# COBRA : agrégation non-linéaire de prédicteurs

Benjamin Guedj

LSTA, UPMC & LTCI, Telecom ParisTech

http://www.lsta.upmc.fr/doct/guedj

En collaboration avec Gérard Biau, Aurélie Fischer et James D. Malley

# Outline

# Aggregating predictors

- The present talk focuses on the prediction problem in a regression setting.

- Countless methods address this topic: Least squares regression, Lasso, ridge regression, Elastic net, nearest neighbors, (PAC-)bayesian methods, random trees and forests, ...

# Aggregating predictors

- The present talk focuses on the prediction problem in a regression setting.

- Countless methods address this topic: Least squares regression, Lasso, ridge regression, Elastic net, nearest neighbors, (PAC-)bayesian methods, random trees and forests, ...

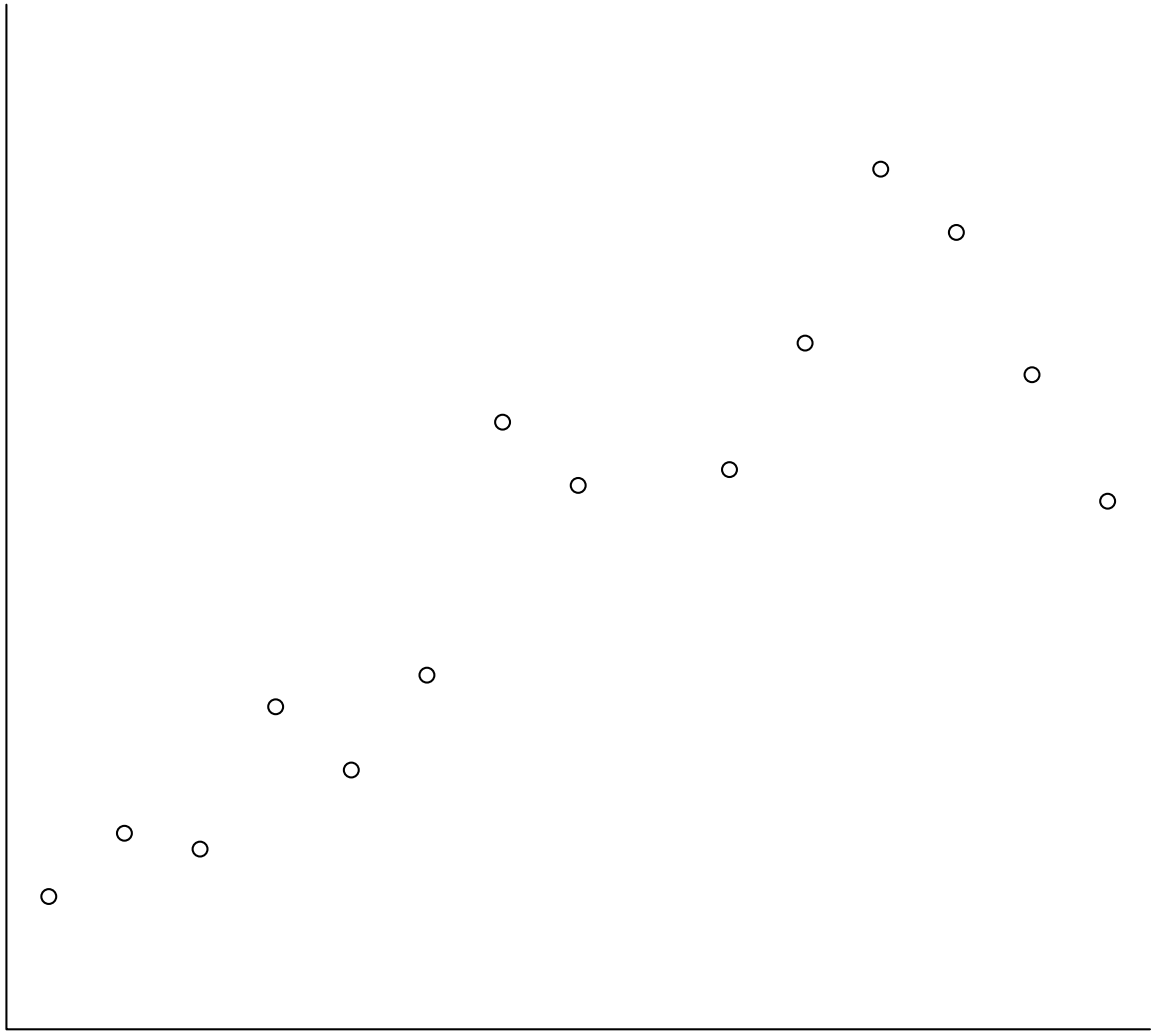
- For a given dataset, how should we decide which method to use?

# Aggregating predictors

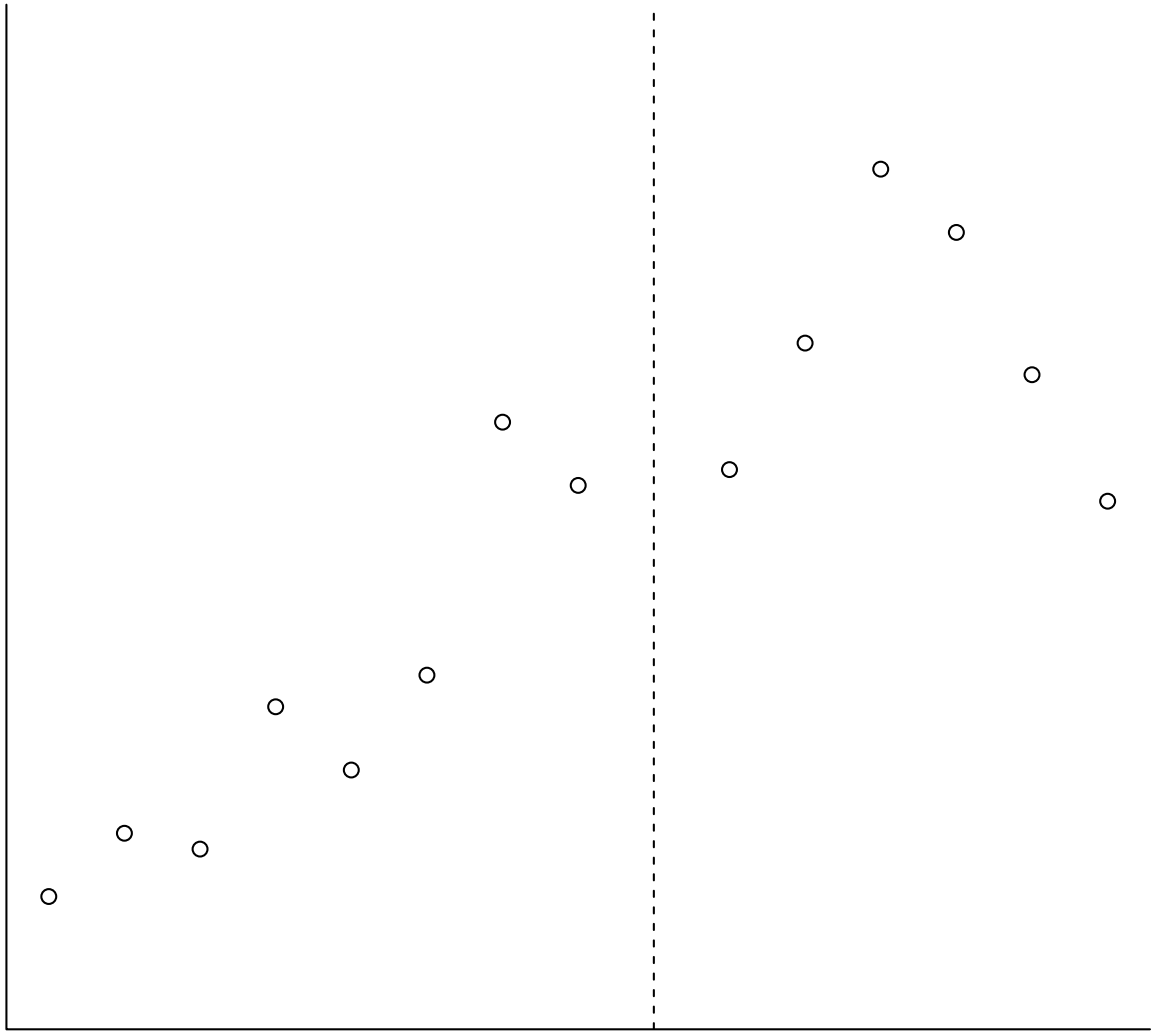
- The present talk focuses on the prediction problem in a regression setting.
- Countless methods address this topic: Least squares regression, Lasso, ridge regression, Elastic net, nearest neighbors, (PAC-)bayesian methods, random trees and forests, ...
- For a given dataset, how should we decide which method to use?
- A nice strategy is to aggregate them: Model selection, linear and convex aggregation, sparse aggregation.

