WIRTSCHAFTS UNIVERSITÄT WIEN

The Design and Analysis of Benchmark Experiments – Part I: Design

Achim Zeileis   Torsten Hothorn   Friedrich Leisch   Kurt Hornik

http://www.ci.tuwien.ac.at/~zeileis/

Overview

- What is a benchmark?
- A framework for comparing perfomances
  - data generating processes
  - algorithms
  - performances
- Application to supervised learning
  - Simulation
  - Competition
  - Real World
- Simulation results
- Conclusions

What is a benchmark?

What is a benchmark?

What is a benchmark?

Benchmarking has its root in land surveying:

A benchmark in this context is a mark, which was mounted on a rock, a building or a wall. It was a reference mark to define the position or the height in topographic surveying or to determine the time for dislocation. (Patterson, 1992)

What is a benchmark?

What is a benchmark?

In statistical learning:
comparison of the performance of learners or algorithms

Reference point:
data generating process

Analogy:
measure performances in a landscape of learning algorithms

But:
take variability into account

What is a benchmark?

Major goal: identify best algorithm among set of candidates.

Typical approaches:

- assess quality of algorithms by point estimates of some performance measure (e.g., MSE, misclassification),
- use bootstrap sampling and cross-validation,
- if independent test samples are available: standard statistical inference,
- else: specialized variance estimators and associated tests,
- problem: no independence between samples in k-fold cross-validation.

Framework

Conceptually different approach:

- fix data generating process DGP,

- draw independent learning samples from DGP

$$\mathcal{L} = \{z_1, \dots, z_n\},$$

- algorithm a: model fitting returns function $a(\cdot \mid \mathcal{L})$ for computing objects of interest,

- use problem specific performance measure $p(a, \mathcal{L})$.

Framework

Obtain *independent* observations from performance distribution:

- draw B independent learning samples from DGP: $\mathcal{L}^1, \dots, \mathcal{L}^B \sim DGP$,

- train K different algorithms $a_k(\cdot \mid \mathcal{L}^b) \sim A_k(DGP)$,

- apply scalar performance measure $p_{kb} = p(a_k, \mathcal{L}^b) \sim P_k = P_k(DGP)$.

$\Rightarrow$ standard statistical test procedures can be used for inference about performance.

$$H_0: \quad P_1 = \cdots = P_K$$

Framework

An algorithm a_k is better than an algorithm $a_{k'}$ iff

$$\phi(P_k) < \phi(P_{k'}).$$

Typically: $\phi(P_k) = \mathsf{E}(P_k)$.

Natural test problem: difference in location.

$$H_0 : \quad P_k = P_{k'} \quad \text{vs.} \quad H_1 : P_k(z) = P_{k'}(z - \Delta)$$

Supervised learning

- **Observations:** inputs and response $z = (y, x)$,
- **Algorithms:** predictors $a(x \mid \mathcal{L}) = \widehat{y}$,
- **Performance:** expected loss $L(y, \widehat{y})$.

Supervised learning

- **Observations:** inputs and response $z = (y, x)$,

- **Algorithms:** predictors $a(x \mid \mathcal{L}) = \widehat{y}$,

- **Performance:** expected loss $L(y, \widehat{y})$.

Example: Regression. Use quadratic loss $L(y, \widehat{y}) = (y - \widehat{y})^2$, then

$$p_{kb} \quad = \quad \mathsf{E}_{a_k} \mathsf{E}_{z=(y,x)} \left(y - a_k \left(x | \mathcal{L}^b \right) \right)^2 .$$

Not yet specified: data generating process DGP.

Supervised learning

1. Simulation:

The learning sample $\mathcal{L}$ has n independent observations $z \sim Z$. Denote by: $\mathcal{L} \sim Z_n$.

Data generating process: $DGP = Z_n$.

Associated hypothesis:

$$H_0 : \quad P_1(Z_n) = \ldots = P_K(Z_n).$$

Performance is usually evaluated by empirical performance $\widehat{P}_k$ on an independent test sample $\mathcal{T} \sim Z_m$ with m *large*.

Supervised learning

2. Competition:

Learning sample $\mathcal{L} \sim Z_n$ is provided but Z is unknown
$\Rightarrow$ use approximation $\hat{Z}$ instead.

Data generating process: $DGP = \hat{Z}_n$.

Performance is evaluated by empirical performance on a provided test sample $\mathcal{T} \sim Z_m$.

