observations from theoretically intractable performance distributions of the learning algorithms, which are defined by either artificial data generating processes or empirical distribution functions from one or more data sets.

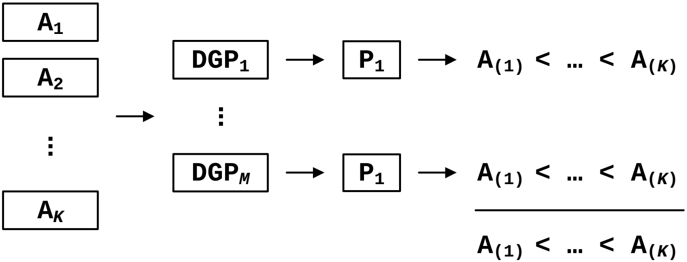
Scenario 1:

$$\begin{array}{c} \boxed{\mathtt{A_1}} \\ \boxed{\mathtt{A_2}} \\ \vdots \\ \boxed{\mathtt{A}_K} \end{array} \longrightarrow \boxed{\mathtt{DGP_1}} \longrightarrow \boxed{\mathtt{P_1}} \longrightarrow \mathtt{A_{(1)}} < \ldots < \mathtt{A}_{(K)}$$

Scenario 2:



Scenario 3:



Abstract levels of benchmark experiments:

Setup: define design of a benchmark experiment; data sets, candidate algorithms, performance measures and resampling strategy.

Execution: execute Setup; parallel computation, monitoring, sequential and adaptive procedures.

Analysis: exploratory and inferential analyses of the raw performance measures; main objective is a statistically correct order of the candidate algorithms.

Scenario 1

"From the theoretical framework to a concrete order of some candidate algorithms"

$\boxed{A_1}$

$\boxed{A_2}$

$\vdots$

$\boxed{A_K}$

$\longrightarrow \boxed{DGP_1} \longrightarrow \boxed{P_1} \longrightarrow A_{(1)} < \ldots < A_{(K)}$

General framework

Data generating process:

Given a data generating process DGP, we draw B independent and indentically distributed learning samples:

$$\mathfrak{L}^1 = \{z_1^1, \ldots, z_n^1\} \sim DGP$$
$$\vdots$$
$$\mathfrak{L}^B = \{z_1^B, \ldots, z_n^B\} \sim DGP$$

Candidate algorithms:

There are $K > 1$ algorithms a_k $(k = 1, \ldots, K)$ with the function $a_k(\cdot \mid \mathfrak{L}^b)$ the fitted model on the learning sample $\mathfrak{L}^b$ $(b = 1, \ldots, B)$:

$$a_k(\cdot \mid \mathfrak{L}^b) \sim \mathcal{A}_k(DGP)$$

Performance measure:

Performance of algorithm a_k when provied with the learning sample $\mathfrak{L}^b$ is measured by a scalar function p:

$$p_{kb} = p(a_k, \mathfrak{L}^b) \sim P_k = P_k(DGP)$$

Supervised learning problems

Observations are of the form $z = (y, x)$ where y denotes the response variable and x a vector of input variables.

Algorithms are learners $\hat{y} = a_k(x \mid \mathfrak{L}^b)$ which, given the input variables, predict the unknown response variable.

Performance is defined by some functional μ of the distribution of a loss function $L(y, \hat{y})$ which measures the discrepancy between true response y and predicted value $\hat{y}$:

$$p_{kb} = p(a_k, \mathfrak{L}^b) = \mu(L(y, a_k(x \mid \mathfrak{L}^b))) \sim P_k(DGP)$$

Performance is defined by some functional μ of the distribution of a loss function $L(y, \hat{y})$ which measures the discrepancy between true response y and predicted value $\hat{y}$:

$$p_{kb} = p(a_k, \mathfrak{L}^b) = \mu(L(y, a_k(x \mid \mathfrak{L}^b))) \sim P_k(DGP)$$

Misclassification:
The loss function is

$$L(y, \hat{y}) = \begin{cases} 0, & y = \hat{y} \\ 1, & \text{otherwise} \end{cases}$$

and the functional μ is the expectation E:

$$p_{kb} = \mathsf{E}_{a_k} \mathsf{E}_{z=(y,x)} L(y, a_k(x \mid \mathfrak{L}^b))$$

with $z = (y, x)$ drawn from the same data generating process as the observations in the learning sample $\mathfrak{L}^b$.