# Outline

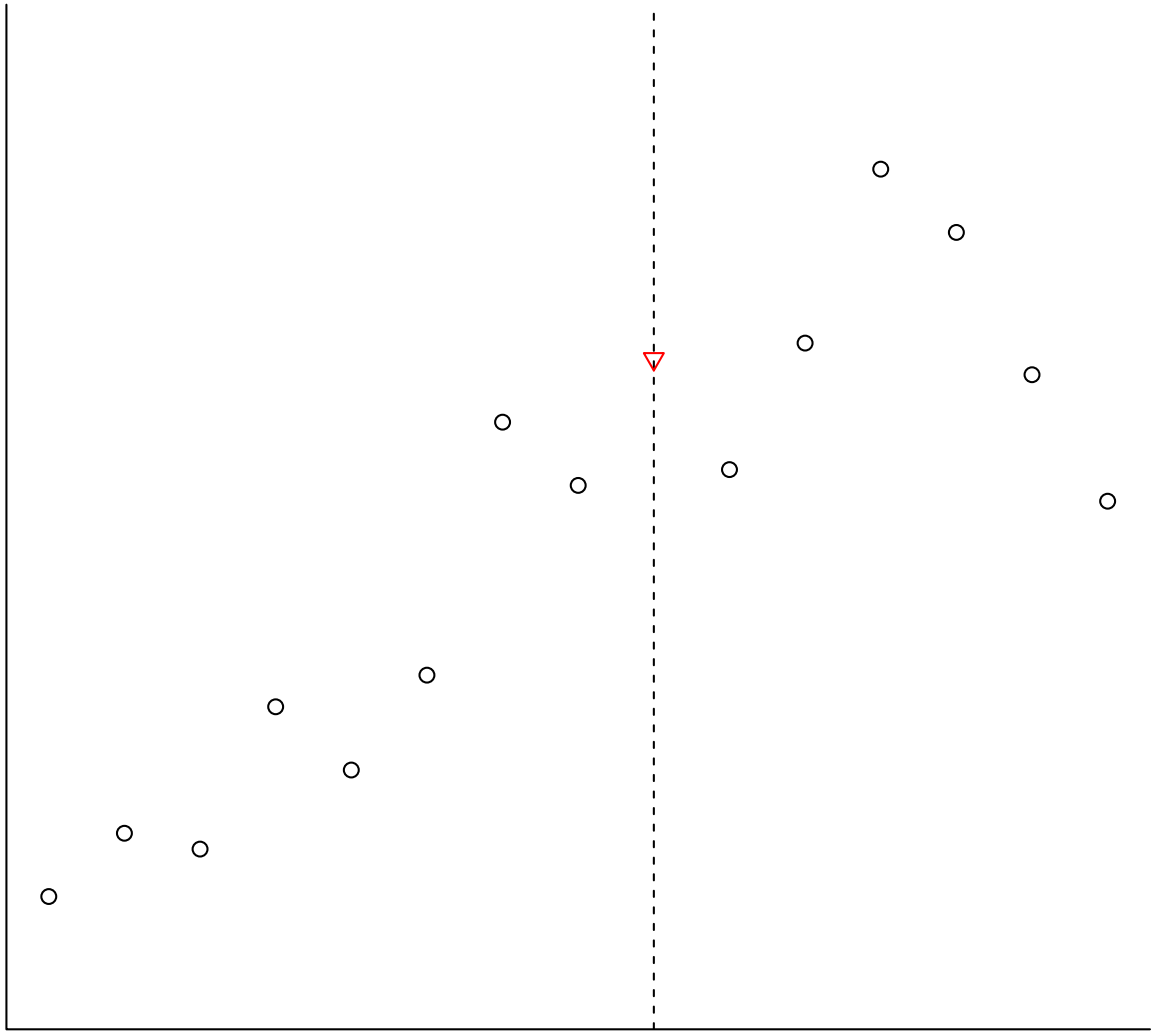
# Example



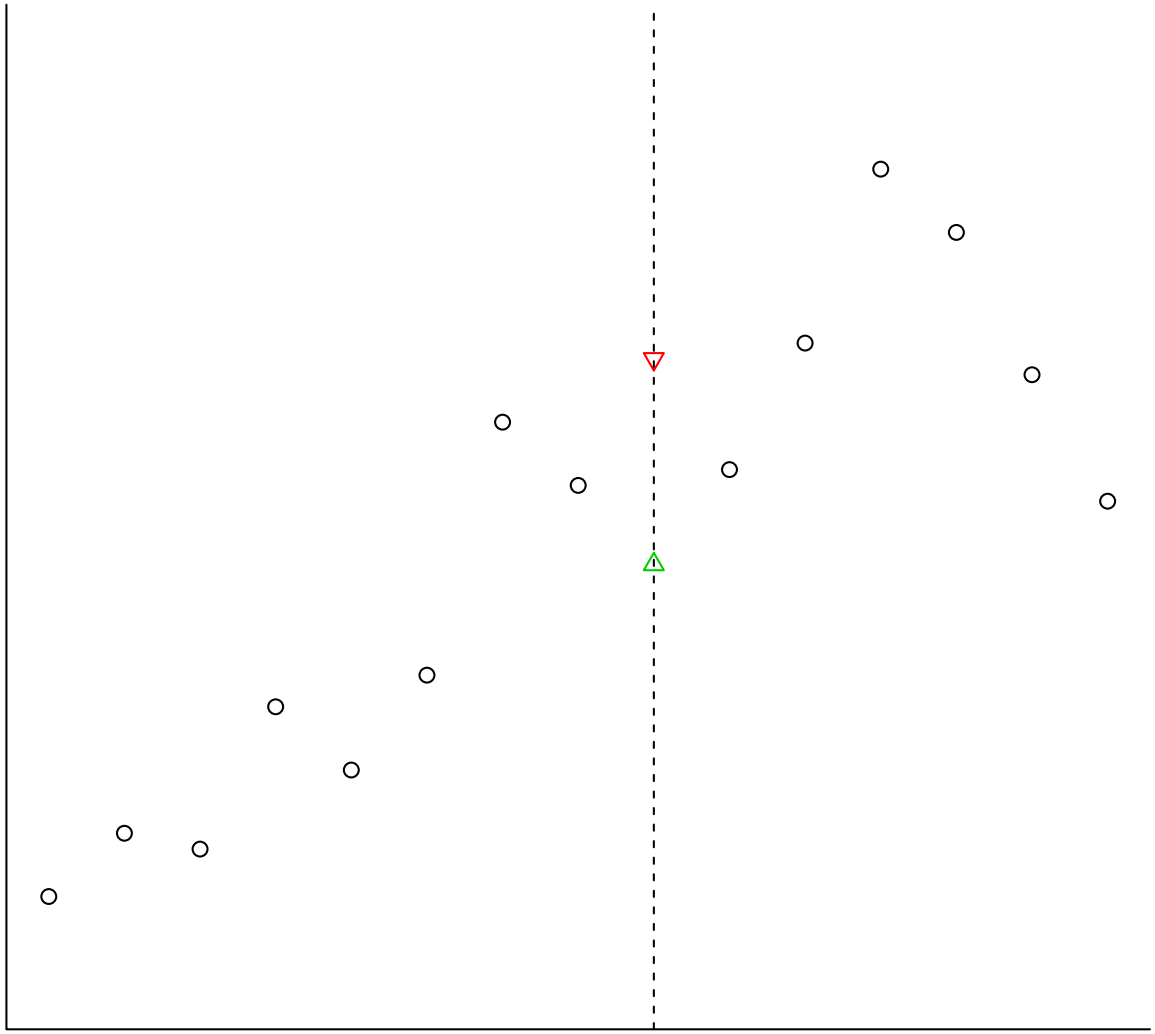
# Example



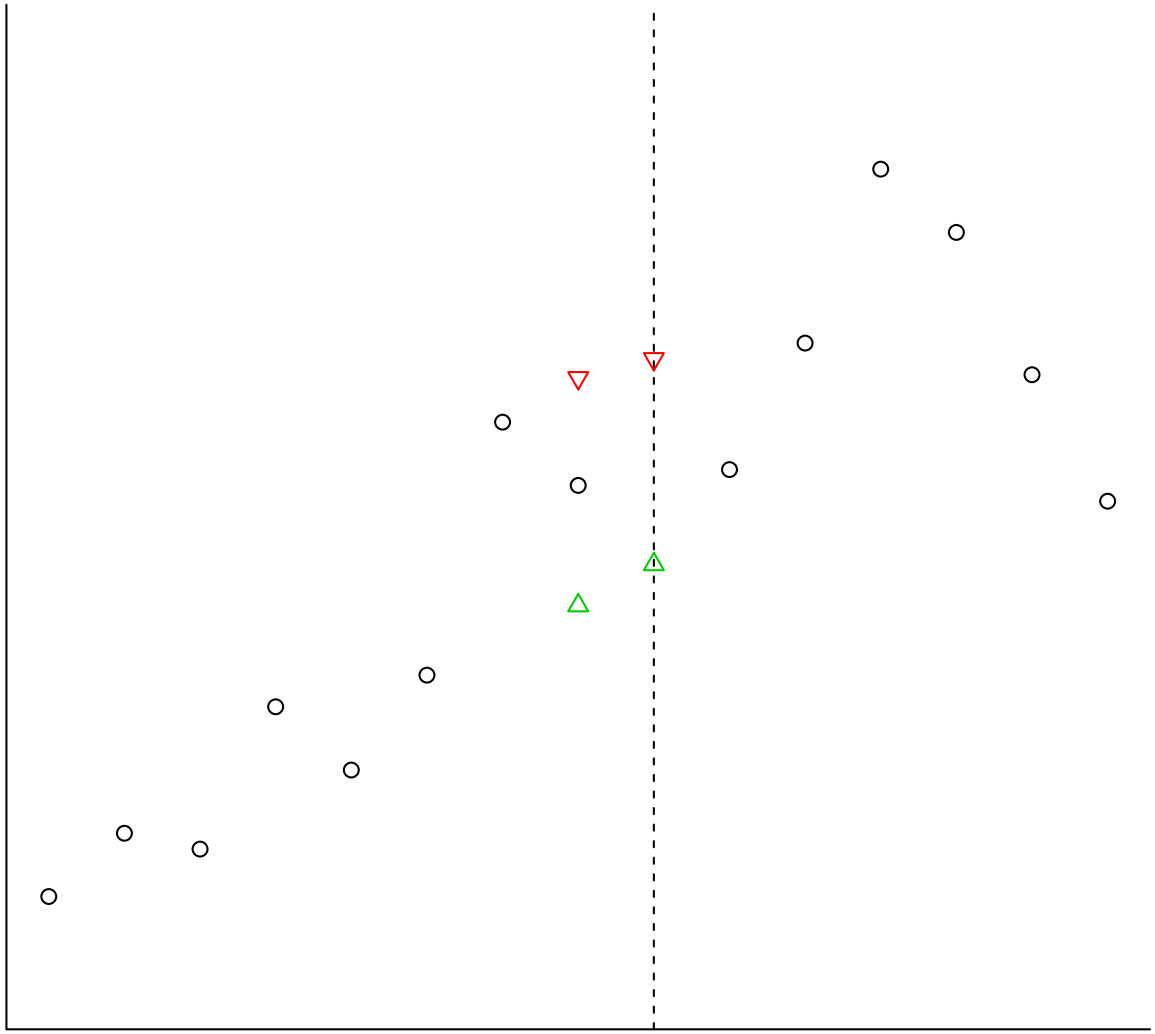
# Example



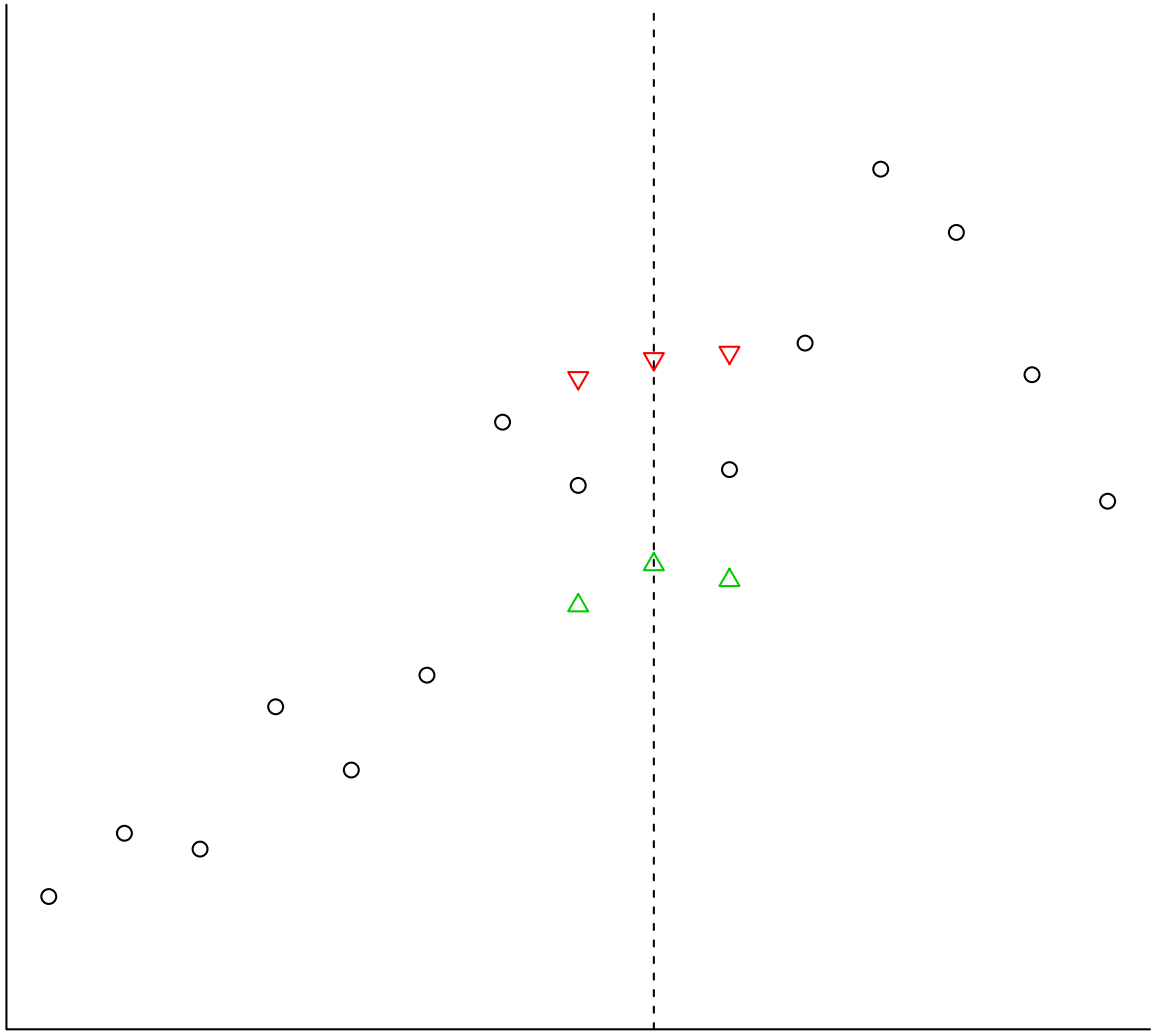
# Example



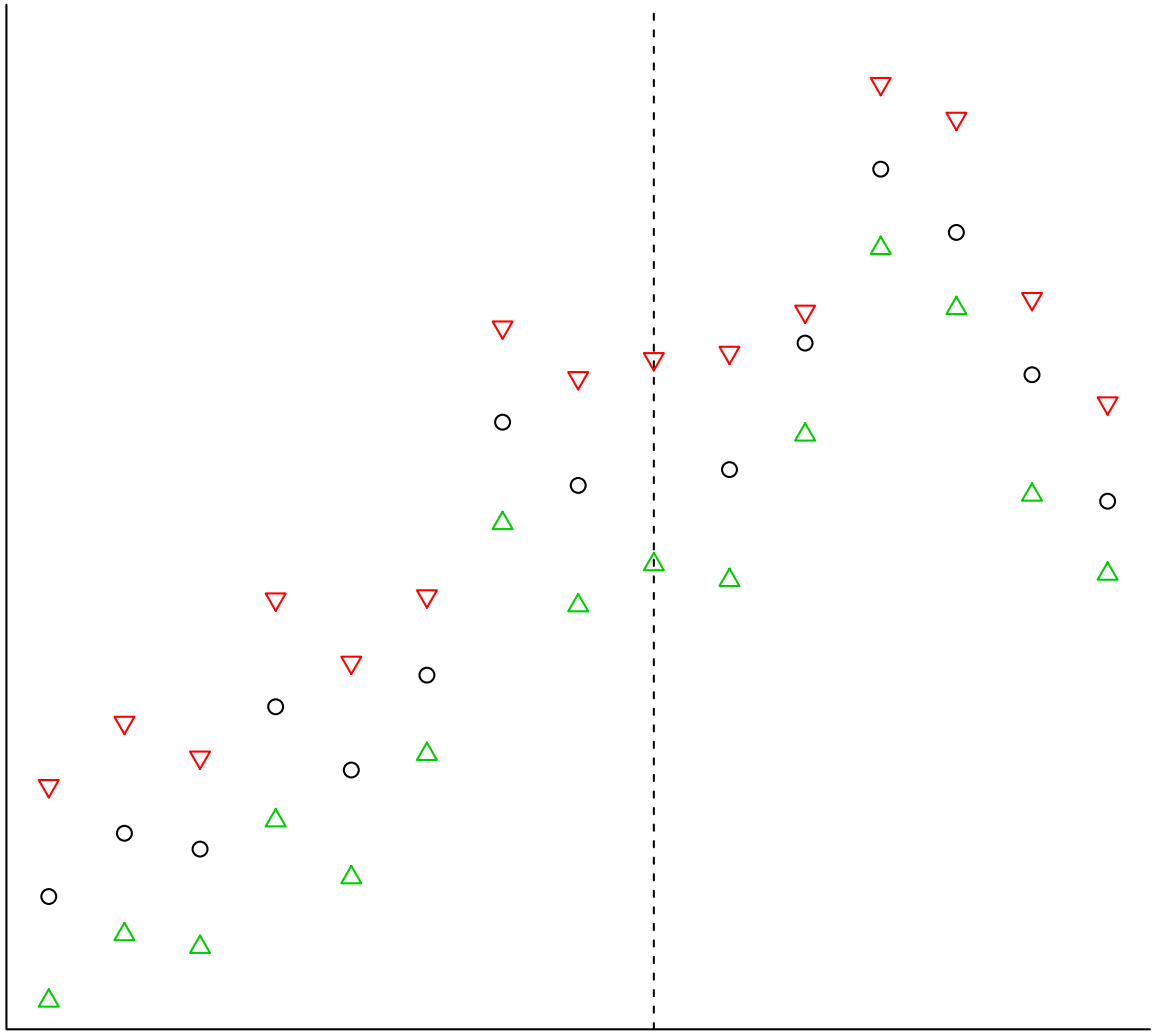
# Example



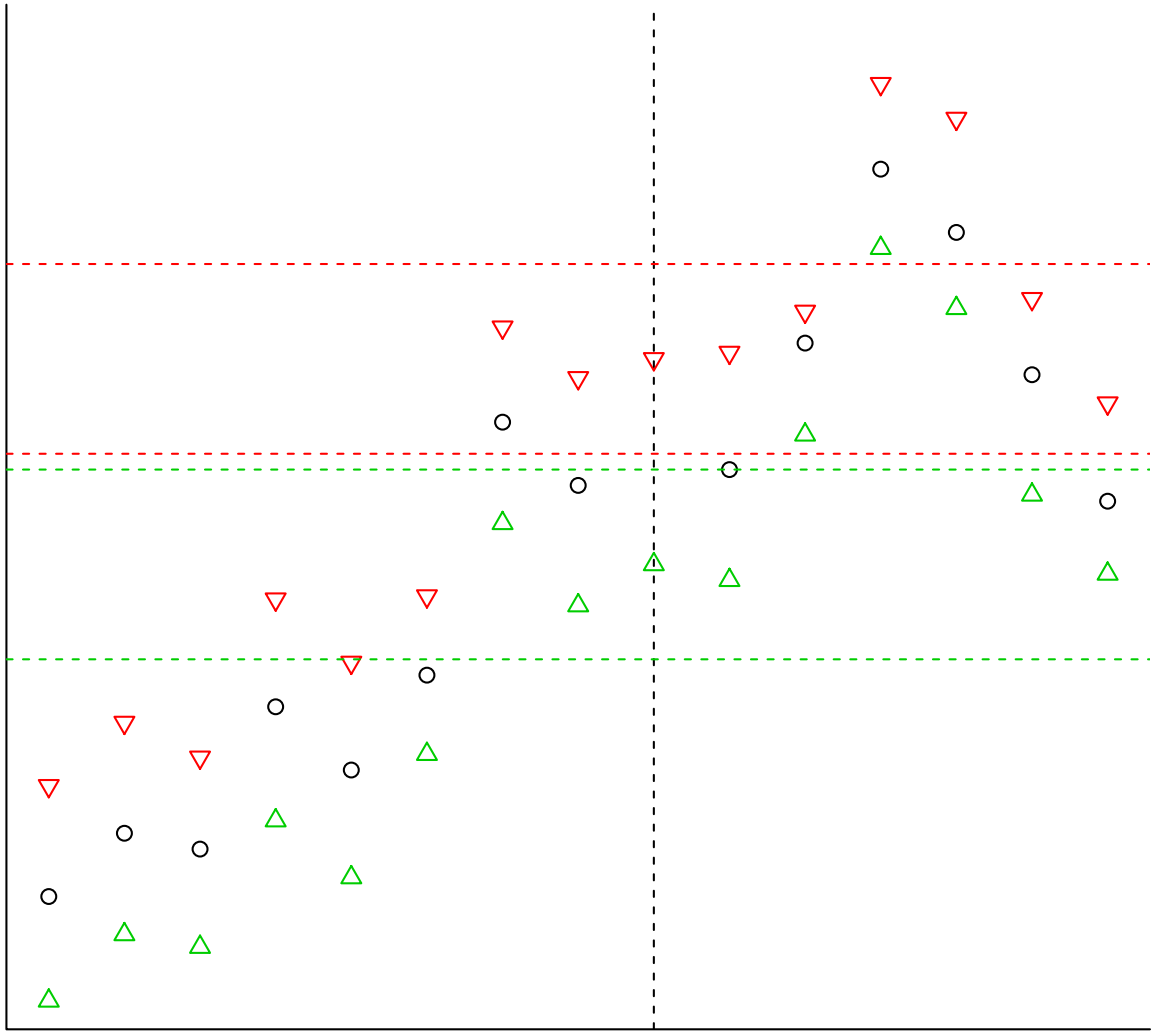
# Example



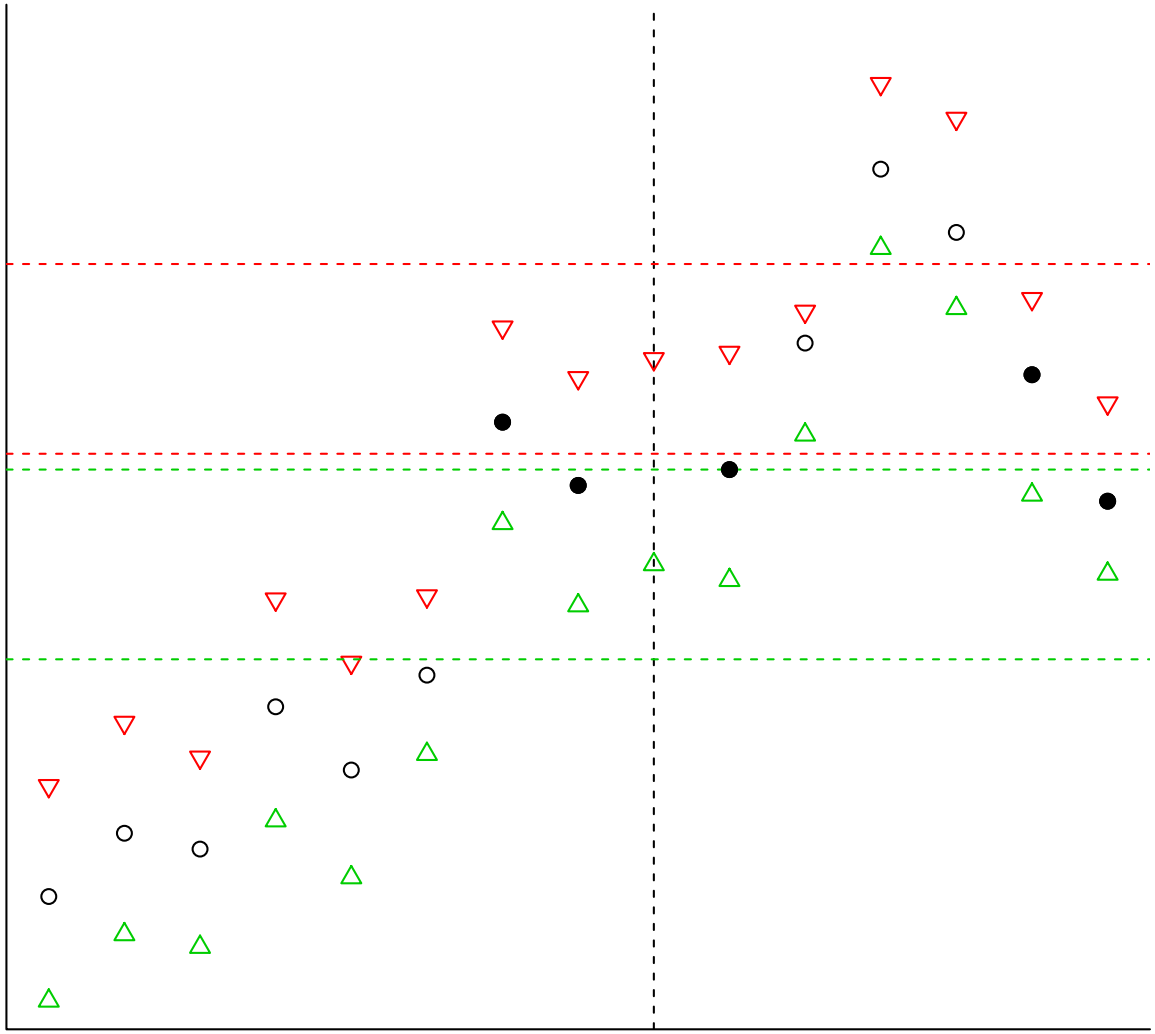
# Example



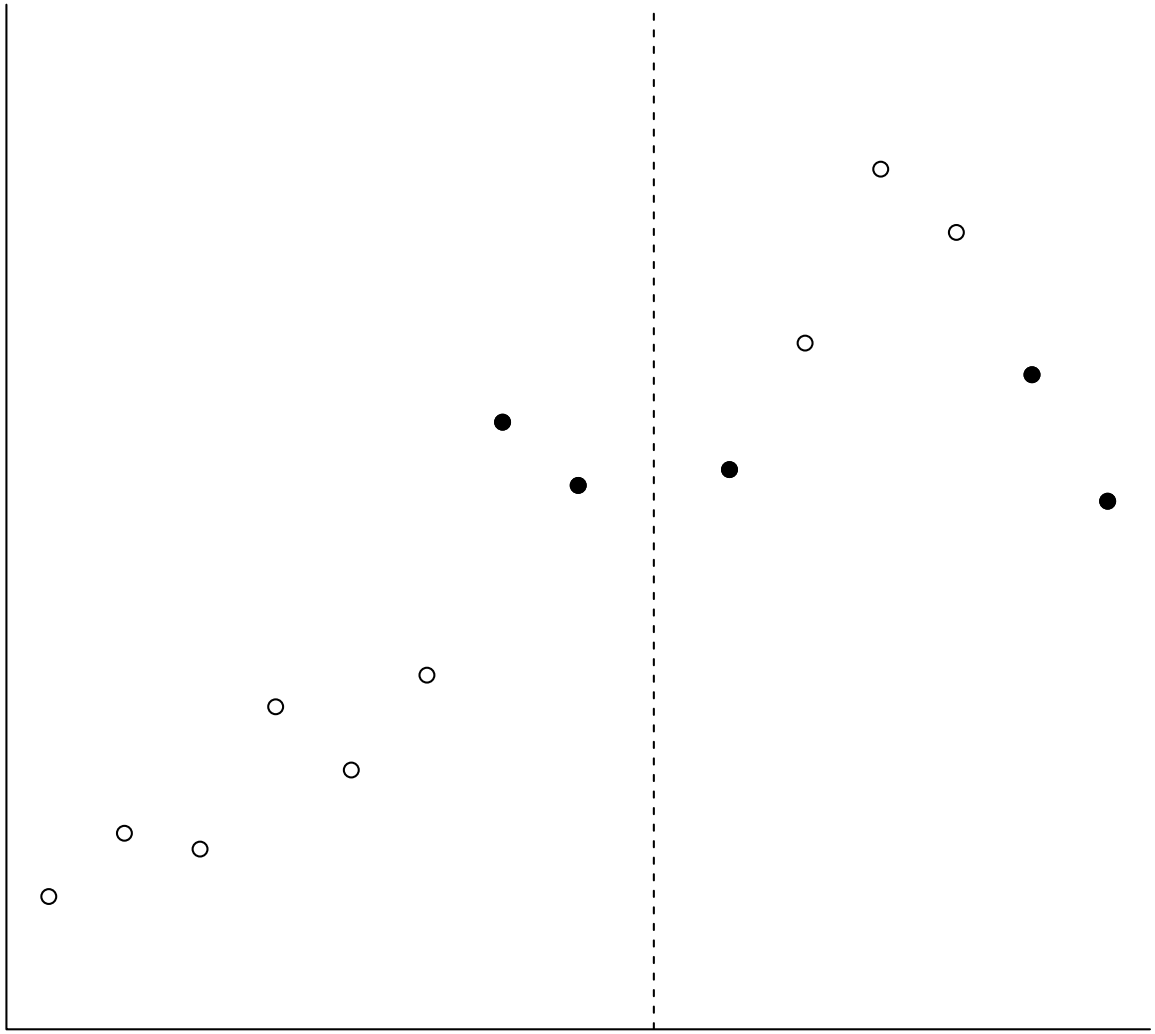
# Example



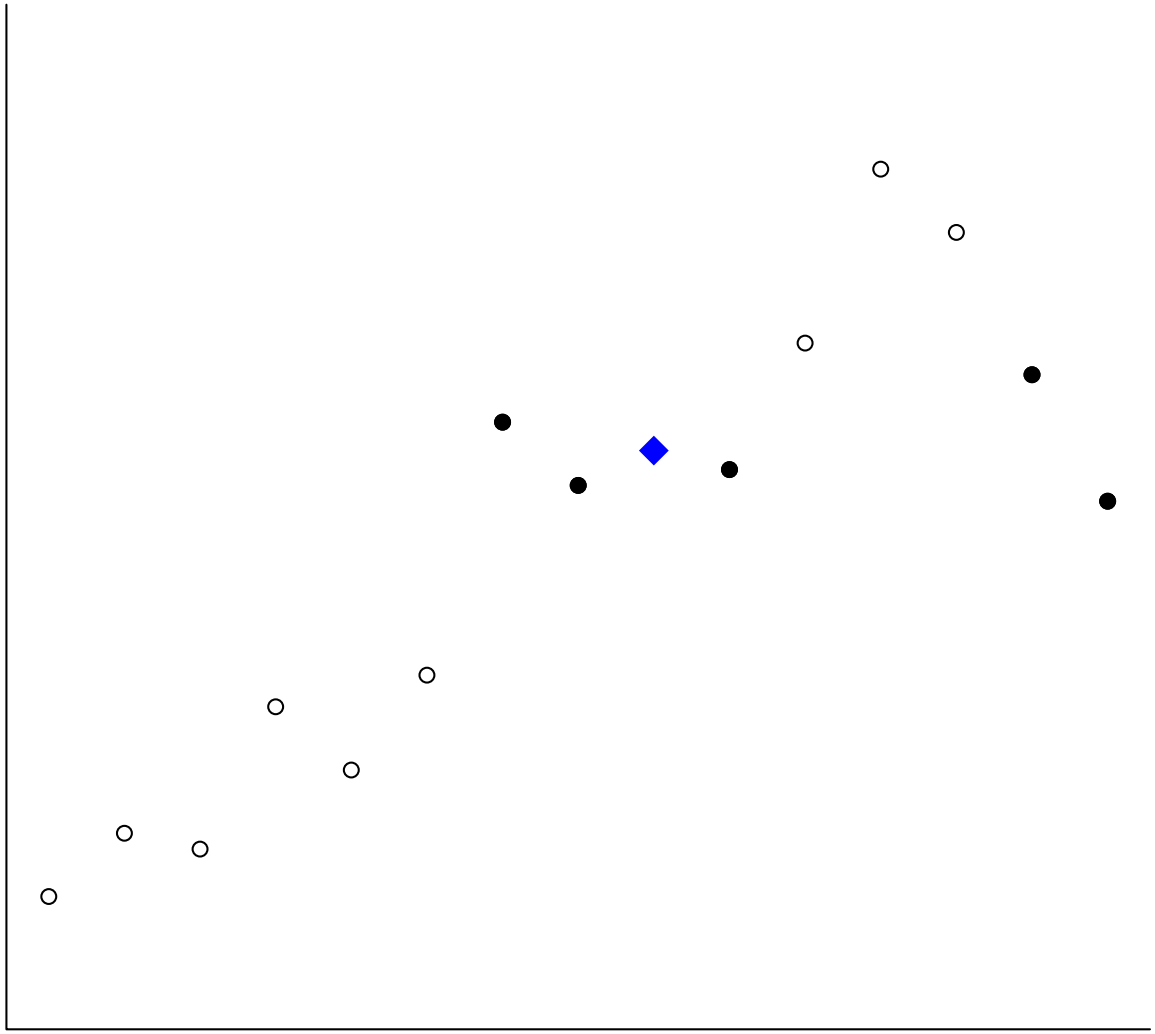
# Example



# Example



# Example



# Notation

- Training sample: $\mathcal{D}_n = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\}$.

- Goal: Estimate the regression function $r^\star : \mathbf{x} \mapsto \mathbb{E}[Y|\mathbf{X} = \mathbf{x}]$.

# Notation

- Training sample: $\mathcal{D}_n = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\}$.
