Results of a benchmark study with focus on SVMs and resample/combine methods

David Meyer

January 23, 2004

SV C-Classification

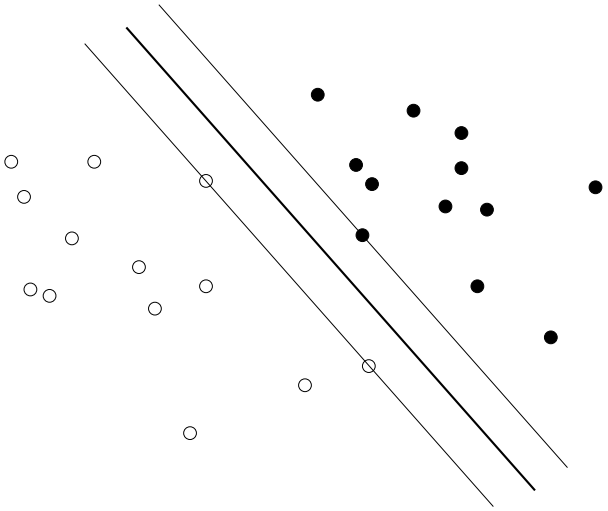
We are given nonoverlapping 2-class training data:

$$\{(\mathbf{x}_i, y_i)\}_1^l, \mathbf{x}_i \in \mathbb{R}^n$$

$$y_i \in \{-1, +1\}$$

Goal: find separating hyperplane.

SV C-Classification



SV C-Classification

Formalization:

Constrained quadratic optimization problem, usually solved in its *dual* representation:

$$\min_{\alpha} \quad \frac{1}{2}\sum_{i,\,j}^{l} y_i y_j \alpha_i \alpha_j \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j)\rangle - \sum_{i}^{l} \alpha_i$$

$$\text{s.t.} \quad \begin{array}{l} \sum y_i \alpha_i = 0 \\ 0 \le \alpha_i \le C \\ i = 1, \ldots, l \end{array}$$

The inner product $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j)\rangle$ is called a *Kernel*.

SV C-Classification

Decision Function:

$$f(x) = \text{sgn}\left(\sum_i^l y_i \alpha_i K(x_i, x) - b\right)$$

This is a weighted superposition of kernel functions.

Note:

The complexity of the 'machine' depends basically on the number of support vectors.

Kernels

For certain functions ϕ, the inner product can be computed by $K(\mathbf{x}_i, \mathbf{x}_j)$ without explicitely knowing $\phi(\cdot)$.

Some popular kernels:

Polynomial	$\left(\gamma\langle \mathbf{x}_i, \mathbf{x}_j\rangle + \beta\right)^d$
Radial Basis Function	$\exp\left(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2\right)$
Sigmoid	$\tanh\left(\gamma\langle \mathbf{x}_i, \mathbf{x}_j\rangle + \beta\right)$

We do have to specify (in addition to C) the hyperparameters d, γ, and β: this usually requires *tuning*.

Chih-Chen Lin's `libsvm`

- Starting point: svm*light* (Joachims, 1998) and Pseudo code for SMO (Platt, 1998)
- Motivation: bad performance of svm*light* on complex cases ⇒ development of `bsvm`
- Drawbacks of `bsvm`: very complex, uses third-party optimization solvers ⇒ simple, usable code: `libsvm`
- Meanwhile: several extensions and optimizations ⇒ very competitive
- Future development: increase usability ("black box")

Chih-Chen Lin's `libsvm`

- All standard formulations (classification, regression, novelty detection)
- `libsvm` won several competitions, e.g.:
  - EUNITE 2001 (competition on electricity load prediction)
  - IJCNN Challenge 2001 (two of three competitions)
- very good (fast) quadratic solver (decomposition method), high standard in numerical methods.