If it is not possible to calculate μ analytically, the empirical analogue $\mu_{\mathfrak{T}}$ based on a test sample $\mathfrak{T} \sim DGP$ has to be used:

$$\hat{p}_{kb} = \hat{p}(a_k, \mathfrak{L}^b) = \mu_{\mathfrak{T}}(L(y, a_k(x \mid \mathfrak{L}^b))) \sim \hat{P}_k(DGP)$$

If it is not possible to calculate μ analytically, the empirical analogue $\mu_{\mathfrak{T}}$ based on a test sample $\mathfrak{T} \sim DGP$ has to be used:

$$\hat{p}_{kb} = \hat{p}(a_k, \mathfrak{L}^b) = \mu_{\mathfrak{T}}(L(y, a_k(x \mid \mathfrak{L}^b))) \sim \hat{P}_k(DGP)$$

Misclassification:
The empirical functional $\mu_{\mathfrak{T}}$ is the expectation E with respect to the test sample $\mathfrak{T}$:

$$\hat{p}_{kb} = \mathsf{E}_{a_k} \mathsf{E}_{z=(y,x)} L(y, a_k(x \mid \mathfrak{L}^b))$$

with $z = (y, x) \in \mathfrak{T}$.

Data generating process in a real world problem:

Given is one learning sample $\mathfrak{L} \sim \mathcal{Z}_n$ of n observations from some distribution function $\mathcal{Z}$. Then the empirical distribution function $\hat{\mathcal{Z}}_n$ covers all knowledge about the data generating process: $DGP = \hat{\mathcal{Z}}_n$.

Learning samples are defined by bootstrapping: $\mathfrak{L}^b \sim \hat{\mathcal{Z}}_n$.

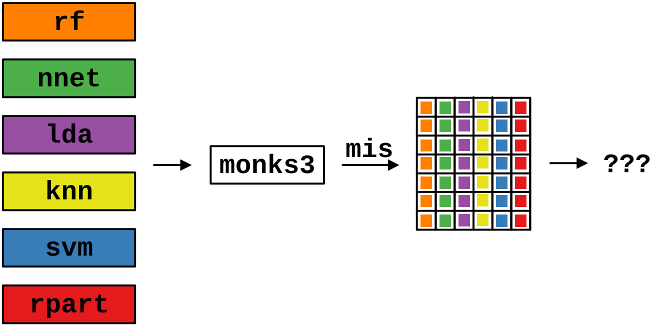
Test samples are defined by the out-of-bootstrap observations: $\mathfrak{T}^b = \mathfrak{L} \setminus \mathfrak{L}^b$.

Empircial performance distributions:

For each candidate algorithm a_k the empirical performance distribution $\hat{P}_k(\mathfrak{L})$ is estimated based on the $\hat{p}_{kb}$.

Exploratory data analysis tools and formal inference procedures to compare them and calculate a statistically correct order.

Exemplar benchmark experiment



(1) classification problem `monks3` (554 observations and 6 nominal attributes); **(2)** algorithms {`randomForest`, `nnet`, `lda`, `knn`, `svm`, `rpart`}; **(3)** misclassification performance measure; **(4)** 250 bootstrap learning samples; **(5)** out-of-bootstrap test samples;

Common practise

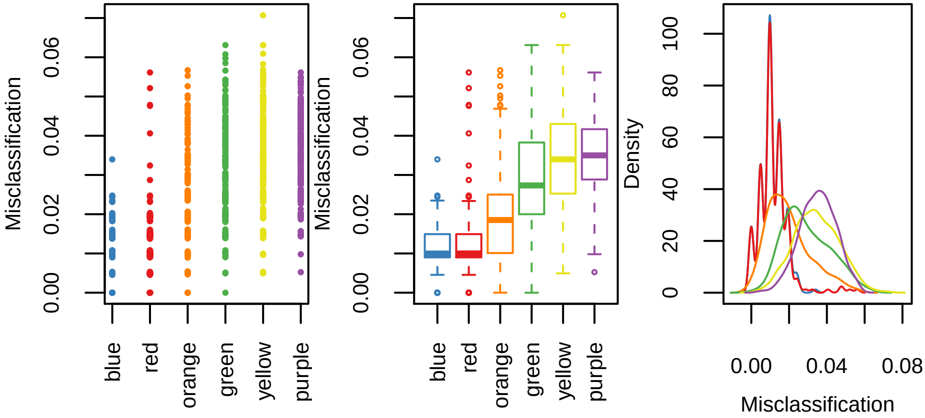
Comparison based on some summary statistics: algorithm a_1 is better than algorithm a_2 iff $\phi(\hat{P}_1) < \phi(\hat{P}_2)$.