- Goal: Estimate the regression function $r^\star : \mathbf{x} \mapsto \mathbb{E}[Y|\mathbf{X} = \mathbf{x}]$.
- Splitting:

$$\mathcal{D}_n = \mathcal{D}_k \cup \mathcal{D}_\ell.$$

# Notation

- Training sample: $\mathcal{D}_n = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\}$.
- Goal: Estimate the regression function $r^\star : \mathbf{x} \mapsto \mathbb{E}[Y|\mathbf{X} = \mathbf{x}]$.
- Splitting:
$$\mathcal{D}_n = \mathcal{D}_k \cup \mathcal{D}_\ell.$$
- Basic {$\emptyset$,semi,non}parametric machines: $\mathbf{r}_k = (r_{k,1}, \ldots, r_{k,M})$.
- Sole requirement: They must deliver an estimation of $r^\star$ on the basis of $\mathcal{D}_k$ only.

# The regression collective

- Define

$$T_n\left(\mathbf{r}_k(\mathbf{x})\right) = \sum_{i=1}^{\ell} W_{n,i}(\mathbf{x}) Y_i,$$

where

$$W_{n,i}(\mathbf{x}) = \frac{\mathbf{1}_{\bigcap_{m=1}^{M}\{|r_{k,m}(\mathbf{x}) - r_{k,m}(\mathbf{X}_i)| \leq \varepsilon_\ell\}}}{\sum_{j=1}^{\ell} \mathbf{1}_{\bigcap_{m=1}^{M}\{|r_{k,m}(\mathbf{x}) - r_{k,m}(\mathbf{X}_j)| \leq \varepsilon_\ell\}}}.$$

- Key idea: Closeness is measured by the basic machines, used as a distance indicator over the data.

# Theoretical performance

**Theorem (Biau, Fischer, G. and Malley, 2013)**

*Under a technical regularity assumption over the machines $\mathbf{r}_k$, with the choice $\varepsilon_\ell \propto \ell^{-\frac{1}{M+2}}$, one has*

$$\mathbb{E}\left|T_n\left(\mathbf{r}_k(\mathbf{X})\right) - r^\star(\mathbf{X})\right|^2 \leq \min_{m=1,\ldots,M} \mathbb{E}\left|r_{k,m}(\mathbf{X}) - r^\star(\mathbf{X})\right|^2 + C\ell^{-\frac{2}{M+2}},$$