The R interface

- `libsvm` is a C++ library with standard C interface
- R interface in package `e1071` on CRAN (`http://cran.R-project.org/`)
- main functions:
  - training (returning an '`svm`' object)
  - predicting
  - tuning (for hyperparameter selection)

libsvm under test

- No extensive benchmark study available comparing SVMs to many methods on many data sets
- Often only compared to other machine learning techniques (e.g., neural networks)
- How do SVMs compare to 'traditional' methods (e.g., LDA, GLMs, ...)?
- R offers a wide variety of methods, all accessible with a similar interface $\Rightarrow$ easy to use in a benchmark study

Methods

16 classification methods and 9 regression methods:

Machine Learning Techniques: Neural Networks, Trees, Learning Vector Quantization, Nearest Neighbour, FDA, MDA

Resample & Combine Methods: Bagged Trees, Random Forests, MART

Spline Models: MARS, BRUTO, Projection Pursuit

'Traditional' Methods: LDA, QDA, (Generalized) Lin. Models

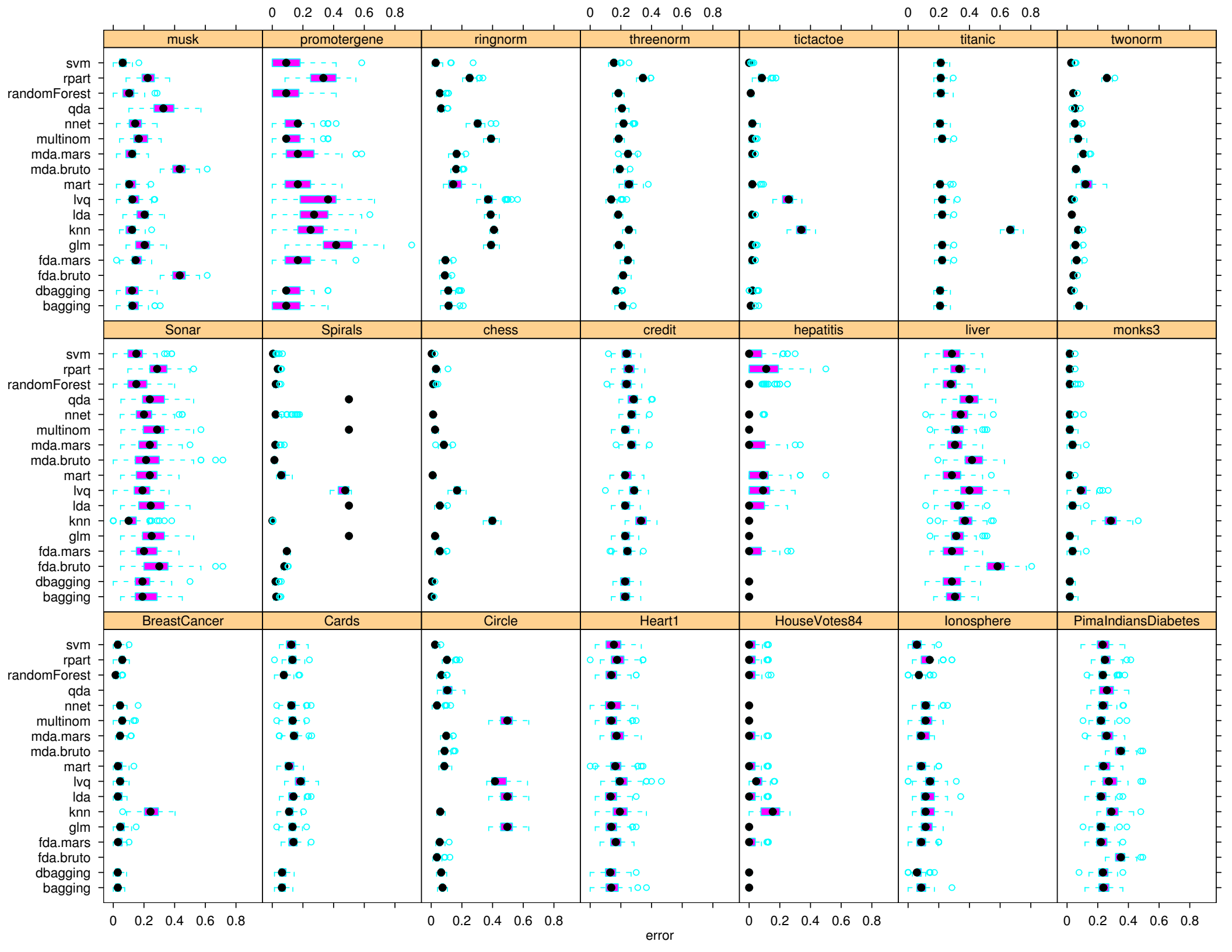
Data sets

Most data sets were taken from the UCI machine learning data base:

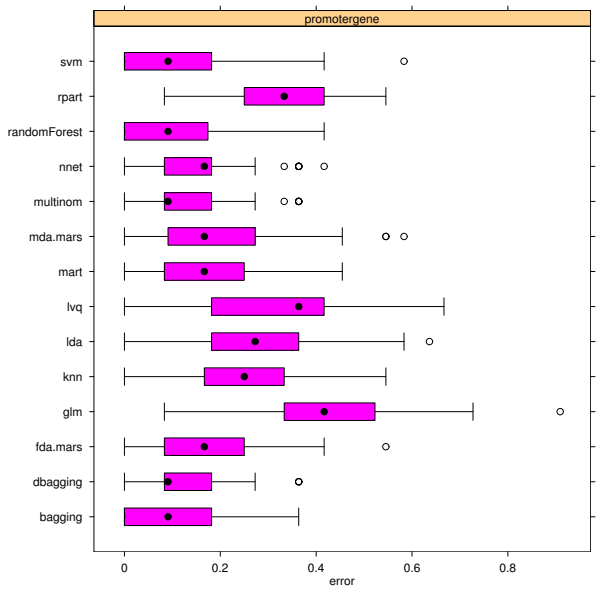
- 21 data sets for (binary) classification
- 12 for regression
- artifical and real data
- small (155) to large (8192) data sets
- Mix of binary, categorical, and metric input variables