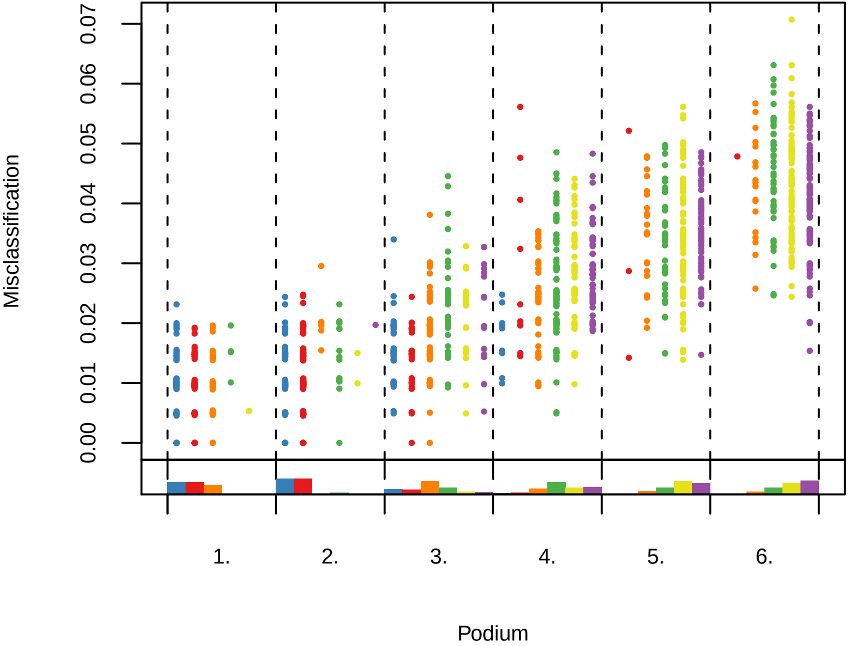
$\phi =$	Mean	SD	Median	Max
`purple`	0.0352	0.0094	0.0350	0.0561
`orange`	0.0197	0.0117	0.0185	0.0567
`yellow`	0.0344	0.0118	0.0340	0.0707
`red`	0.0116	0.0080	0.0100	0.0561
`blue`	0.0110	0.0059	0.0100	0.0340
`green`	0.0293	0.0123	0.0273	0.0631

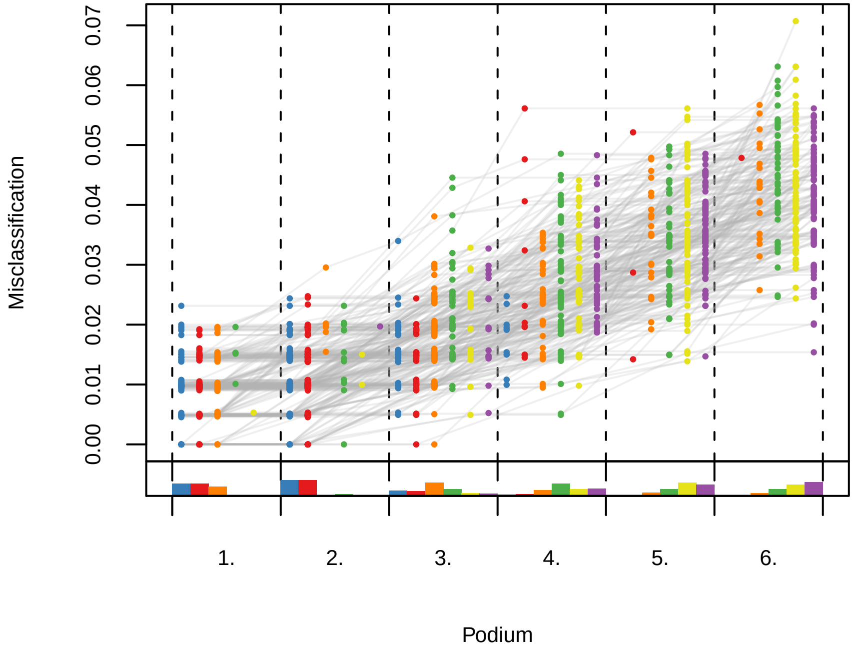
Order based on Mean:

$$\texttt{blue} < \texttt{red} < \texttt{orange} < \texttt{green} < \texttt{yellow} < \texttt{purple}$$

Exploratory analysis




Misclassification
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
1.
2.
3.
4.
5.
6.
Podium


0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
Misclassification
1.
2.
3.
4.
5.
6.
Podium