Associated hypothesis:

$$H_0: \quad \hat{P}_1(\hat{Z}_n) = \ldots = \hat{P}_K(\hat{Z}_n).$$

Supervised learning

3. Real World:

A learning sample $\mathcal{L} \sim Z_n$ is available but no test sample $\mathcal{T}$.

Data generating process: $DGP = \hat{Z}_n$.

Problem: How should performance be computed? Some test sample needs to be "generated".

Supervised learning

Evaluate performance by:

- sample splitting $\rightarrow$ Situation 2.
- use learning sample $\mathcal{T} = \mathcal{L}$
- out-of-bag: for each bootstrap sample $\mathcal{L}^b$ use the observations $\mathcal{L} \setminus \mathcal{L}^b$
- cross-validation: e.g., average performance on folds

Associated hypothesis:

$$H_0: \quad \widehat{P}_1(\hat{Z}_n) = \ldots = \widehat{P}_K(\hat{Z}_n).$$

Simulation results

Data generating process DGP:
Z is a simple regression model

$$y \quad = \quad 2x + \beta x^2 + \varepsilon,$$

where

- $X \sim U(0,5)$,
- $\varepsilon \sim \mathcal{N}(0,1)$,
- $n = 50$.

Loss: $L(y, \widehat{y}) = (y - \widehat{y})^2$.