*for some positive constant $C$ independent of $k$.*

# Theoretical performance

**Theorem (Biau, Fischer, G. and Malley, 2013)**

*Under a technical regularity assumption over the machines $\mathbf{r}_k$, with the choice $\varepsilon_\ell \propto \ell^{-\frac{1}{M+2}}$, one has*

$$\mathbb{E}\left|T_n\left(\mathbf{r}_k(\mathbf{X})\right) - r^\star(\mathbf{X})\right|^2 \leq \min_{m=1,\ldots,M} \mathbb{E}\left|r_{k,m}(\mathbf{X}) - r^\star(\mathbf{X})\right|^2 + C\ell^{-\frac{2}{M+2}},$$

*for some positive constant $C$ independent of $k$.*

Corollary: If at least one machine is consistent, the regression collective inherits this property.

# Outline

# The R package COBRA

- COBRA: COmBined Regression Alternative.
  Freely available on the CRAN website.

# The R package COBRA

- COBRA: COmBined Regression Alternative.
  Freely available on the CRAN website.

- Natively takes advantage of multi-core CPUs.

- Input: training sample + (machines) + testing sample.

# The R package COBRA

- COBRA: COmBined Regression Alternative.
  Freely available on the CRAN website.
- Natively takes advantage of multi-core CPUs.
- Input: training sample + (machines) + testing sample.
- Default machines: `lars`, `ridge`, `FNN`, `tree` and `randomForest`.
- Better: The user can feed COBRA with her/his own preferred machines.

# Extension

- All primal machines are asked to have to same opinion. This can be ruinous if the pool of machines is heterogeneous.

# Extension

- All primal machines are asked to have to same opinion. This can be ruinous if the pool of machines is heterogeneous.

- More sophisticated form:

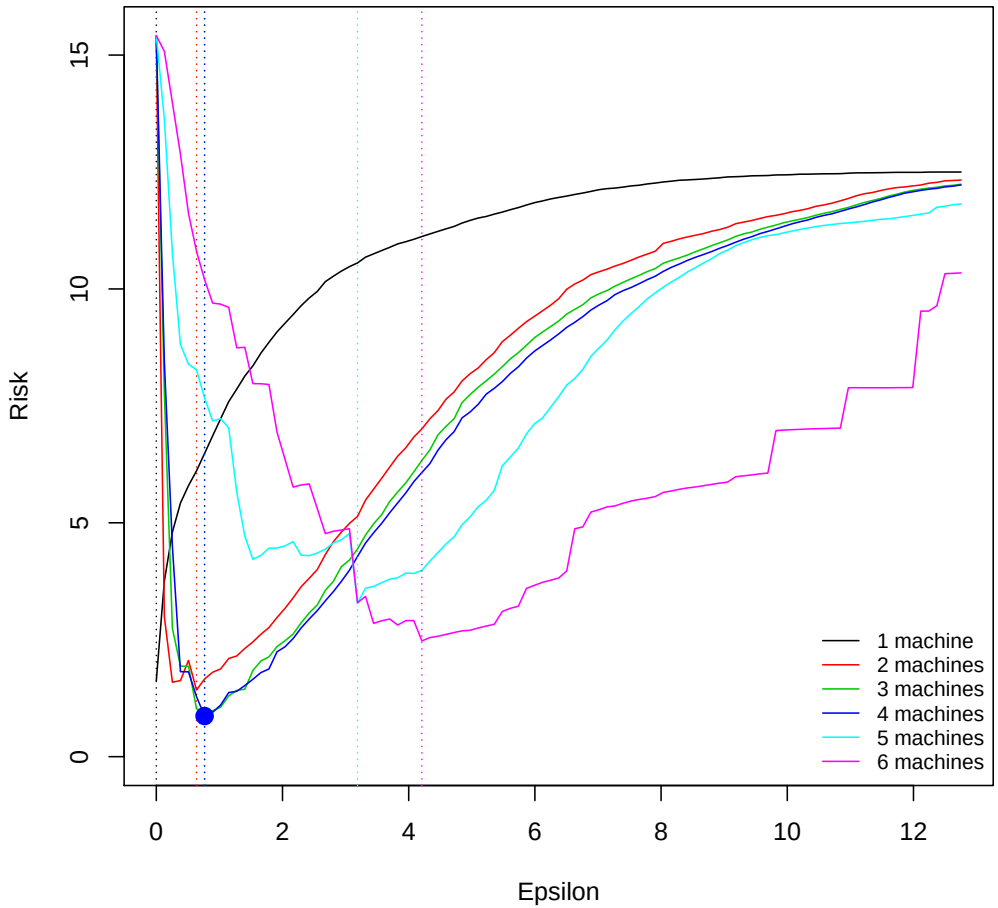
$$W_{n,i}(\mathbf{x}) = \frac{\mathbf{1}_{\{\sum_{m=1}^{M} \mathbf{1}_{\{|r_{k,m}(\mathbf{x}) - r_{k,m}(\mathbf{x}_i)| \leq \varepsilon_\ell\}} \geq M\alpha\}}}{\sum_{j=1}^{\ell} \mathbf{1}_{\{\sum_{m=1}^{M} \mathbf{1}_{\{|r_{k,m}(\mathbf{x}) - r_{k,m}(\mathbf{x}_j)| \leq \varepsilon_\ell\}} \geq M\alpha\}}},$$

$\alpha \in \{1/M, \ldots, 1\}$

# Typical COBRA output

$$\mathbf{X} \sim \mathcal{N}(0, \Sigma), \Sigma_{ij} = 2^{-|i-j|}, \; n = 700, \; d = 20.$$

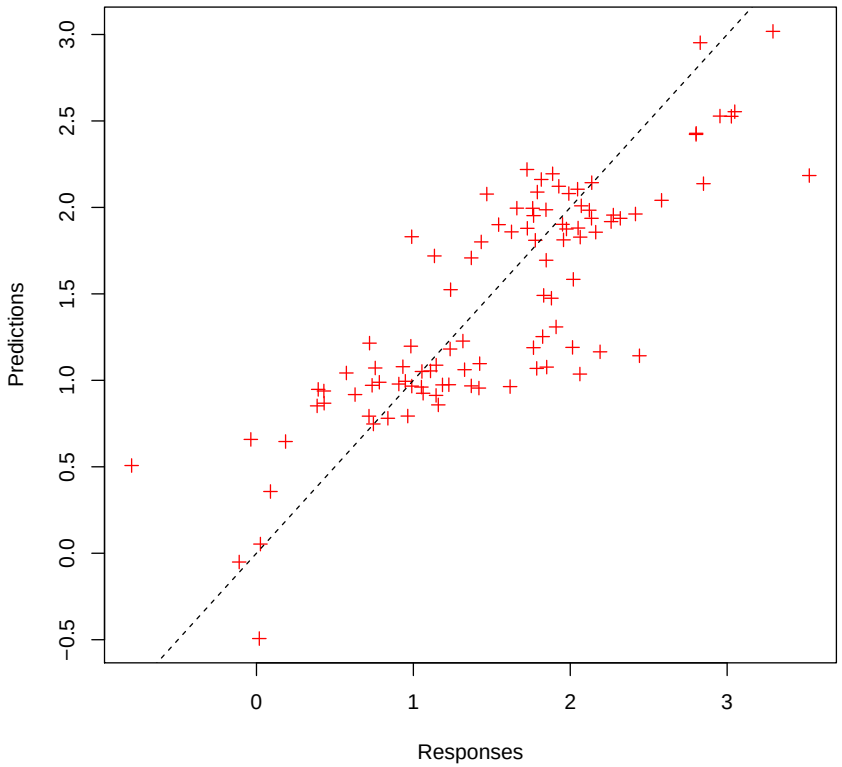
$$Y = \mathbf{1}_{\{X_1 > 0\}} + X_2^3 + \mathbf{1}_{\{X_4 + X_6 - X_8 - X_9 > 1 + X_{14}\}} + \exp(-X_2^2) + \mathcal{N}(0, 0.5).$$

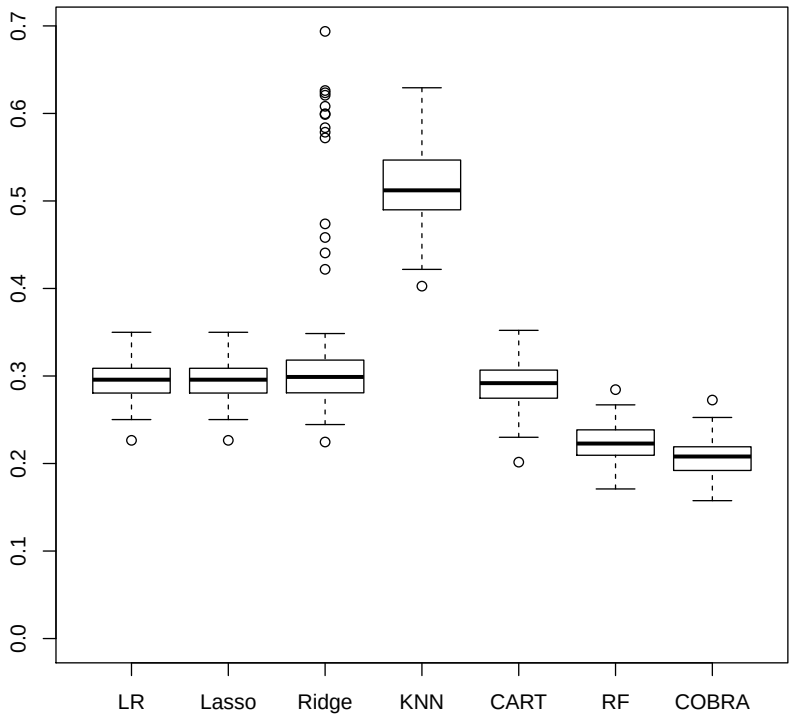


# Predictive performance: Example 1

$$\mathbf{X} \sim \mathcal{U}(-1,1)^d,\ n = 700,\ d = 20.$$

$$Y = \mathbf{1}_{\{X_1>0\}} + X_2^3 + \mathbf{1}_{\{X_4+X_6-X_8-X_9>1+X_{14}\}} + \exp(-X_2^2) + \mathcal{N}(0, 0.5).$$