Inferential analysis

Random block design:

The set of algorithms as experimental factor κ_j, a learning sample as the blocking factor b_i; κ_0 the intercept which expresses the basic performance:

$$p_{ij} = \kappa_0 + \kappa_j + b_i + \epsilon_{ij},$$
$$i = 1, \ldots, B, j = 1, \ldots (K - 1)$$

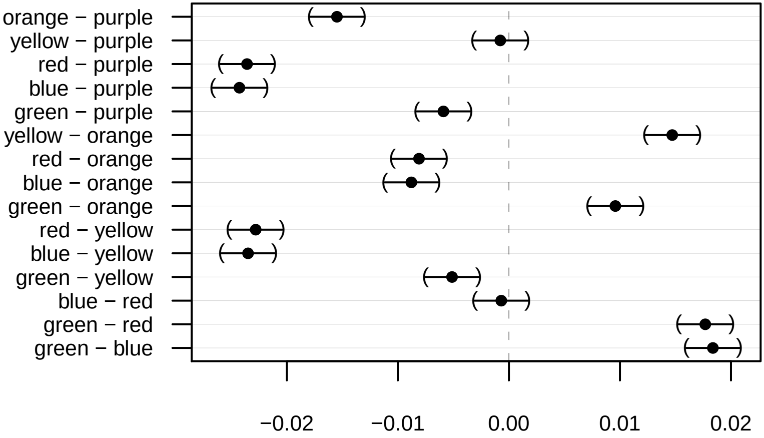
Hypothesis of interest:

$$H_0: \ \kappa_1 = \cdots = \kappa_{K-1} = 0,$$
$$H_A: \ \exists j: \ \kappa_j \neq 0.$$

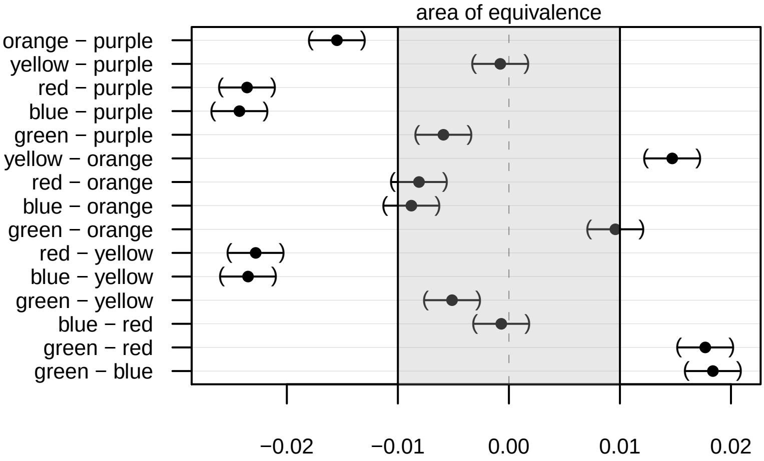
Testing using parametric and non-parametring methods (which take different assumptions on κ_j, b_i and ϵ_{ij}).

Mixed effects model:
k_j fixed effect, b_i random effect and $b_i \sim N(0, \sigma_b^2)$, $\epsilon_{ij} \sim N(0, \sigma^2)$.

Pairwise comparisons with Tukey contrasts; calculation of simultaneous confidence intervals (95%), which allow controlling the experimentwise error.



Pairwise comparisons with Tukey contrasts; calculation of simultaneous confidence intervals (95%), which allow controlling the experimentwise error.



Relations

Equivalence:

Based on the *p*-values: $a_i \approx a_j$ iff their difference is not significant (relevant).

Properties $(i, j, k = 1, \ldots, K)$:

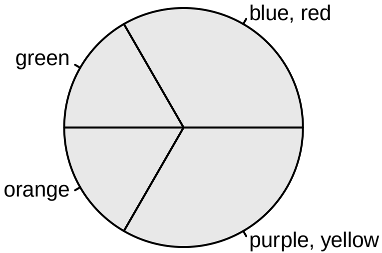
1. reflexive: $a_i \approx a_i$

2. symmetric: $a_i \approx a_j \Rightarrow a_j \approx a_i$

3? transitiv: $a_i \approx a_j \wedge a_j \approx a_k \Rightarrow a_i \approx a_k$

Equivalence classes (based on significance):

	purple	orange	yellow	red	blue	green
purple	1	0	1	0	0	0
orange	0	1	0	0	0	0
yellow	1	0	1	0	0	0
red	0	0	0	1	1	0
blue	0	0	0	1	1	0
green	0	0	0	0	0	1



Strict weak ordering:

Based on the p-values and the directions of the tests: $a_i < a_j$ iff a_i is significantly (or relevantly) better than a_j.