Simulation results

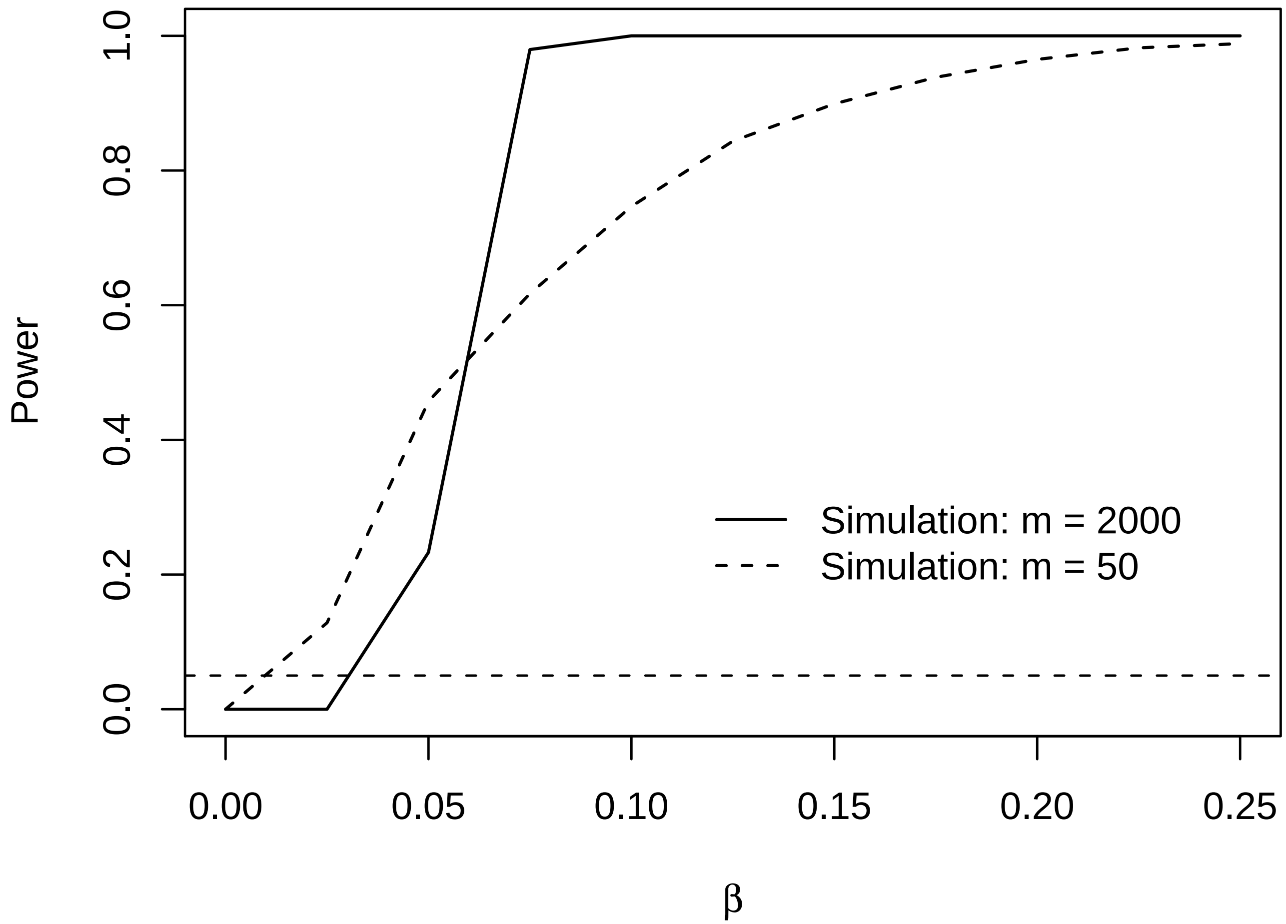
Algorithms: two nested linear models

- a_1: linear regression with input x,
- a_2: quadratic regression with inputs x and x^2.

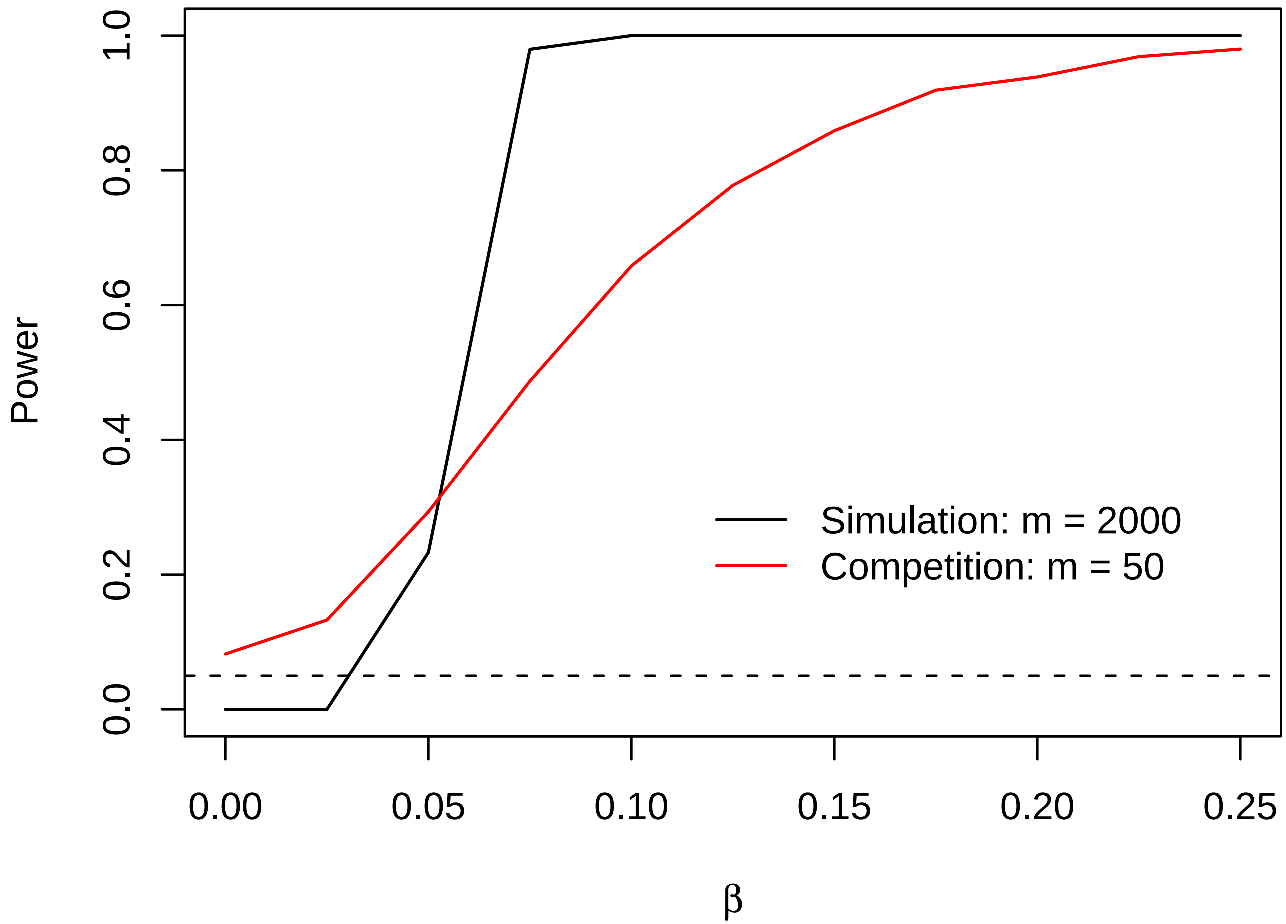
Note: a_1 only unbiased for $\beta = 0$, but with smaller variance.