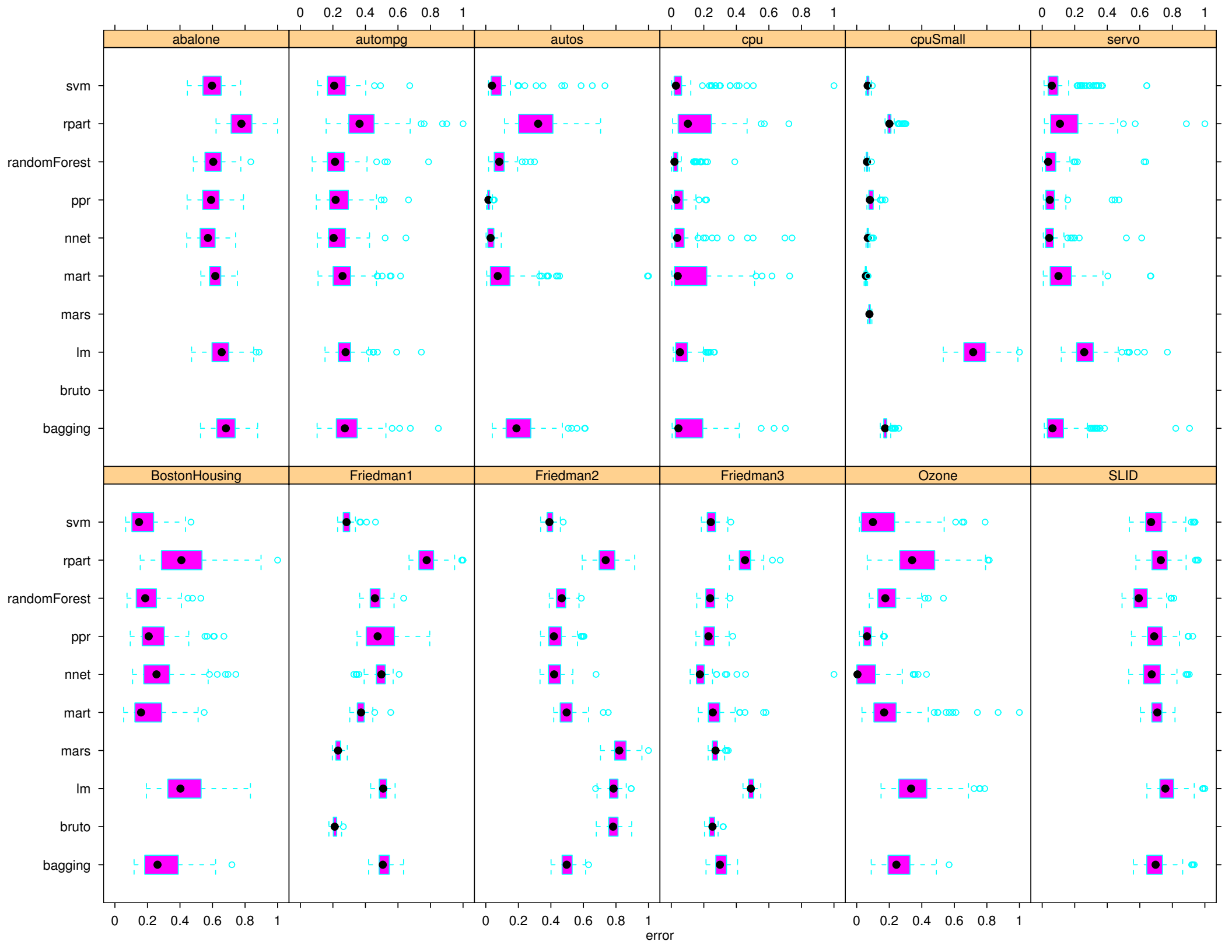
Simulation Setup

- Generation of 100 training and 100 test sets
  (for real data, using 10 times 10-fold cross validation)
- Scaling of data
- Tuning on 2/3 of training set, other 1/3 used for validation
- Training on complete training set
- Performance measure computed on test set
  (Mean Squared Error and Classification Error)

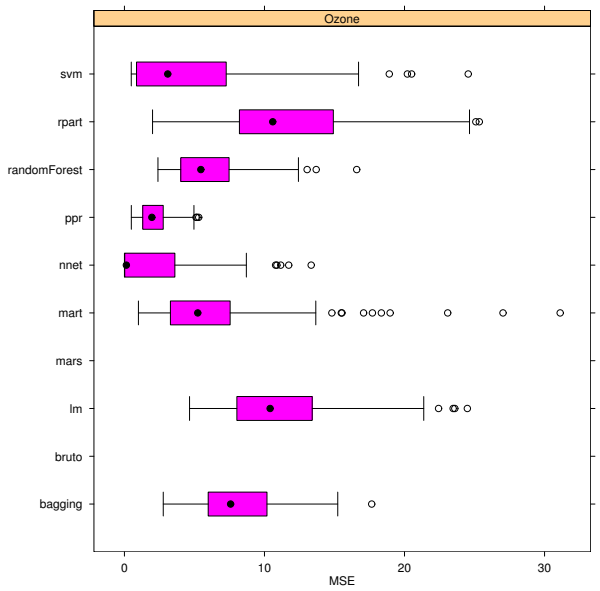

musk
promotergene
ringnorm
threenorm
tictactoe
titanic
twonorm
Sonar
Spirals
chess
credit
hepatitis
liver
monks3
BreastCancer
Cards
Circle
Heart1
HouseVotes84
Ionosphere
PimaIndiansDiabetes
svm
rpart
randomForest
qda
nnet
multinom
mda.mars
mda.bruto
mart
lvq
lda
knn
glm
fda.mars
fda.bruto
dbagging
bagging
0
0.2
0.4
0.6
0.8
error

Results


promotergene
svm
rpart
randomForest
nnet
multinom
mda.mars
mart
lvq
lda
knn
glm
fda.mars
dbagging
bagging
0
0.2
0.4
0.6
0.8
error


abalone
autompg
autos
cpu
cpuSmall
servo
BostonHousing
Friedman1
Friedman2
Friedman3
Ozone
SLID
svm
rpart
randomForest
ppr
nnet
mart
mars
lm
bruto
bagging
0
0.2
0.4
0.6
0.8
1
error

Results



Results

Classification:

- Generally good performance of SVMs (almost always ranked in top 3)
- However, SVMs outperformed on 10 out of 21 data sets
- In addition, SVMs share 5 top ranks with other methods
- Average rankings for dispersion
- Good performance of ensemble methods (bagging, random forests, MART)

Results

Regression:

- Again, good results for SVMs (ranked among top 3)
- However, only ranked first on two data sets
- Average rankings for dispersion
- Good performances by neural networks and random forests

Implementation

- Existing stuff:
  - (most) data in package `mlbench`
  - R code for flexible design of benchmarking studies
- Open issues:
  - Better parameterization in configuration files
  - Flexible data sources
  - File structures (of results)
  - Summaries & Visualization
  - Parallel computing

Resources

- CRAN:
  http://cran.R-project.org/

- `libsvm`:
  http://www.csie.ntu.edu.tw/~cjlin/libsvm/

- Benchmarks:
  http://www.wu-wien.ac.at/am/Download/report78.pdf
  (*Neurocomputing*, September 2003)

Shameless Plug

useR! 2004

The 1st R user conference
Vienna, 20.–22. May 2004

http://www.ci.tuwien.ac.at/Conferences/useR-2004/