Properties ($i, j, k = 1, \ldots, K$):

1. irreflexive: $a_i \not< a_i$

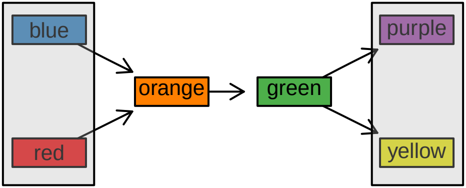
2. asymmetric: $a_i < a_j \Rightarrow a_j \not< a_i$

3? transitiv: $a_i \approx a_j \wedge a_j \approx a_k \Rightarrow a_i \approx a_k$

4? negatively transitiv: $a_i \not< a_j \wedge a_j \not< a_k \Rightarrow a_i \not< a_k$

Equivalence classes and their order (based on significance):

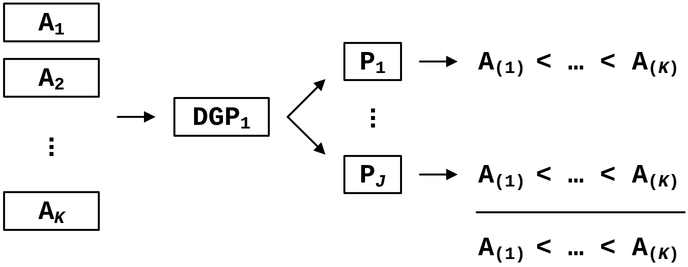
	purple	orange	yellow	red	blue	green
purple	0	0	0	0	0	0
orange	1	0	1	0	0	1
yellow	0	0	0	0	0	0
red	1	1	1	0	0	1
blue	1	1	1	0	0	1
green	1	0	1	0	0	0



$$\texttt{blue} \approx \texttt{red} < \texttt{orange} < \texttt{green} < \texttt{purple} \approx \texttt{yellow}$$

Scenario 2

“Aggregation of different performance measures”



Empirical performance distributions:

For each candidate algorithm a_k J empricial performance distributions $\hat{P}_k^j(\mathfrak{L})$ are estimated based on $\hat{p}_{kb}^j$.

Relation ensemble:

The exploratory and inferential analysis leads to an order relation for each performance measure:

$$\mathcal{R} = \{R_1, \ldots, R_J\}$$

Exemplar benchmark experiment

$\mathcal{R} = \{R_m, R_{mm}, R_c\}$ with R_m the relation based on the misclassification and the mixed effect model; R_{mm} the relation based on the minimax rule applied on the misclassification to control the worst-case scenario; and R_c the relation based on the computation time.

$$\begin{aligned} R_m &= \texttt{blue} \approx \texttt{red} < \texttt{orange} < \texttt{green} < \texttt{purple} \approx \texttt{yellow} \\ R_{mm} &= \texttt{blue} < \texttt{purple} \approx \texttt{red} < \texttt{orange} < \texttt{green} < \texttt{yellow} \\ R_c &= \texttt{red} < \texttt{purple} < \texttt{orange} < \texttt{yellow} < \texttt{green} < \texttt{blue} \end{aligned}$$

Consensus decision making

$$R_m = \texttt{blue} \approx \texttt{red} < \texttt{orange} < \texttt{green} < \texttt{purple} \approx \texttt{yellow}$$
$$R_{mm} = \texttt{blue} < \texttt{purple} \approx \texttt{red} < \texttt{orange} < \texttt{green} < \texttt{yellow}$$
$$R_c = \texttt{red} < \texttt{purple} < \texttt{orange} < \texttt{yellow} < \texttt{green} < \texttt{blue}$$

Aggregate the preferences of voters, i.e. the performance measures' orders of the candidate algorithms, in a way that "all voters can live with".

Borda count:
Relations as "voters ranks": the number of points given to an algorithm equates to the number of related algorithms.

On relation R:

$$\mathrm{borda}_R(a_i) = \#\{a_j \mid (a_i, a_j) \in R, j = 1, \ldots, K\}$$

On the set of relations $\mathcal{R}$ (w_R is a weighting factor):

$$\mathrm{borda}_{\mathcal{R}}(a_i) = \sum_{R \in \mathcal{R}} w_R \cdot \mathrm{borda}_R(a_i),$$

Final order R_{borda} is the order of the algorithms according to their Borda count.