Test: one-sided test for difference in expected performance based on $B = 250$ learning samples. Estimate power by 5000 Monte Carlo replications.

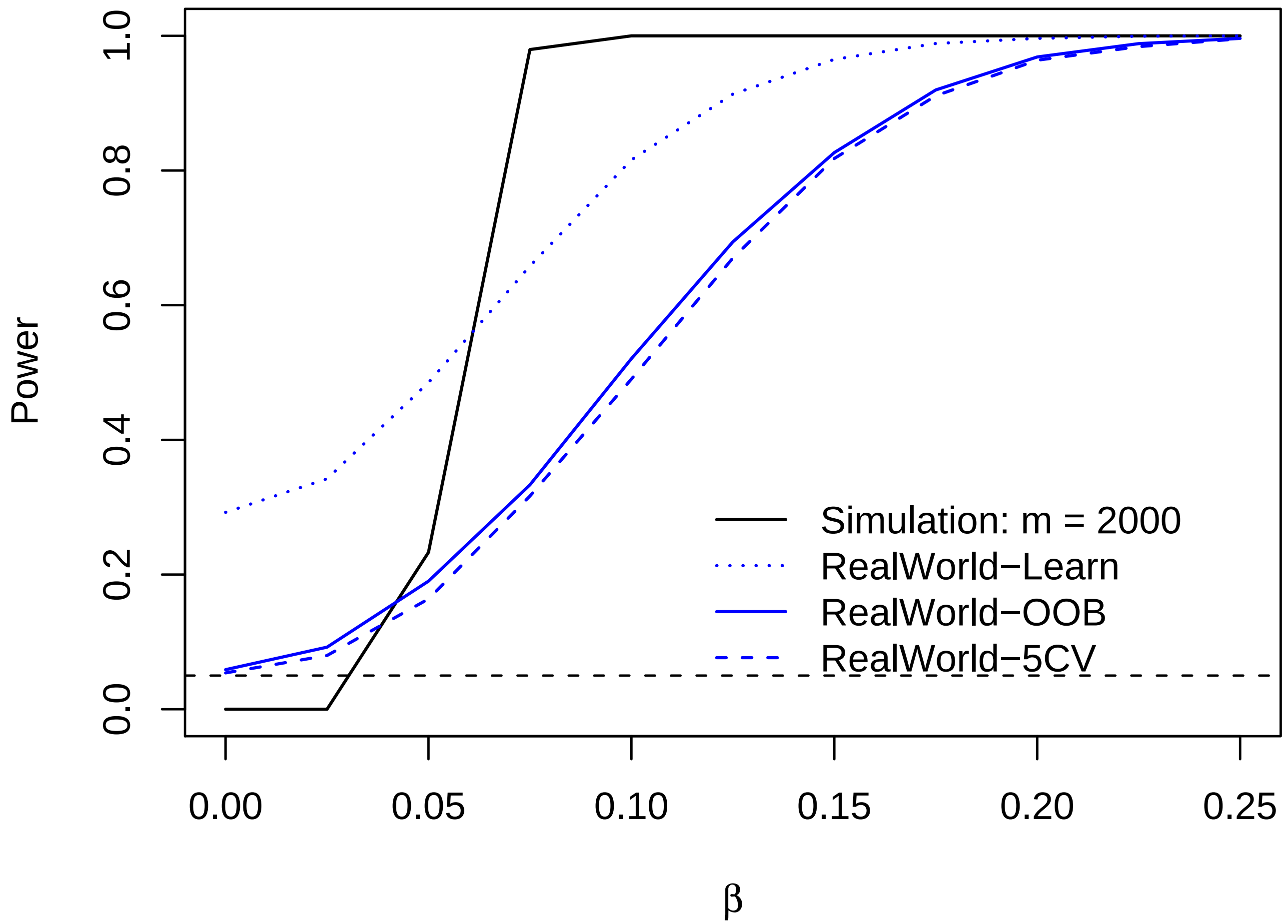
Simulation results



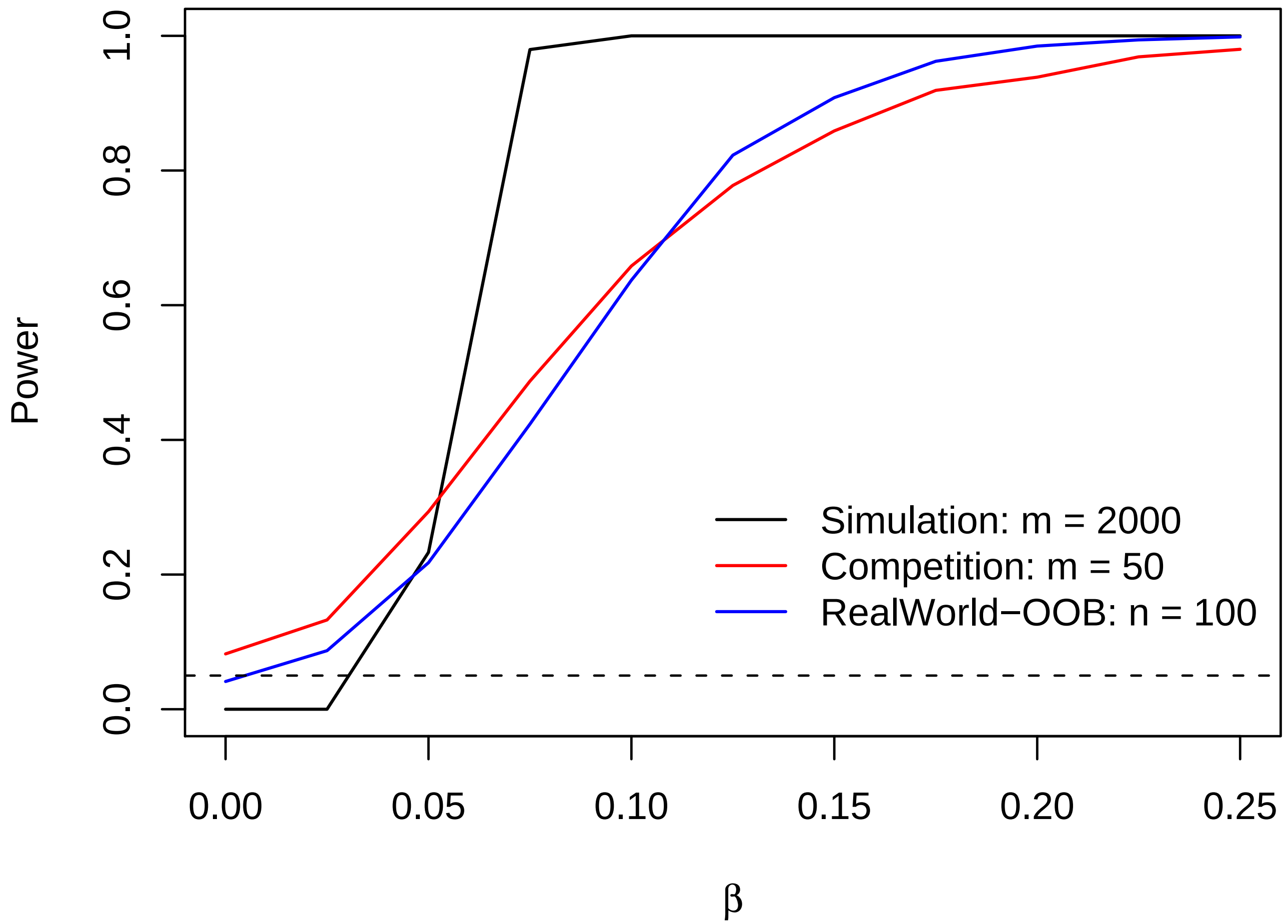
Simulation results



Simulation results



Simulation results



Simulation results

Results indicate:

- using a single test sample favours over-fitting and reduces power,

- cross-validation works well,
  but is computationally expensive

- out-of-bag approach seems to work equally well,
  but is computationally cheaper.

Conclusions

- *unified conceptual framework* for benchmark experiments,
- can be easily adapted to various situations,
- *do it yourself*:
  Just figure out what are the data-generating process, algorithms and performance measures,
- results of the experiment do not require specialized methods for the analysis: the full *standard statistical tool box* can be applied directly.