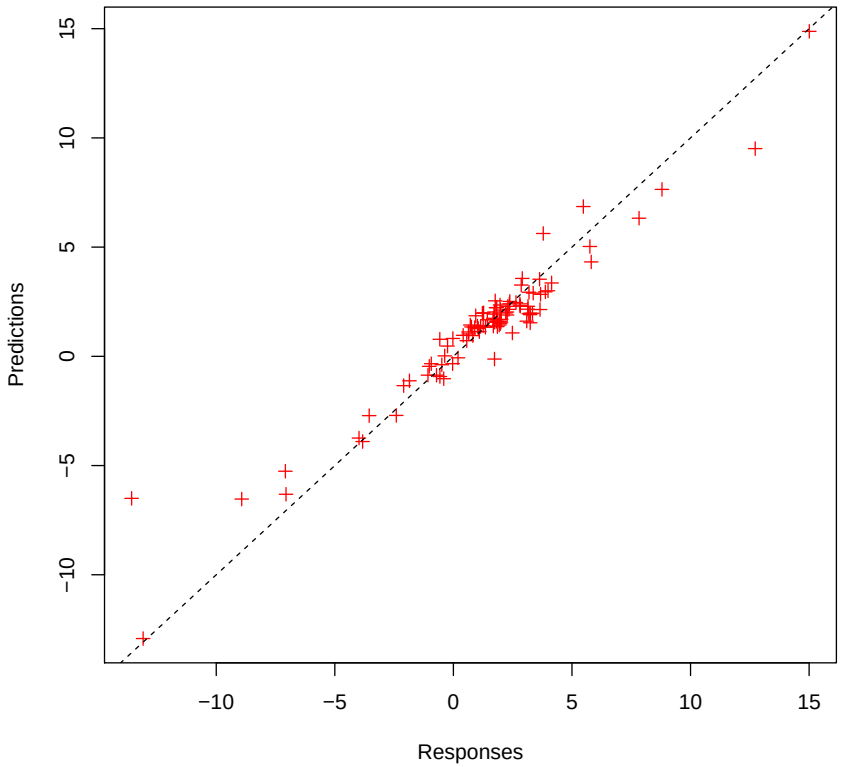


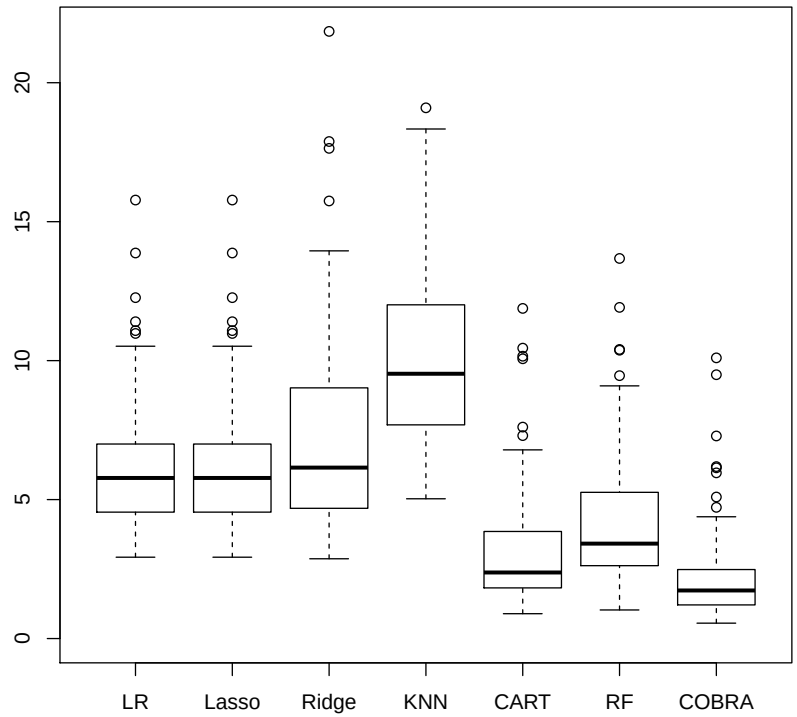



# Predictive performance: Example 2

$$\mathbf{X} \sim \mathcal{N}(0, \Sigma), \Sigma_{ij} = 2^{-|i-j|}, \; n = 700, \; d = 20.$$

$$Y = \mathbf{1}_{\{X_1 > 0\}} + X_2^3 + \mathbf{1}_{\{X_4 + X_6 - X_8 - X_9 > 1 + X_{14}\}} + \exp(-X_2^2) + \mathcal{N}(0, 0.5).$$

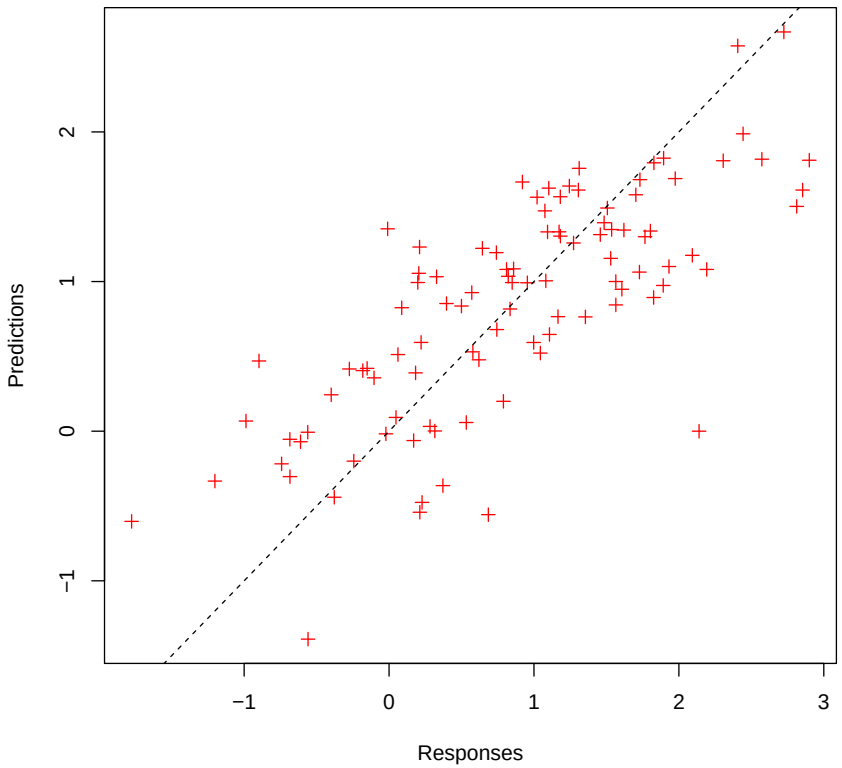


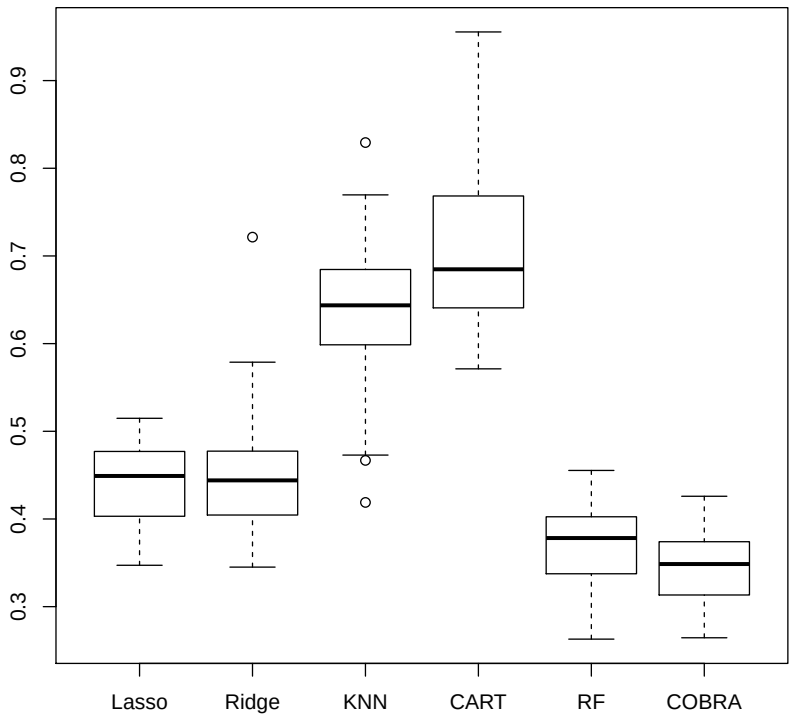



# Predictive performance: Example 3

$$\mathbf{X} \sim \mathcal{U}(-1,1)^d,\ n = 600,\ d = 10.$$

$$Y = \sum_{j=1}^{10} X_j^j + \mathcal{N}(0, 0.1).$$



# Highlights

- Original nonlinear strategy backed up by a sharp oracle inequality.

  Biau, Fischer, G. and Malley (2013). COBRA: A Nonlinear Aggregation Strategy. arXiv preprint.

- R package COBRA: extremely fast and flexible implementation.

  Version 0.99.4 on the CRAN.

# Technical regularity assumption

For any $m = 1, \ldots, M$,

$$r_{k,m}^{-1}((t, +\infty)) \underset{t \uparrow +\infty}{\searrow} \emptyset \quad \text{and} \quad r_{k,m}^{-1}((-\infty, t)) \underset{t \downarrow -\infty}{\searrow} \emptyset.$$