Voters points:

	purple	orange	yellow	red	blue	green
R_m	0	3	0	4	4	2
R_{mm}	3	2	0	3	5	1
R_c	4	3	2	5	0	1
	7	8	2	12	9	4

Final ranking:

red	blue	purple	orange	green	yellow
1	2	3	4	5	6

$$R_{\text{borda}} = \texttt{red} < \texttt{blue} < \texttt{purple} < \texttt{orange} < \texttt{green} < \texttt{yellow}$$

Majority criterion:

If there is a single candidate preferred by a majority of voters to all other candidates, then that candidate should win.

Borda count fails:

$$
\begin{aligned}
51 \text{ voters: } & \texttt{a1} < \texttt{a2} < \texttt{a3} < \texttt{a4} \\
5 \text{ voters: } & \texttt{a2} < \texttt{a3} < \texttt{a4} < \texttt{a1} \\
23 \text{ voters: } & \texttt{a3} < \texttt{a2} < \texttt{a4} < \texttt{a1} \\
21 \text{ voters: } & \texttt{a4} < \texttt{a2} < \texttt{a3} < \texttt{a1}
\end{aligned}
$$

a1	a2	a3	a4
153	205	151	91

$$\texttt{a2} < \texttt{a1} < \texttt{a3} < \texttt{a4}$$

Condorcet methods:
Rank algorithm a_i above algorithm a_j iff the number of individual wins of a_i over a_j exceeds the number of losses.

Voting or Condorcet's paradoxon:

Collective preferences can be cyclic, i.e. not transitive, even if the preferences of individual voters are not.

Optimization approach:

Find the minimal solution of the problem

$$\sum_{R \in \mathcal{R}} w_R \cdot d_\Delta(R_{con}, R) \Rightarrow min_{R_{con} \in \mathcal{C}}.$$

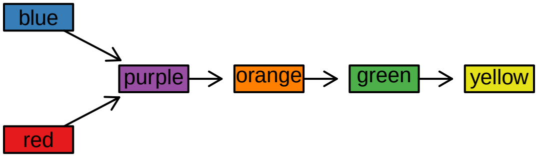
w_R is a weighting factor, and $\mathcal{C}$ defines a suitable set of possible consensus relations, e.g., the set of linear orders or the set of complete preorders.

$d_\Delta(R_1, R_2)$ is the symmetric difference distance:

$$\#\{(a_i, a_j) \mid ((a_i, a_j) \in R_1) \oplus ((a_i, a_j) \in R_2), i, j = 1, \ldots, K\}$$

$$R_m = \texttt{blue} \approx \texttt{red} < \texttt{orange} < \texttt{green} < \texttt{purple} \approx \texttt{yellow}$$
$$R_{mm} = \texttt{blue} < \texttt{purple} \approx \texttt{red} < \texttt{orange} < \texttt{green} < \texttt{yellow}$$
$$R_c = \texttt{red} < \texttt{purple} < \texttt{orange} < \texttt{yellow} < \texttt{green} < \texttt{blue}$$

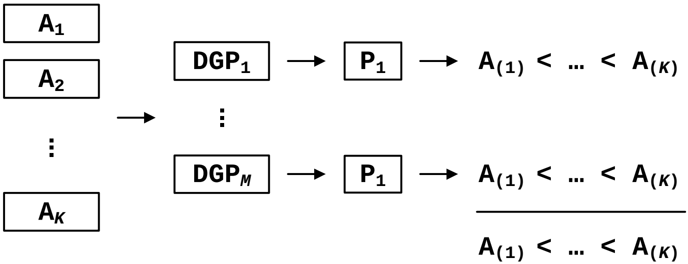
Optimization consensus over the set $\mathcal{C}$ of partial orders:



$$R_{con} = \texttt{blue} \approx \texttt{red} < \texttt{purple} < \texttt{orange} < \texttt{green} < \texttt{yellow}$$

Scenario 3

"More than one data set"



Domain of interest:
Given is a set of data sets $\mathcal{D} = \{\mathfrak{L}_1, \ldots, \mathfrak{L}_M\}$ consisting of M data sets representing the problem domain of interest.

For each data set $\mathfrak{L}_m$ ($m = 1, \ldots, M$) the benchmark experiment is executed and results in $J = 1 \cdot K$ estimations of empirical performance distributions $\hat{\mathcal{P}}_k^j(\mathfrak{L}_m)$ with $j = 1, \ldots, J$ and $k = 1, \ldots, K$.

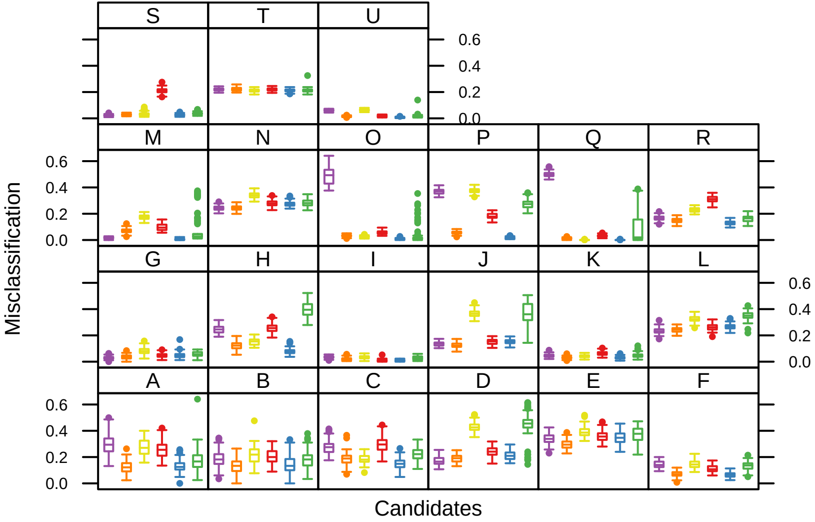
These raw results are aggregated to an order R_m of the candidate algorithms.

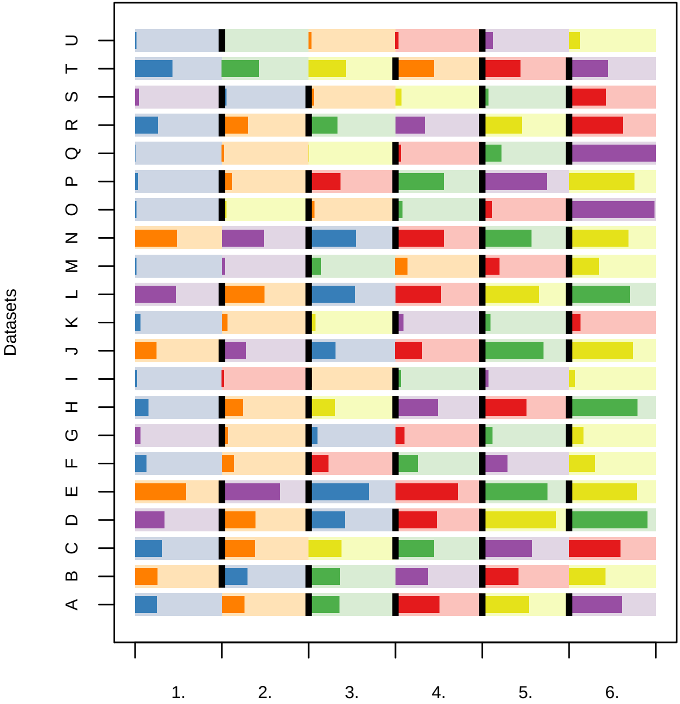
Exemplar benchmark experiment

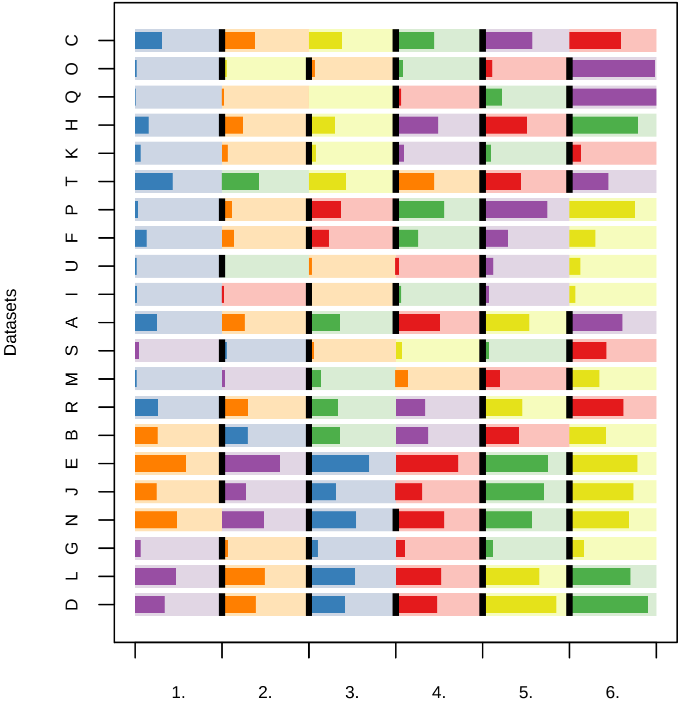
(1) 21 binary classificatin problems originated from the UCI Machine Learning repository: $\mathcal{D} = \{A, \ldots, U\}$ (`monks4` is letter `I`); **(2)** - **(5)** unchanged.

Execution results in 21×6 empirical performance distributions and 21 order relations.

Exploratory analysis




Datasets
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
1.
2.
3.
4.
5.
6.


Datasets
C
O
Q
H
K
T
P
F
U
I
A
S
M
R
B
E
J
N
G
L
D
1.
2.
3.
4.
5.
6.

Consensus ranking

Optimization consensus over the set $\mathcal{C}$ of partial orders:

$$\texttt{blue} \approx \texttt{orange} < \texttt{purple} < \texttt{green} < \texttt{red} < \texttt{yellow}$$

$$\texttt{blue} \approx \texttt{orange} < \texttt{purple} < \texttt{red} < \texttt{green} < \texttt{yellow}$$

Final order:

$$\texttt{blue} \approx \texttt{orange} < \texttt{purple} < \texttt{green} \approx \texttt{red} < \texttt{yellow}$$

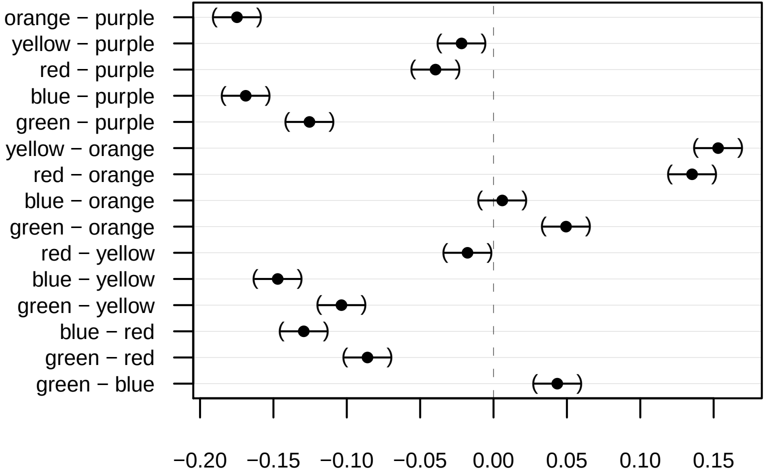
Inferential analysis

Design with two experimental factors, their interactions, and blocking factors at two levels:

$$p_{ijk} = \kappa_0 + \kappa_j + \gamma_k + \delta_{jk} + b_k + b_{ki} + \epsilon_{ijk}$$
$$i = 1, \ldots, B,\, j = 1, \ldots, (K-1),\, k = 1, \ldots, (M-1).$$

The experimental factors are described by κ and γ, their interactions by δ. κ_0 represents the basic performance of each algorithm, κ_j the individual differences from the basic performance. γ_k represents the data set effect. The blocking factors are described by b_k, the data sets, and b_{ki}, the sampling within the data sets. ϵ_{ijk} describes the systematic error.

Mixed effects model:



Final order:

$$\texttt{blue} \approx \texttt{orange} < \texttt{green} < \texttt{red} < \texttt{yellow} < \texttt{purple}$$

and finally ...

Statistical correct orders:
Statistical correct orders within the domain represented by the 21 data sets $\{A, \dots, U\}$:

$$R_{con} = \texttt{blue} \approx \texttt{orange} < \texttt{purple} < \texttt{green} \approx \texttt{red} < \texttt{yellow}$$
$$R_{mem} = \texttt{blue} \approx \texttt{orange} < \texttt{green} < \texttt{red} < \texttt{yellow} < \texttt{purple}$$

Papers and Software

... from `http://statistik.lmu.de/~eugster/benchmark`.

`R` **Package:**

`benchmark` available from R-Forge.

Papers and reports:

Manuel J. A. Eugster and Friedrich Leisch. *Bench plot and mixed effects models: First steps toward a comprehensive benchmark analysis toolbox*. In Paula Brito, editor, Compstat 2008-Proceedings in Computational Statistics, pages 299-306. Physica Verlag, Heidelberg, Germany, 2008.

Manuel J. A. Eugster, Torsten Hothorn and Friedrich Leisch. *Exploratory and Inferential Analysis of Benchmark Experiments*. Technical Report 30, LMU Municht. Submitted.